International Journal of
Interactive Multimedia
and Artificial Intelligence

September 2023, Vol. VIII, Number 3
ISSN: 1989-1660

unir LA UNIVERSIDAD
EN INTERNET

“Once again, the focus of activity in AI began to change, away from disembodied AI systems like expert systems and logical reasoners, toward building agents.”
Michael Wooldridge

Special Issue on Practical Applications of Agents and Multi-Agent Systems

INTERNATIONAL JOURNAL OF INTERACTIVE MULTIMEDIA AND ARTIFICIAL INTELLIGENCE

ISSN: 1989-1660 –VOL. 8, NUMBER 3

Editor's Note

Research in Agents and Multiagent Systems has matured significantly in recent years, representing one of the main branches of Artificial Intelligence and currently there are numerous effective applications of these technologies combined with Deep Learning, Computer Vision or Natural Language Processing, including areas such as healthcare and Ambient Intelligence, smart cities and mobility, Industry 4.0, educational technology, and fintech, among many others. In this regard, the International Conference on Practical Applications of Agents and Multi-Agent System (PAAMS) provides an international forum to present and discuss the latest scientific advances and their effective applications in different sectors, evaluate the impact of the approach and facilitate technology transfer among different stakeholders.

Currently, a series of co-located events specialized in different areas of research are held simultaneously with PAAMS, these being the International Congress on Blockchain and Applications (BLOCKCHAIN), the International Conference on Distributed Computing and Artificial Intelligence (DCAI), the International Conference on Decision Economics (DECON), the International Symposium on Ambient Intelligence (ISAmI), the International Conference on Methodologies and Intelligent Systems for Technology Enhanced Learning (MIS4TEL), and the International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB).

In this regard, the present Special Issue includes a selection of extended papers presented at the 20th International Conference PAAMS 22 and its co-located events and held in L'Aquila (Italy), July 13-15, 2022. Specifically, the present Special Issue includes the topics described below.

Sevilla-Salcedo et al. proposed the application of state-of-the-art natural language generation models to provide social robots with more diverse, less repetitive and friendlier language when interacting with human users. The authors implemented and evaluated a paraphrasing module and a speech generation module that adapts to the user's conversation, showing great potential.

Carrascosa et al. presented research in the field of Federated Learning and Multi-Agent Systems, proposing a consensus-based learning algorithm called Co-Learning by the authors. Co-Learning uses a consensus process to share the artificial neural network models that each agent learns using its private data and computes the aggregated model.

Michelena et al. proposed an intelligent classification model for Denial of Service (DoS) attack detection, evaluating the performance of six supervised classification techniques (Decision Trees, MLP, Random Forest, SVM, Fisher Linear Discriminant, and Bernoulli and Gaussian Naive Bayes) combined with the PCA feature extraction method for DoS attack detection in MQTT networks.

Martí et al. presented a survey on demand-responsive transportation for rural and interurban mobility. This work brought together papers that discuss, analyze, model, or experiment with demand-responsive transportation systems applied to rural settlements and interurban transportation, discussing their general feasibility, as well as the most successful configurations.

Ferarria et al. presented an investigation on different text representations to train an artificial immune network for text clustering. This work investigated four classes of text structuring methods to prepare documents to be clustered by an artificial immune system called aiNet, evaluating the influence of each structuring method on the quality of the clustering performed.

López-Flórez et al. proposed a solution adopting a YOLOv5 network model for automatic cell recognition and counting in a case study for laboratory cell detection using images from a CytoSMART Exact FL fluorescence microscope. A laboratory test was also performed to confirm the feasibility of the results, successfully recognizing and counting the different cell types.

Durães et al. presented experiments conducted using in-car audio data and deep learning frameworks for the purpose of violence identification. In this regard, the authors created a custom dataset tailored to this specific scenario. Based on the results obtained for that dataset, the EfficientNetB1 neural network demonstrated the highest accuracy (95.06%).

Bernabé-Sánchez et al. proposed a framework to manage and detect errors and malfunctions of the devices that compose an IoT system, considering simple devices such as sensor or actuators, as well as computationally intensive Edge devices which are distributed geographically. This work also presented the edge-cloud ontology (ECO) to organize the IoT system information.

López-Blanco et al. studied the evolution of pollutants in different Spanish cities using Generative Additive Models (GAM), proven to be efficient for making predictions with detailed historical data which have strong seasonalities. This study concluded that during the COVID-19 pandemic containment period, there was an overall reduction in the concentration of pollutants.

We would like to thank everyone who contributed to the research topic, including the authors, the reviewers, the organizers of the PAAMS conference and its co-located events, and the IJIMAI editorial and production offices.

Funding: This research has been supported by the project "COordinated intelligent Services for Adaptive Smart areaS (COSASS), Reference: PID2021-123673OB-C33, financed by MCIN /AEI /10.13039/501100011033 / FEDER, UE.

Ricardo S. Alonso[1,2]

Pablo Chamoso[3]

Sara Rodríguez-González[3]

Paulo Novais[4]

[1] AIR Institute (Spain)

[2] Universidad Internacional de La Rioja (Spain)

[3] Universidad de Salamanca (Spain)

[4] Universidade do Minho (Portugal)

DOI: 10.9781/ijimai.2023.08.008

TABLE OF CONTENTS

OPEN ACCESS JOURNAL

ISSN: 1989-1660

The International Journal of Interactive Multimedia and Artificial Intelligence is covered in Clarivate Analytics services and products. Specifically, this publication is indexed and abstracted in: *Science Citation Index Expanded, Journal Citation Reports/ Science Edition, Current Contents®/Engineering Computing and Technology.*

Using Large Language Models to Shape Social Robots' Speech

Javier Sevilla-Salcedo*, Enrique Fernández-Rodicio, Laura Martín-Galván, Álvaro Castro-González, José C. Castillo, Miguel A. Salichs

Universidad Carlos III de Madrid, Madrid (Spain)

Received 14 February 2023 | Accepted 20 July 2023 | Published 27 July 2023

Abstract

Social robots are making their way into our lives in different scenarios in which humans and robots need to communicate. In these scenarios, verbal communication is an essential element of human-robot interaction. However, in most cases, social robots' utterances are based on predefined texts, which can cause users to perceive the robots as repetitive and boring. Achieving natural and friendly communication is important for avoiding this scenario. To this end, we propose to apply state-of- the-art natural language generation models to provide our social robots with more diverse speech. In particular, we have implemented and evaluated two mechanisms: a paraphrasing module that transforms the robot's utterances while keeping their original meaning, and a module to generate speech about a certain topic that adapts the content of this speech to the robot's conversation partner. The results show that these models have great potential when applied to our social robots, but several limitations must be considered. These include the computational cost of the solutions presented, the latency that some of these models can introduce in the interaction, the use of proprietary models, or the lack of a subjective evaluation that complements the results of the tests conducted.

Keywords

Human-Robot Interaction, Large Language Models, Social Robots.

DOI: 10.9781/ijimai.2023.07.008

I. Introduction

INTELLIGENT conversational agents are increasingly being integrated into public-facing tasks such as customer service tasks, including troubleshooting and providing information; assistive chatbots have been used in multiple fields [1]. For interactions with the agent to be smooth, the agent must present itself naturally and intelligently to the user [2]. One of the main barriers is people's reluctance to interact with artificial systems due to unnatural text and, as a result, a conversation experience that is not very fluent, which leads to mistrust and uneasiness [3]. Social robots are embodied conversational agents, and to enable them to advance, we must focus on what makes a conversation natural and fluid, for example, by moving away from predefined and repetitive texts that lack naturalness [4].

One important limitation of current social robots is related to the robot's ability to convey information orally. It is common for robots to use the same expressions repeatedly, which leads to user boredom and, thus, a loss of interest in interacting with the robot. While it is desirable to give robots the ability to generate and adapt the content of their speech, there are situations in which the use of handcrafted texts might have advantages (e.g. when the robot is requesting information from the user). However, this can turn into a limitation if these tailored speeches become repetitive to the user. In these cases, a possible solution could be the development of strategies for phrasing these handcrafted texts differently but maintaining their original meaning.

Alternatively, in the field of social robotics, the robot is expected to be able to take the initiative in the interaction, so mechanisms for generating engaging topics and developing them are an interesting addition to this type of robot [5]. These mechanisms can be extended to allow the user to choose the topic to be discussed and to allow the robot to elaborate a discourse around the chosen subject. Furthermore, an interesting adaptation mechanism in human-human interaction is the ability of each speaker to adapt their speech to the other speaker, considering aspects such as age, familiarity, or background. In this sense, it would be desirable to endow the robot with the ability to perform a similar adaptation.

Both of the objectives presented above can be achieved through the use of Natural Language Processing (NLP). This field has attracted significant attention in recent years across multiple disciplines, including robotics. For example, in 2014, Woo et al. [6] combined different NLP techniques to create a conversational system for robotics. Their system uses predefined rules to construct sentences based on the user's input, with a set of fallback sentences for cases in which no sentences is constructed. That same year, Fujita et al. [7] presented an NLP-based model for the Todai robot with the goal of passing the entrance test for the University of Tokyo. The proposed model was trained by using previous exams as inputs. The results obtained were good for mathematics and physics, but the robot failed to pass the history and language portions of the exam. In 2016, Hammed [8] proposed a conversational system for social robots that uses a neural network to build a user profile with knowledge extracted from dialogues with the user and then uses this profile to adapt the conversation. A year later, Williams [9] proposed a text generation framework for robots that relied on a memory model distributed over

* Corresponding author.

E-mail addresses: javier.sevilla@uc3m.es

two levels. The long-term memory level stores information about locations, objects, and people, while the short-term memory level focuses on the context of the dialogue.

In 2018, Kahuttanaseth et al. [10] presented a system for guiding a mobile robot using natural language. This system uses a Recurrent Neural Network (RNN) encoder-decoder system that filters unnecessary words from the inputs received (e.g. 'please') and then extracts movement command sequences. Although in a limited context, this system can manage multiple commands in a single input. More recently, in 2020, Budiharo et al. [11] tested recurrent and convolutional neural networks as encoders in a question-answering model. The proposed system uses a bidirectional attention flow mechanism [12] to find the similarity between the question asked by the user and the questions in a dataset. Then, it uses this similarity to find the proper answer to the question. Their results showed that the RNN-based encoder performed relatively well. In 2023, Arroni et al. [13] proposed using the Transformer architecture for sentiment analysis. For this, they proposed a network consisting on a single transformer block with 11 attention heads, using learned embeddings of dimension 12 for positional and token encoding. Their results showed that this model outperformed a pre-trained DistilBERT when evaluating them using the Spearman correlation, although not in validation accuracy. That same year, Zhou et al. [14] evaluated the use of ChatGPT for conveying knowledge about gastric cancer (through a medical knowledge test), providing consultation recommendation to patients, and analysing endoscopy reports. Their results show that, while displaying high levels of appropriateness and consistency in its responses, ChatGPT may not always provide accurate responses and suggestions, which indicates that we should not over rely on this model for critical tasks like clinical diagnosis.

Similar to the works presented above, our research also aims to use NLP to allow conversational agents to provide a more natural conversational experience [15]. How the agent expresses itself is crucial for smooth interactions, and the user's perception of the robot can be affected by the responsiveness of its speech. We have used natural language models to tackle these problems, as they provide a new approach to generating non-prewritten texts. On top of that, given that our system has been integrated into a real social robot that interacts with Spanish people, one of the requirements is that it must work in Spanish, so we will consider two methods when exploring language models: on the one hand, we will evaluate multilingual models that include the Spanish language, and on the other hand, we will evaluate models in English, which a priori offer better performance, and use translation tools to generate the desired output.

Another constraint the system must consider is the computing power necessary to run large language models locally and in the cloud. In social robotics, this becomes even more important since these platforms often have limited resources that have to be shared by multiple modules within their software architectures. In particular, the *'two-second rule'* is used to set the maximum delay between interaction turns to two seconds [16], although other authors have reported that users prefer shorter times of around one second [17].

In this work, we seek to mitigate the issues introduced by relying only on predefined texts, which include the reduced fluidity and naturalness of the interactions with users [18], without losing the possibility of using handcrafted texts in situations in which they can be beneficial. In short, when it comes to speech, there are two possibilities: the robot can use predefined sentences or generate new text. We intend to provide solutions to both problems using NLP to obtain a fluent and spontaneous experience during conversations between humans and social robots. Therefore, the first contribution of this article is a methodology that allows the robot to generate pop-up conversation topics and information about them. In addition, the system can dynamically adapt the information to different user profiles. This makes the information more accessible and appealing to users. The idea behind this methodology is that, in social robotics, it is desirable for robots to have the ability to establish non-predefined conversations with users, even without a specific objective. A social robot must be able to interact even when it is not performing a specific task, and these interaction mechanisms must include speech. Therefore, it is important to have mechanisms capable of generating text on the fly on non-predefined topics.

The second contribution of this work is the integration of language models to enrich the verbal capabilities of a social robot, allowing it to paraphrase its repertoire of predefined sentences to achieve greater variability and reduce monotony. We expect that this paraphrasing will improve the quality of the interaction, preventing the robot from becoming monotonous and repetitive.

The rest of this paper is structured as follows. Section II presents the tools and models used in this work. Next, Section VI covers the evaluation setup and metrics employed to assess the proposed methods. Next, Section III presents the first contribution of this work, which deals with the use of NLP models to create semantic descriptions of topics of interest that are adapted to users. Section IV discusses the second main contribution of this work, the paraphrase generator, which will provide the robot with more variability in its speech. Section V describes the integration of both contributions into a real social robot. Section VI presents the evaluation setup, the evaluation of our approach and the models used. In Section VII, the main results of the developed methods are presented. Finally, Sections VIII and IX discuss the main results and draw conclusions.

II. Materials

This section reviews the language models used during the development of this project. Since the inclusion of transformers in the world of deep learning, the development of language models has grown remarkably, along with their capabilities. In this paper, we used these models to give our robots the ability to generate text from scratch and paraphrase the handcrafted texts that they use automatically.

A. Transformers in Language Modelling

A transformer is a deep learning model built on self-attention mechanisms. These mechanisms assess the input data by weighting the significance of each component. While self-attention mechanisms were first included in RNN structures, transformers are built on self-attention alone, and they provide a better performance than RNNs [19]. Transformers have an encoder-decoder architecture in which the encoder layer consists of modules that sequentially handle the input sequence one module at a time. On the other hand, the decoder layer is composed of modules that handle the encoder's outputs. Each encoder layer generates new encoding vectors. These vectors contain information about the components of the inputs that are relevant to each other. Each decoder layer extracts the generated encodings and builds an output sequence out of the decoding and the encoded contextual information [20]. To achieve this, each encoder and decoder module uses the self-attention mechanism [21]. The use of transformers as a new base architecture for language models has brought about a significant change in the capabilities of language models [22]. At the language model level, an increasing number of transformer-based networks have emerged and continue to emerge [23].

B. From Fine-Tuning to Prompt Learning

The way in which models are trained for specific tasks has evolved over time. Traditionally, models have been trained from scratch, starting from random weights and adjusting these weights using a dataset relevant to the task to be performed; as the size of these models has grown, this has become a long and tedious process. Over the years, to improve the results and reduce training times, researchers have adopted new methodologies, such as transfer learning, which uses the knowledge (weights) of a model trained for a known task as initial weights to train a new model for a new task [24]. Subsequently, fine-tuning was introduced; it allows pre-trained models to be taught new specific tasks. In this context, fine-tuning refers to a technique in deep learning in which the weights of a pre-trained model are readjusted by training without losing their initial settings. The main layers of the model are frozen so that it is better adapted to the tasks for which it is trained [25]. One of the advantages of this is that fine-tuning does not involve training the entire model but rather involves updating its gradient, which is a significantly faster process. Later, more advanced models, such as Generative Pre-trained Transformer 3 (GPT-3) and Text-to-Text Transfer Transformer (T5), went a step further and evolved from fine-tuning to prompt learning [26]. In contrast to fine-tuning, prompt learning or in-context learning refers to a technique in which examples are added within the model's input prompt so that the model can understand the expected inference behaviour without the need to be fine-tuned or specifically trained. Occasionally, prompt learning is referred to as n-shot learning; for example, there can be 'few-shot', 'one-shot', or 'zero-shot' learning. These terms refer to the number of examples of the task given to the model: there can be few, one, or zero examples, respectively.

Fig. 1 shows examples of different types of prompt learning. One of the advantages of n-shot learning is the inclusion of encodings that help the model understand the task it has to perform. For example, in the example shown in Fig. 1, we find the encoding '=>'. This pattern indicates that the model has to look at the sequence before the encoding and then add its result afterwards. In this case, the examples added to the prompt will help the model understand the task of adding integers. When designing a prompt, the number of examples required for correct functioning will depend on the task to be performed, the complexity of the text, and even the sequence format. There is a trade-off between the number of examples and the length of the input sequence, as a longer input sequence means that the model must process more data; hence, more resources must be used, and the inference times will be longer.

```
- Zero-shot -
input_Prompt:
Add two integers:  #task description
2 + 5 =>           #prompt
```

```
- One-shot -
input_Prompt:
Add two integers:  #task description
8 + 4 => 12        #example
2 + 5 =>           #prompt
```

```
- Few-shot -
input_Prompt:
Add two integers:  #task description
8 + 4 => 12        #example
1 + 4 => 5         #example
2 + 9 => 11        #example
3 + 5 => 8         #example
2 + 5 =>           #prompt
```

Fig. 1. Prompt learning examples. Top: Zero-shot learning; Middle: One-shot learning; Bottom: Few-shot learning.

C. Models Used

For the development of the functionalities proposed in this work, several models have been considered, and all of them are listed in Table I. These models are well-known for their performance and are extensively used in different applications. One constraint that has to be kept in mind is the language in which these models have been trained. As stated in the introduction, our robots have been specifically designed to interact with older adults who only speak Spanish. This means that all the predefined texts used by our platforms, which are the texts that we need to paraphrase, are written in this language. For the development of the specific modules for user-adapted semantic description generation presented in Section III, GPT-3 has been used exclusively due to its great adaptability to all the objectives of the application [27] [28]. While GPT-3 has been trained with a corpus of text written in English (which means that we have to translate the results obtained into Spanish), we decided that the level of performance of this model justified the need for this translation step. While the user-adapted semantic description generation pipeline generates text from scratch, the deep learning-based paraphrase generation module does need to work with texts that are prewritten in Spanish, which is a limitation that has to be considered when selecting the model to use for this task. We tested two approaches: (i) using models to paraphrase Spanish sentences directly, and (ii) translating the sentences into English, using models for paraphrasing English sentences, and finally, translating the results back into Spanish. For this task, in addition to testing the performance of GPT-3, we also tested T5, *multilingual T5* (mT5), *Pre-training with Extracted Gap-sentences for Abstractive Summarisation Sequence-to-sequence* (PEGASUS), and BERT2BERT.

GPT-3 is an auto-regressive language model capable of producing human-like text [29] from an input sequence. It was trained with about 45 TB of text data, which led to the refined learning of the language domain and allowed the model to learn new mnemonic rules online. It has been implemented in sentiment analysis [30], used to generate programming code [31], and used for text summarisation [32], among other things. Within GPT-3, there are several versions of the model depending on the size of its architecture, and thus, it has varying capabilities. For our work, we used the Babbage and Davinci models, which are described in Table I.

T5 [33] is a model designed for NLP tasks like translation, summarisation, and question answering, which are all reframed as text-to-text problems. This makes it possible to reuse models, hyperparameters, and loss functions during training for different tasks. This method explores the advantages of scaling the model and the corpus size by using 11 billion parameters during training and the Colossal Clean Crawled Corpus (C4) [33], which includes hundreds of GB of natural-language text. In this work, we tested two checkpoints of the T5 model from HuggingFace [34]. We will refer to them as *PMO-T5*[1] and *Parrot*[2].

A variation of the T5 model used in this work is multilingual T5 (mT5) [35]. It has a similar architecture but has been trained to work in languages other than English using a multilingual version of the C4 dataset. A second difference is that mT5 only uses non-supervised learning, which means that it has to be fine-tuned for any task it will be used for, as shown in Table I. We have fine-tuned an mT5 model[3] with the Spanish instances of the PAWS-X multilingual dataset [36], using the process recommended by HuggingFace[4] (because of this, we will refer to it as HFT5 in the evaluation section).

[1] https://huggingface.co/ceshine/t5-paraphrase-paws-msrp-opinosis

[2] https://huggingface.co/prithivida/parrot_ paraphraser_on_T5

[3] https://huggingface.co/seduerr/mt5-paraphrases-espanol

[4] https://huggingface.co/docs/transformers/training

TABLE I. Table That Summarises the Different Models Used in This Work. The Columns Are, in Order: (I) the Name of Each Model; (Ii) the Number of Parameters That the Model Uses; (Iii) the Architecture of the Model; (Iv) What Task the Models Were First Trained for; (V) If the Model Allows or Not for Multiple Languages; (Vi) How Accessible Is Each of the Models; and (Vii) the Process Required for Adapting the Model to a new Task

Model		Parameters	Architecture	Original Task Training	Multilingual	Access	Task-Specific Training
GPT-3	Babbage	1.3B	Decoder	Next Word Prediction	Yes (Fine-Tuning)	Limited (Paid-Propietary API)	Prompt Learning & Fine-Tuning
	Davinci	175B					
T5		11B	Encoder-Decoder	Masked Language Modeling “span-corruption”	No	Open Source	Prompt Learning & Fine-Tuning
Multilingual T5		300M	Encoder-Decoder	Masked Language Modeling “span-corruption”	Yes	Open Source	Fine-Tuning
PEGASUS		568M	Encoder-Decoder	Gap-Sentence-Generation	No	Open Source	Prompt Learning & Fine-Tuning
BERT2BERT		110M + 110M	Encoder-Decoder	Masked Language Modelling & Next Sentence Prediction	Yes (Fine-Tuning)	Open Source	Fine-Tuning

TABLE II. Inference Parameters Used for Each Pipeline Module. Engine Indicates the Type of Model Used; Temperature Controls the Degree of Randomness of the Response. Response Length Limits the Maximum Number of tokens to Be Generated; Top P Controls to Some Extent the Randomness and Creativity of the Response. Frequency Penalty Penalises the Repetition 1 of Tokens in the Output, While Presence Penalty Penalises the Generation of 2 New tokens Already Present in the Input Text. Finally, Stop Sequences Are Encodings to Stop Generation

Parameter	*Random Topic Generation*	*Semantic Description Generation*	*User-adapted text modification*
Engine	*Babbage*	*Babbage*	*Davinci*
Temperature	0.64	0	0.6
Response length	54	500	200
Top P	1	1	1
Frequency penalty	2	1.92	0.4
Presence penalty	2	0	0.2
Stop Sequences	\n,subject:	\n	“””

The PEGASUS model [37] was designed for summarising texts with different words (abstractive summarisation). It was first pretrained to predict missing sentences in an input text, as this is similar to abstract summarisation. This was done through self-supervised training using a corpus of documents extracted from the web, like the C4 or HugeNews datasets, and then the model was tuned using 12 datasets for abstractive summarisation. Compared with the T5 model, PEGASUS presents similar results with 5% of the parameters. In this work, we have fine-tuned the PEGASUS model for paraphrase generation[5].

The BERT2BERT model has an encoder-decoder architecture in which both components are modelled as *Bidirectional Encoder Representations from Transformers* (BERTs) [38]. The goal is to achieve a larger size with lower resource usage compared to the T5 and PEGASUS models. BERT2BERT's language model has a deeper knowledge of a language's context thanks to bidirectional training. It was trained for *masked language modelling* (predicting missing words in a text) and *next sentence prediction* (predicting the sentence that follows another sentence), as shown in Table I. During training, the weights for the layers present in the BERT model are initialised with the original values from this model, while the layers specific to the BERT2BERT model are initialised to random values. In this work, we used a version of BERT2BERT that uses an encoder and decoder trained in Spanish [39], and we fine-tuned it for paraphrase generation[6].

III. User-Adapted Semantic Description Generation

This section presents the application of natural language generation techniques to generate user-adapted semantic descriptions. We have proposed a modular design; the pipeline is divided into three main modules, and each one can work independently. Fig. 2 shows a diagram of the developed pipeline, where the random topic generator module generates a topic that is later fed to the semantic description generator module, which creates a description from the given topic. Finally, the user-adapted text modifier module takes the generated description and adapts it according to the type of user with whom the robot is interacting. For conciseness, in the rest of the text, the entire pipeline will be referred to as user-adapted semantic description generation (UASDG). Each module performs an independent inference in the pipeline with individually tuned parameters for optimal performance. Table II lists the parameters used in each module.

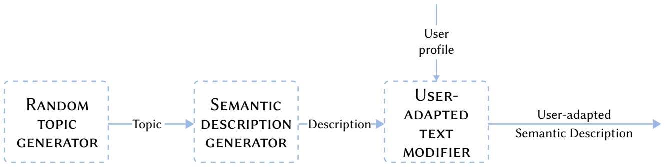

User profile
Random topic generator
Topic
Semantic description generator
Description
User-adapted text modifier
User-adapted Semantic Description

Fig. 2. User-Adapted Semantic Description Generator Diagram.

A. Random Topic Generation

This module can randomly generate a topic, which gives the social

[5] https://huggingface.co/tuner007/pegasus_paraphrase

[6] https://huggingface.co/mrm8488/bert2bert_shared-spanish-finetuned-paus-x-paraphrasing

robot a certain spontaneity that enriches human-robot interactions [40]. The idea was to make inferences with the model and obtain a different theme with each execution. As mentioned in Section II, one of the significant advantages of using language models, such as GPT-3, is the use of prompt learning to adjust the model to the needs of a specific task without the need for application-specific fine-tuning training. With this in mind, several input prompts were designed and tested for the model to generate random topics. Fig. 3 (lines 1 to 10) shows how using prompt learning, the model is given several examples with the desired behaviour. The header, together with the examples, helps the model to understand what we are aiming for. Additionally, the inference parameters *Temperature* and *Top P* are adjusted to always favour creativity and not repetitiveness, as shown in Table II. In addition, using the encoding '*subject: -topic-*', the model understands the expected behaviour in a refined way. As seen in the figure, the input prompt shows the desired pattern with eight examples so that the model can understand the pattern to follow. In this case, the topic generated was 'Trees'.

```
 1  input_Prompt: The following is a list of random
                  subjects for generating knowledge
                  about the world:
 2  subject: Pencils

 3  subject: Radiology

 4  subject: Fishing

 5  subject: Napoleon

 6  subject: Dolphins

 7  subject: Keys

 8  subject: Pigeons

 9  subject: Newspapers

10  subject:

11  model Output: Trees
```

Fig. 3. Random Topic Generator example.

B. Theme-Based Semantic Description Generator

This module is designed as a tool to generate a semantic description given an input topic. To this end, as with the topic generation module, numerous input prompts were designed and implemented to test the functionality and exploit its potential using GPT-3. Initially, among other prompts, we tested *'Let me give relevant semantic information about -any topic-'*; however, although the model generated sufficient text, the model generated numerous facts surrounding the theme but failed to provide useful information. Finally, after numerous iterations, the final prompt was *'I have been doing deep research about -any topic-, let me give you brief, relevant semantic information about it'*, which is shown in Fig. 4. This prompt generated a sufficient amount of information in a precise way without too much extra text. In this case, unlike in the previous module, the inference parameters, which are shown in Table II, have been adjusted to favour the text's veracity and thus compromise its creativity.

The adjusted model provides promising results, as it fits the initial purpose correctly, even with zero-shot learning (see Section II). As for the implementation, like the previous module, it is developed as a stand-alone component. Fig. 4 shows the results for the input prompt with the input topic *ladders*.

```
input_Prompt:   I have been doing deep research about
                ladders,  let me give you brief, relevant
                semantic information about it:

model Output:   Ladder is a tool used to climb up or down.
                It can be made of wood, metal, plastic and
                other materials. Ladders are also called as
                steps or rungs in the ladder system. They
                are usually placed on walls and floors for
                easy access to different levels of buildings
                such as apartments, offices etc.
```

Fig. 4. Semantic Description Generator example.

C. User-Adapted Text Modification

Social robots interact with very different and diverse types of people; for example, they can go from talking to a child to talking to an older adult. Therefore, when talking about specific or complex topics, these interactions could be enhanced by a mechanism capable of adapting the text we want to transmit and paraphrasing it according to the user. To this end, after previous experience working with generative models, we have searched for a prompt with a natural language structure that is capable of generating this type of adaptation using zero-shot learning with GPT-3.

Within the input prompt, we briefly describe a person, indicating the type of user (an older person, a child, etc.) so that the model can adapt the paraphrase. We used the encoding (""") within the prompt; this indicates a change from the description to the text to paraphrase. Before the definition, we inform the model that a person has asked us to tell them what the text means. Fig. 5 shows two examples of inference in which we copy the definition and history of gravity from an encyclopaedia[7] and ask the model to adapt it in the first example for an *older person* and in the second example for a *child*. Line *1* of the figure shows the input prompt used for this application. Although only two not-very-descriptive types are shown in the examples, the user information obtained from the robot's perception system can describe the user in a considerable amount of detail to help the model fit the requirements of different users.

As with the applications described above, this application has been implemented modularly as a stand-alone function. In this case, the input prompt has kept the main structure shown in Figure 5; however, making use of the information about the user, we modify the first and last sentences of the prompt, and the text to be modified is introduced into the input prompt following the defined structure, as shown in line 3 of Fig. 5. The output of the module will be the modified text. As shown in the figure, both examples provide adapted descriptions. The output for an older adult explains gravitation as an attractive force among objects using less technical and lighter language. On the other hand, the output for a child simplifies the description even further, leaving behind the 'attraction' concept and focusing instead on the concepts of 'falling' and 'pulling down', which are more relatable to a child.

IV. Deep Learning-Based Paraphrase Generation

While giving social robots the ability to generate texts from scratch is a desirable feature that can improve the naturalness of interactions with users, there are still situations in which having the ability to handcraft the speech of the robot can provide some advantages. However, it is important that these predefined interactions have some variability so that they do not become repetitive. A solution for this is adding methods for paraphrasing these prewritten texts. The proposed method receives the original sentence and returns a

[7] https://www.britannica.com/science/gravity-physics

```
input_Prompt:   A friend, who is really old, asked me
                what this passage means:
"""
Gravity, also called gravitation, in mechanics, the
   universal force of attraction acting between all
   matter. It is by far the weakest known force in
   nature and thus plays no role in determining the
   internal properties of everyday matter. On the
   other hand, through its long reach and universal
   action, it controls the trajectories of bodies in
   the solar system and elsewhere in the universe and
   the structures and evolution of stars, galaxies, and
   the whole cosmos. On Earth all bodies have a weight,
   or downward force of gravity, proportional to their
   mass, which Earth's mass exerts on them. Gravity is
   measured by the acceleration that it gives to freely
   falling objects. At Earth's surface the acceleration
   of gravity is about 9.8 metres (32 feet) per second
   per second. Thus, for every second an object is in
   free fall, Its speed increases by about 9.8 metres per
   second. At the surface of the Moon the acceleration of
   a freely falling body is about 1.6 metres per second
   per second.
The works of Isaac Newton and Albert Einstein dominate
   the development of gravitational theory. Newton's
   classical theory of gravitational force held sway from
   his Principia, published in 1687, until Einstein's
   work in the early 20th century. Newton's theory is
   sufficient even today for all but the most precise
   applications. Einstein's theory of general relativity
   predicts only minute quantitative differences from the
   Newtonian theory except in a few special cases. The
   major significance of Einstein's theory is its radical
   conceptual departure from classical theory and its
   implications for further growth in physical thought.
"""
I rephrased it for him in plain language an elder can
understand:
"""
model Output:   Gravity is a force that exists between
                two objects. It causes the objects to
                have an attractive force on each other.
                The more massive the object is, the
                stronger this attractive force becomes.
```

```
input_Prompt:   A friend, who is a child, asked me what
                this passage means:
"""
- Text about Gravity -
"""
I rephrased it for him in plain language a child can
understand:
"""
model Output:   Gravity is the force that makes things
                fall down. It is the
                weakest force in nature, but it is very
                powerful because it affects
                everything in the universe. Gravity is
                measured by how fast something
                falls. The faster it falls, the more
                gravity is pulling it down. Gravity is
                what makes the Earth's surface curved.
```

Fig. 5. Examples of text modification adapted to the user. The example shown above is adapted for an elderly person, while below is the same text adjusted for a child.

paraphrased version of the input text. The platform into which this method has been integrated has Spanish as its default language. However, because the body of work that focuses on solving NLP tasks in English is larger than that in Spanish, we have decided to test two different approaches to perform this task: (i) we generate the paraphrase directly in Spanish, or (ii) we use a model trained on texts in English, translate the texts used by the robot from Spanish to English before passing them to the model, and translate them back into Spanish after they have been paraphrased. This last approach can be seen in Fig. 6.

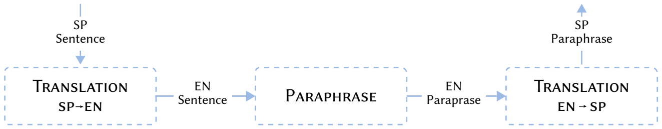


Fig. 6. Diagram of the exploitation process for models that paraphrase sentences, translating them to English before passing the text through the model, and translating the output of the model back to Spanish.

When considering which model to use for our application, one of the key factors was the similarity between the meanings of the original and paraphrased sentences. Our goal was to develop a model that is able to add variability to the robot's speech while maintaining the meaning conveyed by the original utterances. To paraphrase sentences in Spanish, we used the mT5, HFT5, and BERT2BERT models, and fine-tuned them using a Spanish paraphrase dataset, the Paws-x [36]. This task proved to be difficult due to the limited amount of available pretrained models and paraphrase datasets in Spanish. For models that paraphrase sentences in English, we were able to find models that had already been pretrained and fine-tuned on paraphrasing tasks. In particular, we tested PMO-T5, Parrot, PEGASUS, and GPT-3. When these models were used, the original text in Spanish was translated before passing it through the model, and the output was translated back into Spanish.

When we run our paraphrase generator, we can specify which of the tested models will be used to generate paraphrases. If we want to select a model that paraphrases text in English, we can also choose which translator the pipeline will use to convert the texts into English and back into Spanish. The translators that have been tested are the Google Translate[8], DeepL[9], and Argos translators[10].

As stated before, resource usage can be a limitation when working with deep learning models. This is particularly concerning for the task we are trying to perform, as the paraphrase generation module will be involved in the majority of the interactions between the user and the robot, and thus, it must abide by the time constraints that exist in any conversation. While this limitation can be mitigated by deploying these models on specialised hardware, there might be situations in which this is not an option (for example, if the robot is in a location with bad internet). To obtain as much flexibility as possible, our paraphrase module allows both the local and remote execution of the language model. When the module is running externally, the robot sends the utterance that has to be transformed to the server, paraphrase generation is performed there (along with the required translations if needed), and then the server sends the resulting utterance back to the robot.

[8] https://cloud.google.com/translate/

[9] https://www.deepl.com

[10] https://pypi.org/project/argostranslate/

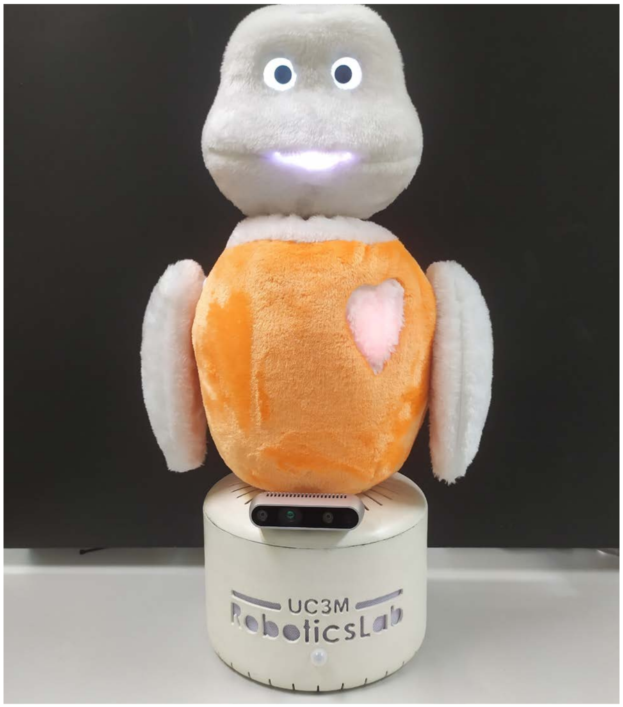


Fig. 7. Mini, a social robot developed for interacting with older adults suffering from mild cognitive impairment.

V. Integrating Our NLP Applications Into the Mini Robot

The models presented in Sections III and IV have been integrated into the social robot Mini [41], which is shown in Fig. 7. Mini is a tabletop robot with a soft appearance that is designed to assist older adults with mild cognitive impairment. This robot has five degrees of freedom (one per shoulder, another on the waist, and two more on the neck and head), OLED screens placed on the face to act as eyes, and coloured LEDs on its cheeks and on its chest. Regarding its perception capabilities, Mini is equipped with touch sensors on the shoulders and belly, a microphone and loudspeaker for speech-based interactions, and a touch screen that can be used both for interacting with users through menus and for displaying multimedia content.

Mini's architecture has been designed following a modular approach, as shown in Fig. 8. At the top of the architecture, a decision-making system (DMS) controls what the robot does at any given time based on stimuli coming from the environment, the inputs given by the user, and the knowledge the robot possesses. Below the DMS, there is a series of modules that allow Mini to perform different tasks: the skills. Examples of these skills include playing cognitive stimulation games, showing the user pictures, videos, music, and other multimedia content, and reading the news to the user, among other things. Here, we find the UASDG pipeline presented in Section III. It has been integrated as an individual skill that can be activated and deactivated by the DMS.

While the DMS and the skills control what task the robot performs at a given moment and how these tasks are performed, a second set of modules in Mini's architecture provides a series of transversal features for any task that Mini needs to complete. The liveliness module generates random behaviours (e.g. motions for all the joints or changes in gaze) to give Mini a lively appearance. The Perception Manager controls the modules capturing information from the environment and the user.

The Human-Robot Interaction (HRI) Manager is the module that controls any interactions between Mini and the robot. Whenever one of the skills needs to start an interaction or respond to a user command, it sends all the information necessary to the HRI Manager, which in turn ensures that the interaction is completed successfully. For example, whenever the UASDG pipeline generates a new text that has to be conveyed to the user, it is sent to the HRI Manager, which in turn ensures that the message is uttered properly and that there are no conflicts with any other interaction requests coming from other skills.

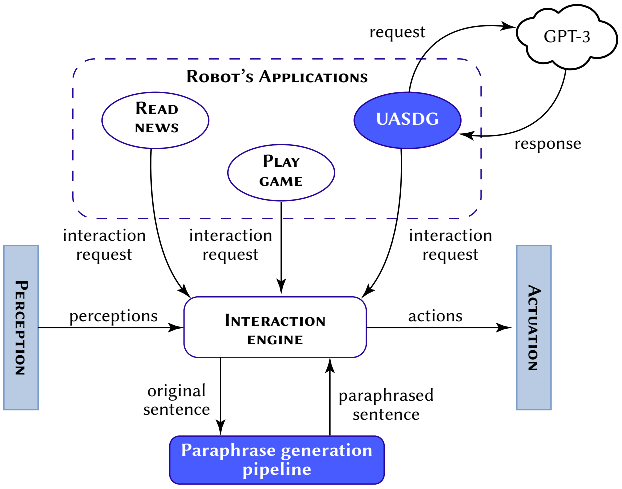


Fig. 8. Schematic view of Mini's architecture. The work presented in this manuscript has been integrated in the blocks in dark blue.

The Expression Manager controls how the robot's messages are conveyed and ensures no conflicts between them. It uses state machine-like structures to model multi-modal expressions. Among the elements in this module, the ones that are relevant for this work are the Interface Players. These Players receive each of the uni-modal actions that make up the expression (e.g. lifting an arm, saying a sentence, etc.) and send commands to the modules controlling the output interfaces (e.g. the drivers for the motors, the text-to-speech module, etc.). The Emotional Text-To-Speech Player is one of the Players and receives sentences that the robot has to utter, prepares them, and sends them to the text-to-speech module. The paraphrasing method described in Section IV has been integrated into this player.

During the startup stage of the software architecture, the ETTS Player loads a YAML configuration file for the paraphrase module. This file, shown in Fig. 9, specifies the model to be loaded and the translator that will be used (if the model parameters contain the name of one of the models trained to paraphrase sentences in English), a parameter that indicates whether the robot's utterances have to be paraphrased or not (this allows us to bypass the paraphrase module if we do not need it), and the deployment mode for the paraphrase module.

This is the case, for example, for the paraphrase generation module, as it is integrated into the Expression Manager, a key element in the interactions between the user and the robot. On the other hand, if the architecture presented in this section is deployed on a platform with enough resources, we might prefer to run the models locally to avoid potential problems caused by communication with external machines. Because of this, the tools presented in this work can be deployed in one of three manners. GPT-3 models are only accessible through the API provided by OpenAI, and thus they run the inferences on their servers. For those models that are accessible to researchers, we provide two possible solutions: running the models directly on Mini or deploying them on our external server. This server has been specifically designed to run machine learning models, which are too computationally demanding for Mini's hardware. It has an Intel Core i9-10900K CPU that runs at 3.7 GHz, an NVIDIA GeForce RTX 3090 GPU, and 64 GB of RAM.

```
paraphrase_config:{
     'model': 't5',
     'translator': 'deepl',
     'mode': 'local',
     'paraphrase': 'on',
     'pauses': 'on'
}
```

Fig. 9. Example of the YAML file used for configuring the paraphrase module.

VI. Evaluation Methods

This section outlines the evaluation setup for assessing the quality and efectiveness of paraphrased sentences, the evaluation of the user-adapted semantic description generation approach and the models used for paraphrase generation in Spanish and English. First, we describe the metrics that will allow us to compare the models used for user-adapted text modification and paraphrase generation. In this last case, separate comparisons have been performed for models trained in Spanish and models trained in English. Next, we describe the evaluations that have been conducted to test the two contributions presented in this manuscript.

A. Metrics

There are three main factors that we need to keep in mind when evaluating the quality of a paraphrased sentence: (i) the inference time should be as low as possible so that it does not hinder the interaction with the user; (ii) the meanings of the original and paraphrased sentences should be as close as possible; and (iii) the original and paraphrased sentences should be as different as possible. The inference time is a critical factor when using AI models in human-robot interactions, as studies have shown that responses in a conversation can lose their meaning if they are delivered too late. Times over two seconds make it impractical to achieve an optimal interaction. Moreover, to measure the similarity between two sentences' meanings and how they are written, we use two metrics widely used in NLP: the BiLingual Evaluation Understudy (BLEU) and BERT scores.

1. BLEU

The *BiLingual Evaluation Understudy* score [42] is used to evaluate the quality of an automatic translation; that is, it indicates the similarity between the translation generated by the model and a translation made by a human being. The main advantage of this metric is that it is easy to calculate and interpret, is language-independent, and tends to match human evaluations. Since its inception, it has spread from automatic translation to other NLP tasks, such as paraphrase generation. Using BLEU in our evaluation, we will be able to assess if the paraphrasing process generates sentences that are different enough to add real variability to the robot's speech.

BLEU compares matching words in both sentences, known as *n-grams*, where *n* indicates the number of words compared simultaneously. This metric also penalises the candidate sentence based on the lengths of the original and candidate sentences. Once the metrics for the individual n-grams have been computed, we can calculate the cumulative BLEU score. This value can go from zero, i.e. two sentences are completely different, to one , i.e. both sentences are identical. We will attempt to obtain the lowest BLEU score possible because we want to obtain a sentence distinct from the original. In our research, we used the BLEU-2 and BLEU-3 metrics. The former computes the geometrical average of the 1-gram and 2-gram precisions, while the latter computes the geometrical average of the 1-gram, 2-gram, and 3-gram precisions. These metrics have been initialised with the weights shown in Table III.

TABLE III. Weights Used for the BLEU-2 y BLEU-3 Metrics

	1-gram	*2-grams*	*3-grams*
BLEU-2	0.25	0.25	0
BLEU-3	0.33	0.33	0.33

2. BERT

The BERT score [43] evaluates the semantic similarity between sentences. To do this, contextual embeddings are generated using BERT to represent the tokens in both the original and candidate sentences. Tokens are then compared using the cosine similarity. In the BERT score computation, *precision* and *recall* are calculated based on this comparison. Precision is determined by the proportion of tokens in the candidate sentence with a high cosine similarity with any token in the original sentence. It measures the relevancy of the generated tokens to the original sentence. On the other hand, recall represents the proportion of tokens in the original sentence with a high cosine similarity with any token in the candidate sentence. It measures the coverage of the generated tokens compared to the original sentence.

Using both values, F1-score is computed. The F1-score provides an overall measure of the similarity between the meanings of the original and generated sentences. Its value is between 0 and 1, with 1 indicating the highest possible similarity. Thanks to this metric, we will be able to ensure that the paraphrased sentences maintain the meaning of the original utterance while adding variability to the robot's speech, which could hinder the interaction.

B. Evaluation of the User-adapted Semantic Description Generation Approach

In this evaluation, we measured the response times of the three modules of the user-adapted semantic description generation pipeline: the random topic generator, the semantic description generator, and the user-adapted text modification module. The evaluation process involved running the system 200 times. Each run started with a random topic generated by the topic generator. The semantic description generator then produced an arbitrary description related to the topic. This description was then adapted to the user using the text modification module. The response times of the pipeline were analyzed throughout the iterations.

C. Evaluation of the Models Used For Paraphrase Generation

This evaluation compared different models for their effectiveness in paraphrasing sentences. The evaluation was conducted separately for models trained in Spanish and English. A set of 539 sentences was used for the evaluation. These can range from having one word to 80. Table VIII in Appendix A shows an example of sentences extracted from the set used for evaluating our solution. When evaluating the models trained in English, the sentences have been translated first from Spanish and then back to this language after being paraphrased.

We tested the mT5 and BERT2BERT models to paraphrase sentences directly in Spanish. We fine-tuned the mT5 model ourselves (the HFT5 model). During the evaluation, we passed every sentence through both models and compared the paraphrased sentences generated by the models with the original sentences using the BLEU and BERT scores. We also measured the time required to obtain the paraphrased sentences. For this, the paraphrase pipeline returns, alongside the paraphrasing result, the timestamp at four points in the process: (i) when the paraphrase request is received; (ii) when the translation from Spanish to English is completed; (iii) when the model has returned the paraphrasing result; and (iv) after the paraphrased sentences have been translated back into Spanish. For this first test, there was no translation, so we only used the timestamps for points (ii) and (iii). In this evaluation, the models were deployed locally. Once all the sentences were paraphrased, we calculated the average values of the metrics and the response time.

Regarding the models trained to perform paraphrasing in English, we tested the PMO-T5, Parrot, PEGASUS, and GPT-3 models. We evaluated these models using the same sentences used to evaluate the models trained in Spanish. This means that, in this case, the sentences had to be translated from Spanish to English, and the paraphrase results had to be translated back into Spanish. For this evaluation, we used the DeepL translator web service. Because of this, we present two separate sets of measurements: (i) the BERT and BLEU scores for the original sentences after translating them into English and the sentences generated by the models before translating them back into Spanish (which demonstrates the performance of the models themselves, without the translation from Spanish to English and from English to Spanish); and (ii) the BERT and BLEU scores for all four models calculated by comparing the original sentence in Spanish and the generated sentence after translating it back into Spanish (the performance of the entire pipeline).

VII. Results

In this section, we discuss the results obtained for the evaluation of the User-adapted Semantic Description Generation approach and the Paraphrase generation models. These results include both the ones obtained using the metrics described in Section VI, as well as the inference time for each of the two contributions presented.

A. Results of the User-adapted Semantic Description Generation Approach

This section will cover the quantitative results of user-adapted semantic description generation. By analysing the response times of our pipeline, we evaluated the system iteratively to validate its performance. The results are shown in Table IV. Appendix B shows various user-adapted semantic description generation examples from the set used to evaluate our pipeline. We can see that the median response time of the entire pipeline is 4.87 seconds. Within the pipeline, the topic generation module is relatively fast, with a median time of 0.23 seconds, followed by the description generation module. Finally, the text adaptation module is the slowest, with a median time of 3.29 seconds.

TABLE IV. Inference Time Response Statistic Analysis for the User-adapted Semantic Description Generation Pipeline

	Random Topic Generation	Semantic Description Generation	User-adapted text modification	Complete Pipeline
Min (s)	0.20	0.36	0.82	1.46
Max (s)	12.64	13.16	8.42	16.89
Median (s)	**0.23**	**1.02**	**3.29**	**4.87**

Regarding the user-adapted text modification module, we used the metrics described in subsection VI.A to evaluate the ability of the model to maintain the original content of the text; however, it should be noted that our aim in developing the user adaptation module is not to remain faithful to the text itself but rather to ensure that the end user understands the text. As in evaluating response times, we performed 200 iterations with the module configured to interact with an older adult using the prompts shown in Section III to analyse its performance. The average value of the BERT score is 0.76; this value suggests that we are not losing the main ideas and intentions of the original texts. On the other hand, for the BLEU scores, we obtain averages of 0.33 for BLEU-2 and 0.28 for BLEU-3; these are low scores overall, which may mean that the text has different sentence structures and words in more complete adaptations.

Several videos in which the application is used have been recorded to demonstrate its use. To make the part of the text being adapted to the user easier to perceive, the topic that was chosen is wine; in the first video, the generated text is shown without adaptation[11]. The second video and the third video show, respectively, the adapted text for an elderly person[12] and a child[13].

B. Results of the Models Used For Paraphrase Generation

As covered in section VI, we have compared the different models in Spanish and English. Regarding the Spanish models, the results, shown in Table V show that the mT5 and HFT5 models obtained similar BLEU-2/3 and BERT values (0.76/0.74 and 0.77, respectively), while the difference between these values is higher for the BERT2BERT model (0.62 and 0.50/0.43). Regarding the inference time, mT5 and BERT2BERT were able to generate new sentences in under two seconds (0.88 s for mT5 and 1.42 s for BERT2BERT). For HFT5, the time required to obtain a prediction averaged 4.04 seconds, which is significantly slower than the inference time of the pretrained mT5 model, although their BERT and BLEU-2/3 scores were similar.

TABLE V. Evaluation Results for the Models Fine-Tuned for Paraphrasing Sentences in Spanish

	BERT	BLEU-2	BLEU-3	local t (s)
mT5	0.77	0.76	0.74	0.88
HFT5	0.79	0.78	0.76	4.04
BERT2BERT	0.62	0.50	0.43	1.42

When we manually evaluated the paraphrase generation results in Spanish, we observed that the sentences generated by the mT5 model were either identical to the original sentences or lost their original meaning. The latter problem was also observed in the sentences generated by the BERT2BERT model. We also observed that the mT5 model truncated sentences greater than a certain length. These issues also appeared in the sentences generated with t he HFT5 model. Finally, during this manual review, we observed that there were cases in which the paraphrased sentences present objectively good results, as they maintain the meaning of the original sentence while changing how it is written, but they might not make complete sense or might be phrased in a way that will sound weird to users.

As far of the models trained to perform paraphrasing in English, the results, shown in Table VI, give an idea of how good the selected paraphrasing models are and how good the proposed translator-paraphrase-translator architecture is for our application (paraphrasing sentences in Spanish).

When we analyse just the paraphrase step, the results obtained are very similar for the Parrot and PEGASUS models, which have BERT scores that are higher than their BLEU-2/3 scores (0.69 and 0.44/0.38 for the former, 0.66 and 0.32/0.37 for the latter). On the other hand, we saw an increase in all the metrics for the PMO-T5 model, which has high BERT and BLEU-2/3 scores (0.88 and 0.76/0.7, respectively). Finally, the GPT-3 model scores (a BERT score of 0.74 and a BLEU-2/3 score of 0.55/0.48) were between those obtained for the PMO-T5 model and those obtained for the Parrot and PEGASUS models. When we add the translation steps before and after paraphrase generation, we can see a similar increase in all metrics. Finally, we compare the times required

[11] https://youtube.com/shorts/E7azQgY4HD8?feature=share

[12] https://youtube.com/shorts/mRUOn1MBzuQ

[13] https://youtube.com/shorts/Vy3_n-VBITM

TABLE VII. Examples of Failed Paraphrases When Using Models Trained in Spanish

	Spanish sentence	English translation
Original	¿Quieres continuar con el juego?	Do you want to continue with the game?
Paraphrased	En el juego, ¿Quieres continuar con el juego?	In the game, do you want to continue with the game?
Original	Me llamo Mini y soy un robot social.	My name is Mini, and I am a social robot.
Paraphrased	Me llamo Mini y soy un robot social.	My name is Mini, and I am a social robot.
Original	¿Cómo te llamas?	What is your name?
Paraphrased	¿Cómo se llama la llama?	What is the flame's name?
Original	Me llamo Mini y soy un robot social.	My name is Mini, and I am a social robot.
Paraphrased	Me llamaron Mini y somos una mente social.	They called me Mini, and we are a social mind.

to obtain a paraphrased sentence. We see that the PMO-T5 and GPT-3 models show the best results, both when they are run locally (2.04 s for PMO-T5, 2.01 s for GPT-3) and when they are run on the external server (1.61 s for PMO-T5, 1.59 s for GPT-3), compared with the Parrot (3.16 s when run locally, 2.27 s when run on the server) and PEGASUS (3.46 s when run locally, 2.05 s when run on the server) models. If we evaluate the models individually (without taking the translation steps into account), we see that the PMO-T5 model ran faster than GPT-3 (0.85 s/0.61 s for PMO-T5 and 1.06 s/0.74 s for GPT-3 when they are run locally / on the server), while Parrot proved to be the slowest (2.76 s when run locally and 1.2 s when run on the server).

TABLE VI. BERT and BLEU Scores, and the Inference Time for Local and Remote Execution, for the Models Trained in English When Evaluating Only the Paraphrase (Paraph), and When Evaluating the Entire Pipeline (Trans-Paraph-Trans). The Highest BERT Score and the Lowest BLEU Score and Inference Times Have Been Highlighted in Bold for Evaluations That only Consider the Paraphrase and for Evaluations That Consider the Entire Pipeline

	BERT	BLEU-2	BLEU-3	local t (s)	remote t (s)
PMO-T5 paraph	0.88	0.76	0.7	0.85	0.61
PMO-T5 trans-paraph-trans	0.77	0.56	0.46	2.04	1.61
Parrot paraph	0.69	0.44	0.38	2.76	1.2
Parrot trans-paraph-trans	0.5	0.31	0.25	3.16	2.27
PEGASUS paraph	0.66	0.44	0.37	2.57	1.12
PEGASUS trans-paraph-trans	0.51	0.32	0.25	3.46	2.05
GPT-3 paraph	0.74	0.55	0.48	1.06	0.74
GPT-3 trans-paraph-trans	0.56	0.39	0.31	2.01	1.59

Finally, as a proof of concept for the paraphrase module, we used one of Mini's applications: telling stories to the user. We chose one of the stories and recorded one video in which the robot tells the story as is[14] and another in which the robot paraphrases the story before telling it[15]. This is done by passing the sentences in the story through the paraphrase pipeline one by one.

VIII. Discussion

The results of the Spanish paraphrasing models indicated that mT5 had high BLEU and BERT scores, meaning similar meanings to the originals but with limited wording variation. In contrast, BERT2BERT produced different sentences that lost some original meaning. In the case of the English paraphrasing models, PMO-T5 had the highest scores for both metrics, sacrificing some original meaning for more diverse sentences. There was a trade-off between semantic and text similarity, and the Spanish paraphrased sentences were generally of lower quality than English, as shown in Table VII. Therefore, we decided to focus on those modules finetuned for paraphrasing sentences in English.

If we focus on the adaptation that the UASDG pipeline performs, on top of the text and semantic similarities, there is a third factor that also plays a role: how well the paraphrased text takes into account the profile of the user interacting with Mini (if the user is a child or an older adult). The analysis of the results obtained by computing the BERT and BLEU-2/3 scores indicates that the method maintains semantic relevance in the context of both texts. In contrast, the modified text does not bear much resemblance to the original, which reflects the effort made by the model to adapt the text so that it can be better understood by the user. A possible reason for these results, compared to those observed for the paraphrase module, is that the text has to be adapted to different audiences, and this introduces a certain level of variability. User-adapted text modification tends to highlight the main concepts found in the original text, leading to the omission of things that may be too complex for the user or not important for understanding the original topic, as shown in Table IX, Appendix B. There must be a trade-off between omitting complex elements and not undermining the understanding of the text.

Conversely, focusing on the entire UASDG pipeline, the ability to adapt to the user's profile and the possibility of autonomously selecting conversational topics may enhance the perceived intelligence and naturalness of the robot, thus improving its interactions. Nevertheless, there are still some challenges to overcome in full integration. In general, integrating generative language models in this scope gives us flexibility and provides creativity to some extent without losing naturalness. The correct design of the prompt in each module allowed us to correctly match the expected performance in the preliminary results. Although the initial objectives have been met, despite the fact that we made several templates adjusted to the profiles shown in the examples, a dynamic adaptation should also be created to not restrict the possible user profiles [24]. Our pipeline can generate semantically rich text efficiently. Topic generation has proven to be a useful tool providing the system with spontaneity and creativity. On the other hand, the text adapted to the user shows high grammatical malleability without losing the semantics of the original text, which helps the message reach the user in optimal shape.

[14] https://youtube.com/shorts/rERpBROzhtw?feature=share

[15] https://youtube.com/shorts/WGWZ4NN6fz8?feature=share

Even though the quality of the text generated by the models we use in our applications is a key aspect that can be used to assess the usability of the modules presented in this manuscript, another factor must be taken into account. These models will be integrated into a robot designed for human-robot interaction. As we mentioned in Section VII, messages conveyed by a participant in a conversation can lose their meaning if they are delivered with an extreme delay. Some studies set the maximum delay between interaction turns to 2 seconds [16], although other works contend that this time should be lower (around 1 second) [17]. Because of this, it is important to consider the inference time when deciding whether a model can be integrated into our architecture. The HFT5 model was unfit for real interactions using the paraphrase module, as its inference time is above two seconds. While the BERT2BERT model does meet the two-second threshold, its inference time is still too close to this threshold, meaning that the rest of the robot's modules involved in conveying responses to the user would have to perform their tasks in under 0.6 seconds for the total response time to be under 2 seconds. Finally, the pretrained version of mT5 is the only model trained in Spanish that could perform at the speed required in real interactions. For the models trained in English, only the PMO-T5 and GPT-3 models can perform below the selected threshold, and they can do this only when they are deployed on the external server. However, in both cases, the mT5 model in Spanish has the same issue (the total time is too close to the limit). Here, it is important to mention two things. First, the measured times do not consider the delay introduced by the communication between the robot and the server. Second, the inference time was obtained by averaging the time required to paraphrase the entire list of sentences used by Mini. However, some of the sentences used were significantly longer than the rest, increasing this average value. Most of the sentences used by Mini in common situations are shorter, and thus the time required to paraphrase them will be lower.

When it comes to UASDG response times, on the other hand, we found relatively longer overall times for the entire pipeline. The topic generation module is faster than the rest of the modules because it generates a single term and has a short prompt, which means that the model can work with a smaller amount of text. On the other hand, the slowest module in the pipeline is the module that has to handle the largest amount of text, which is the user-adapted modification module; however, there is a key difference between the UASDG and paraphrase modules. The UASDG functionality will be part of a robot's skills, which means it will not be part of every interaction between Mini and the user. Additionally, this module is not used to respond to the user's inputs, which softens the time requirements. Additionally, in the case of excessively long waiting times during the execution of the pipelines, we have deterrent techniques with utterances for the robot to use to fill these gaps without affecting the interaction. For these reasons, making UASDG follow the two-second rule is not as critical as ensuring that the paraphrase module follows this rule.

Finally, while the results observed are encouraging, a series of limitations must be addressed. One of the main limitations, which is due to the large sizes of these models, is the computational capacity required for training and inference and its related costs. In the case of the text generation module in the UASDG pipeline, because it is a large decoder-based model, its use leads to a higher latency in the inference that, when implemented with social robots, can affect its immediacy and thus the naturalness and fluidity of the interaction. We were able to mitigate this limitation by training and deploying our models on an external server, but these tasks can still be challenging. A second limitation connected to the selected models is that some of them (like GPT-3) are proprietary models, which limits the level of access that we have to them. Regarding the evaluation of the proposed modules, we decided to focus on objective evaluations, as they can help us determine if a particular model can or cannot be integrated into our architecture, and they give us a good idea of how these models are going to perform. However, sometimes the perception that the user has of a robot does not coincide with the results provided by objective metrics. For example, while the BERT score might indicate that Mini's dialogues are losing part of their meaning after going through the paraphrase module, this may not be an issue for the user, and the interaction might still be satisfactory. Thus, conducting a subjective evaluation of the modules presented in this manuscript would be useful. Also, another limitation of the evaluation of the paraphrase generation module is that the sentences generated by the models fine-tuned for paraphrasing sentences in English were evaluated by Spanish native speakers, which could have affected their perception of the appropriateness of these sentences. Finally, some technical limitations related to the paraphrase module must be mentioned. The first one is connected to the format that the paraphrase module expects the input sentence to have. Text-to-speech modules used in robotics can provide special commands for modifying how a sentence is uttered (for example, introducing pauses into the speech or altering the prosodic features of the voice) or for introducing non-verbal sounds (like a laugh or a yawn). However, the proposed method for paraphrasing sentences does not allow these commands. Thus, if this module has to be used with a TTS module that allows these commands, it would be necessary to remove the commands before sending the sentence to the paraphrase module and then put them back once the output sentence is received. Finally, one last limitation that has to be considered is that paraphrasing sentences has a chance of resulting in text that makes no sense, which could hinder interactions (although our results show that this is not common). Regardless of these limitations, the results obtained by evaluating the integration of the NLP applications presented in this manuscript into Mini's architecture indicate that our work was completed successfully.

IX. Conclusions

In this work, we have presented how language models can enhance human-robot interactions. In particular, we have addressed two problems. First, we implemented a mechanism that allows robots to talk about topics that have not been considered beforehand. To this end, we used the GPT-3 model to generate an appropriate topic of conversation and then to obtain relevant information about this topic. Moreover, the received information needs to be adapted to the person the robot is talking to. Thus, the robot adapts the conversation to the profile of the user. With this mechanism, when the robot is, for example, interacting with a child, it uses language that is not technical so that the child can understand it.

Second, when interacting with robots that use predefined utterances, the user might perceive the robots as repetitive and monotonous. To mitigate this issue, we have integrated different language models for paraphrasing predefined texts written in Spanish. The results have shown a trade-off between the variety we can introduce in the text and the amount of meaning that is lost in the process. Additionally, when English-based models are used, English-Spanish translations produce significantly more variability than the direct use of Spanish-based models.

Both mechanisms have been integrated into our social robot, Mini, considering the fact that additional interaction delays might reduce the interaction quality. While the results obtained are encouraging, there are still some limitations that should be tackled in future work. These limitations include the computational power required to run some of the larger language models, the latency that these modules introduce in interactions, and the lack of control over proprietary language models (like GPT-3). Regardless, the results point towards the advantages that integrating transformer-based NLP solutions can provide for the interaction capabilities of social robots.

Appendix

A. Examples of Sentences Used for the Evaluation of the Paraphrase Pipeline

TABLE VIII. Examples of Paraphrases When Using Models Trained in Spanish, With Their Translation to English

Spanish original sentence	English translation
¿Cúanto es 9 menos 4?	What is the result of 9 minus 4?
¿En qué ciudad se encuentra esta torre?	In which city is this tower located?
¡Empezamos!	Let's start!
Muéstrame una tarjeta con un objeto verde.	Show me a card with a green object.
Vamos con una fácil para empezar.	Let's start with an easy one.
Claro, a mi las noticias a veces me aburren.	Sure, I sometimes get bored of the news.
¿Quieres elegir otro cartón?	Do you want to choose another card?
Podemos repetirlo en otro momento.	We can repeat this another time.
Los árboles eran el pino, el abeto, el roble y el sauce.	The trees were the pine, the fir, the oak, and the willow.
¡Muy bien!	Very good!
Para saber la respuesta dividimos 12 entre 3. El resultado es 4 lápices por persona.	In order to find the answer, we divide 12 by 3. The result is 4 pencils per person.
Se trata de la catedral de Zamora. Pero qué bonita es esta ciudad.	It is the cathedral in Zamora. Oh, how beautiful this city is.
¿Cuáles de estas palabras son deportes?	Which of these words are sports?
Esta palabra es algo que se encuentra en un baño.	This word is something that you can find in a bathroom.
¿Cuánto da si resto 8 a 20?	What is the result of subtracting 8 to 20?
¿Estás seguro de que no tienes línea?, mira bien tu cartón. Seguimos para bingo.	Are you sure that you don't have a line? Check your card again.
Por favor, muéstrame un hexágono.	Please, show me an hexagon.
Juan va al mercado. La carne le cuesta 12 euros, y paga con un billete de 20. ¿Cuánto le tienen que devolver?	Juan goes to the market. Meat costs 12 euros, and he pays with a 20 euro bill. How much change is he getting?
Acuérdate de contestar usando el micrófono.	Remember to answer using the microphone.
¿De qué animal se trata?	Which animal is this?
¿Qué palabra de la pantalla está relacionada con agua?	Which word in the screen is related to water?
La respuesta era casa, mochila, alfombra y pelota.	The answer was house, backpack, carpet, and ball.
En este juego, te voy a ir enseñando objetos, y luego tú, tienes que pulsar en la persona de la pantalla que utiliza ese objeto para su profesión.	In this game, I will show different objects to you, and you have to select among the people in the screen the one that uses that object in their work.
La solución es plátano.	The answer is bannana.
Este ejercicio es para ver cómo de bien conoces la ciudad de Zamora. Yo te voy a ir enseñando edificios conocidos de la ciudad y tú me tienes que decir cómo se llaman.	This exercise aims at evaluating how well do you know the city of Zamora. I will show you known buildings in the city, and you have to tell me their name.
La solución era mochila.	The answer was backpack.
En la imagen había 1 euro y 14 céntimos.	The image showed 1 euro and 14 cents.
En este ejercicio, voy a ir mostrando colores por el corazón y tú tienes que seleccionar en la pantalla, el objeto que sea del mismo color.	In this exercise, I will light my heart in different colours, and you have to select on the screen the object that is the same colour.
¿Qué edificio es el que muestro ahora?	Which building am I showing now?
En este juego, voy a ir poniendo diferentes fotos de comida y me tienes que decir a qué zona de España pertenece. Para que no sea my difícil, te voy a dar a elegir entre tres opciones. A ver qué tal se te da.	In this game, I will show you different pictures of food, and you have to tell the region in Spain that food belongs to. Just so it is not that hard, I will give you three options to choose from. Let's see how you do.
¿Cuáles son las palabras que te he dicho? Dímelas por el micrófono de una en una.	What are the words that I just said? Say them through the microphone one by one.
¡Bravo! Cómo me gusta esta canción. Espero que a ti también. Bueno, vamos a seguir con más ejercicios.	Bravo! Oh, how I love this song. I hope you like it too. Well, let's continue with more exercises.
Contesta cuando escuches el pitido.	Answer after you hear the beep
Parece que no nos estamos entendiendo, debe de haber algún problema.	It seems that we are not understanding each other, there must be a problem.

B. Examples of Generations Used for the Evaluation of the UASDG Pipeline

TABLE IX. Examples of Semantics Description Generations Used for Evaluation. It Is Worth Noting How User Adaptation Is Able to Synthesise Long and Detailed Texts to Improve the End-user's Understanding

Topic	Semantic Description	Elder Adaptation	Child Adaptation
Glasses	Glasses are a type of optical device that is worn on the eyes to correct vision. They can be used for reading, watching television, or computer use. Glasses are also called spectacles and they come in many different types and styles. The most common types of glasses include bifocals (which have two lenses), trifocals (which have three lenses) and progressives (which allow you to see objects at progressively closer distances)	Glasses are something you wear on your eyes to help you see better. They come in many different kinds and styles, and can be used for different things like reading, watching TV, or using a computer. The most common types are called bifocals (with two lenses), trifocals (with three lenses), and progressives (which help you see things up close).	Glasses are like special helpers for your eyes. They help you see things better, like when you read a book or look at a computer. There are many different kinds of glasses, but the most common ones have two or three lenses. These help you see things up close or from far away.
Pizza	Pizza is a flat bread that originated in Italy. It is usually topped with tomato sauce, cheese and various meats such as ham, sausage or bacon. The word pizza comes from the Italian word pizze which means “pie” (as in pie crust)	Pizza is a type of food that started in Italy. It's like a flat round cake with tomato sauce, cheese, and different meats like ham, sausage, or bacon on top. The word “pizza” comes from the Italian word “pizze” which means a type of pastry.	Pizza is a type of food that started in Italy. It's a flat bread with yummy things on top like sauce, cheese, and meats like ham or sausage. The name “pizza” comes from a word in Italian that means “pie”.
The Great Pyramid	The Great Pyramid of Giza is the oldest and largest of the three pyramids in the Giza Necropolis bordering what is now El-Gizah, Egypt. It was built as a tomb for Fourth Dynasty Egyptian pharaoh Khufu (Cheops) around 2560 BC or earlier during the Old Kingdom period. The pyramid, originally named Khufu's Pyramid but known to its builders as ¨the Great Pyramid¨, has an area of about 230,000 m^2 (23 acres), though it is not quite square. It is made from 2.5 million blocks ranging in size from 1 to 10 tons each. Originally covered with casing stones that formed a smooth outer surface, some ancient sources claim that this covering was removed at some point in antiquity; however it remains one of only two large stone structures ever built by humans without any metal tools, It stands 648 feet tall and contains over 100 chambers on its four sides: King's Chamber – where pharaohs were buried – Queen's Chamber – where queens were buried, Grand Gallery - which connects King's chamber with Queen's chamber via an underground passage called “Queen's tunnel”, and Subterranean Chambers - which are located under all other chambers except for Kings' chamber	The Great Pyramid of Giza is a very old building in Egypt. It was made a long, long time ago, around 2560 BC. It was built for a king named Khufu and was used as a place to keep his body after he died. The pyramid is very big and was made with big stones that weigh a lot. It's 648 feet tall and has many rooms inside, including a room for the king and one for the queen. There's also a big hallway and some rooms under the ground.	The Great Pyramid of Giza is a really old and big building in Egypt. It was made a long, long time ago to be a tomb for a king named Khufu. It's made of lots and lots of big blocks of stone, and it's so big that it's like an entire town! It's also very tall, almost as tall as a big building with 20 floors! Inside the pyramid, there are many rooms and chambers where the king and queen were buried. There's even a secret underground passage that connects the rooms!

Acknowledgement

The research leading to these results has received funding from the projects: Robots sociales para mitigar la soledad y el aislamiento en mayores (SOROLI), PID2021-123941OA-I00, funded by Agencia Estatal de Investigación (AEI), Spanish Ministerio de Ciencia e Innovación. Robots sociales para reducir la brecha digital de las personas mayores (SoRoGap), TED2021-132079B-I00, funded by Agencia Estatal de Investigación (AEI), Spanish Ministerio de Ciencia e Innovación. This publication is part of the R&D&I project PLEC2021-007819 funded by MCIN/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR. This work has been supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with UC3M (“Fostering Young Doctors Research”, SMM4HRI-CM-UC3M), and in the context of the V PRICIT (Research and Technological Innovation Regional Programme).

References

[1] L. Laranjo, A. G. Dunn, H. L. Tong, A. B. Kocaballi, J. Chen, R. Bashir, D. Surian, B. Gallego, F. Magrabi, A. Y. S. Lau, E. Coiera, “Conversational agents in healthcare: a systematic review,” *Journal of the American Medical Informatics Association*, vol. 25, pp. 1248–1258, Sept. 2018.

[2] J. Cassell, J. Sullivan, E. Churchill, S. Prevost, *Embodied Conversational Agents*. MIT Press, 2000.

[3] L. Clark, N. Pantidi, O. Cooney, P. Doyle, D. Garaialde, J. Edwards, B. Spillane, E. Gilmartin, C. Murad, C. Munteanu, V. Wade, B. R. Cowan, “What Makes a Good Conversation? Challenges in Designing Truly Conversational Agents,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, May 2019, pp. 1–12.

[4] V. Klingspor, Y. Demiris, M. Kaiser, “Human- robot-communication and machine learning,” *Applied Artificial Intelligence*, vol. 11, 03 1999.

[5] C. Clavel, Z. Callejas, “Sentiment Analysis: From Opinion Mining to Human-Agent Interaction,” *IEEE Transactions on Affective Computing*, vol. 7, pp. 74–93, Jan. 2016.

[6] J. Woo, J. Botzheim, N. Kubota, “Conversation system for natural communication with robot partner,” in *2014 10th France-Japan/ 8th Europe-Asia Congress on Mecatronics (MECATRONICS2014- Tokyo)*, Nov. 2014.

[7] A. Fujita, A. Kameda, A. Kawazoe, Y. Miyao, “Overview of Todai robot project and evaluation framework of its NLP-based problem solving,” in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, May 2014, pp. 2590–2597, European Language Resources Association (ELRA).

[8] I. A. Hameed, “Using natural language processing (NLP) for designing socially intelligent robots,” in *2016 Joint IEEE International Conference on*

Development and Learning and Epigenetic Robotics (ICDL-EpiRob), Sept. 2016.

[9] T. Williams, "A consultant framework for natural language processing in integrated robot architectures," *IEEE Intelligent Informatics Bulletin*, vol. 18, pp. 10–14, 2017.

[10] W. Kahuttanaseth, A. Dressler, C. Netramai, "Commanding mobile robot movement based on natural language processing with RNN encoderdecoder," in *2018 5th International Conference on Business and Industrial Research (ICBIR)*, May 2018, pp. 161–166.

[11] W. Budiharto, V. Andreas, A. A. S. Gunawan, "Deep learning-based question answering system for intelligent humanoid robot," *Journal of Big Data*, vol. 7, p. 77, Dec. 2020, doi: 10.1186/s40537-020-00341-6.

[12] M. Seo, A. Kembhavi, A. Farhadi, H. Hajishirzi, "Bidirectional Attention Flow for Machine Comprehension," *arXiv:1611.01603 [cs]*, June 2018. arXiv: 1611.01603.

[13] S. Arroni, Y. Galán, X. Guzmán-Guzmán, E. R. Núñez-Valdez, A. Gómez, "Sentiment analysis and classification of hotel opinions in twitter with the transformer architecture," *International Journal of Interactive Multimedia and Artificial Intelligence, vol. 8, no. 1, pp. 53-63*, 2023.

[14] J. Zhou, T. Li, S. J. Fong, N. Dey, R. González-Crespo, "Exploring chatgpt's potential for consultation, recommendations and report diagnosis: Gastric cancer and gastroscopy reports' case," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, no. 2, pp. 7-13, 2023.

[15] M. Rheu, J. Y. Shin, W. Peng, J. Huh-Yoo, "Systematic Review: Trust-Building Factors and Implications for Conversational Agent Design," *International Journal of Human–Computer Interaction*, vol. 37, pp. 81–96, Jan. 2021.

[16] R. B. Miller, "Response time in man-computer conversational transactions," in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, 1968, pp. 267–277.

[17] T. Shiwa, T. Kanda, M. Imai, H. Ishiguro, N. Hagita, "How quickly should communication robots respond?," in *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2008, pp. 153–160, IEEE.

[18] R. R. Murphy, T. Nomura, A. Billard, J. L. Burke, "Human–Robot Interaction," *IEEE Robotics Automation Magazine*, vol. 17, pp. 85–89, June 2010, doi: 10.1109/MRA.2010.936953. Conference Name: IEEE Robotics Automation Magazine.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, Curran Associates, Inc.

[20] I. Sutskever, O. Vinyals, Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 27, 2014, Curran Associates, Inc.

[21] "The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time." [Online]. Available: https://jalammar.github.io/illustrated-transformer/.

[22] D. Rothman, *Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more.* Packt Publishing Ltd, Jan. 2021. Google-Books-ID: Cr0YEAAAQBAJ.

[23] A. M. P. Bras¸oveanu, R. Andonie, "Visualizing Transformers for NLP: A Brief Survey," in *2020 24th International Conference Information Visualisation (IV)*, Sept. 2020, pp. 270–279. ISSN: 2375-0138.

[24] K. Weiss, T. M. Khoshgoftaar, D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, p. 9, May 2016.

[25] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, "Improving Language Understanding by Generative Pre-Training," *OpenAI*, p. 12, 2018.

[26] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to- Text Transformer," *arXiv:1910.10683 [cs, stat]*, July 2020. arXiv: 1910.10683.

[27] L. Floridi, M. Chiriatti, "GPT-3: Its Nature, Scope, Limits, and Consequences," *Minds and Machines*, vol. 30, pp. 681–694, Dec. 2020, doi: 10.1007/s11023-020-09548-1.

[28] C. Stevenson, I. Smal, M. Baas, R. Grasman, H. van der Maas, "Putting GPT-3's Creativity to the (Alternative Uses) Test," in *International Conference on Innovative Computing and Cloud Computing*, 2022, arXiv. Version Number: 1.

[29] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, "Language Models are Few-Shot Learners," *ArXiv*, May 2020.

[30] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, W. Chen, "What makes good in-context examples for gpt-3?," *arXiv preprint arXiv:2101.06804*, 2021.

[31] G. Poesia, O. Polozov, V. Le, A. Tiwari, G. Soares, C. Meek, S. Gulwani, "Synchromesh: Reliable code generation from pre-trained language models," *arXiv preprint arXiv:2201.11227*, 2022.

[32] T. Goyal, J. J. Li, G. Durrett, "News summarization and evaluation in the era of gpt-3," *arXiv preprint arXiv:2209.12356*, 2022.

[33] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, *et al.*, "Exploring the limits of transfer learning with a unified text-to- text transformer.," *The Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[34] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 2020, pp. 38–45.

[35] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al- Rfou, A. Siddhant, A. Barua, C. Raffel, "mT5: A massively multilingual pre-trained text-to-text transformer," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, June 2021, pp. 483–498, Association for Computational Linguistics.

[36] Y. Yang, Y. Zhang, C. Tar, J. Baldridge, "PAWS- X: A cross-lingual adversarial dataset for paraphrase identification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, Nov. 2019, pp. 3687–3692, Association for Computational Linguistics.

[37] J. Zhang, Y. Zhao, M. Saleh, P. J. Liu, "Pegasus: Pre- training with extracted gap-sentences for abstractive summarization," in *Proceedings of the 37th International Conference on Machine Learning*, ICML'20, 2020, JMLR.org.

[38] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[39] J. Canete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, "Spanish pre-trained bert model and evaluation data," *Pml4dc at iclr*, vol. 2020, pp. 1–10, 2020.

[40] X. Zhao, B. F. Malle, "Spontaneous perspective taking toward robots: The unique impact of humanlike appearance," *Cognition*, vol. 224, p. 105076, July 2022, doi: 10.1016/j.cognition.2022.105076.

[41] M. A. Salichs, A. Castro, E. Salichs, E. Fernandez, M. Maroto, J. J. Gamboa, S. Marques, J. C. Castillo, F. Alonso, M. Malfaz, "Mini: A New Social Robot for the Elderly," *International Journal of Social Robotics*, vol. 12, pp. 1231–1249, Dec. 2020, doi: 10.1007/s12369-020-00687-0.

[42] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[43] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," in *Eighth International Conference on Learning Representations*, Apr. 2020.

Javier Sevilla Salcedo

Javier Sevilla Salcedo is a Ph.D. candidate and researcher at the Robotics Lab at the Carlos III University of Madrid. His academic journey began with a B.Sc. in Industrial Electronics and Automation Engineering from the University of Jaén. He furthered his education with a M.Sc. in Robotics and Automation at his current institution. His career includes over four years of extensive research in diverse areas such as Artificial Intelligence, Robotics, and NLP, which involved his work at two prominent research labs. Although Natural Language Processing remains his main focus, his research integrates it with Social Robotics, Deep Learning, Robot Perception, and Cognitive Robotics.

Enrique Fernández Rodicio

Enrique Fernández Rodicio received the B.Sc. degree in Industrial Engineering from the Carlos III University of Madrid in 2014, the M.Sc. degree in Robotics and Automation from the Carlos III University of Madrid, Spain, in 2016, and the Ph.D in Electric, Electronic, and Automation Engineering from the Carlos III University of Madrid in 2021. He is a research assistant at the RoboticsLab Research Group, inside the Department of Systems Engineering and Automation of the Carlos III University of Madrid, Madrid, Spain. His present research lines are related to human-robot interaction, dialogue management, and expressiveness management. Enrique Fernández Rodicio received the B.Sc. degree in Industrial Engineering from the Carlos III University of Madrid in 2014, the M.Sc. degree in Robotics and Automation from the Carlos III University of Madrid, Spain, in 2016, and the Ph.D in Electric, Electronic, and Automation Engineering from the Carlos III University of Madrid in 2021. He is a research assistant at the RoboticsLab Research Group, inside the Department of Systems Engineering and Automation of the Carlos III University of Madrid, Madrid, Spain. His present research lines are related to human-robot interaction, dialogue management, and expressiveness management.

Laura Martín Galván

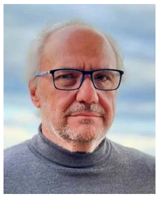
Laura Martín Galván received the BSc. degree in Electronic, Robotic, and Mechatronic Engineering from the University of Málaga, Spain, in 2020. She then received her MSc. in Robotics and Automation from the Carlos III University of Madrid in 2022. Since 2021, she has been involved in the Robotics Lab at the University Carlos III of Madrid, where her research interests have included Natural Language Processing with the use of Artificial Intelligence in Social Robots.

Álvaro Castro González

Álvaro Castro González received the B.Sc. degree in computer engineering from the University of León, León, Spain, in 2005, and the M.Sc. and Ph.D. degrees in robotics and automation from the Carlos III University of Madrid, Madrid, Spain, in 2008 and 2012, respectively. He is currently an Assistant Professor with the Department of Systems Engineering and Automation, Carlos III University of Madrid, and member of the Robotics Lab Research Group. He has been involved in several national, European, and corporate sponsored research projects. His research interests include human–robot interaction, social robots, expressiveness in robots, decision-making, and artificial emotions.

José Carlos Castillo

José Carlos Castillo is an assistant professor at the University Carlos III of Madrid. He obtained his Ph.D. in Computer Science from the University of Castilla-La Mancha, Spain, in 2012. From 2006 to 2012, he worked at the natural and artificial Interaction Systems group at the Albacete Research Institute of Informatics, Spain, working on computer vision techniques for detecting human activities and frameworks for intelligent monitoring and activity interpretation. From 2012 to 2013, he worked as a post-doctoral researcher at the Institute for Systems and Robotics (ISR), Instituto Superior Técnico (IST) of Lisbon, where he was involved in the development of networked robot systems, robotics and computer vision and intelligent control systems. Since September 2013, he has been combining teaching and research at the Robotics Lab at the University Carlos III of Madrid. He focuses on multimodal perception for Human–Robot Interaction for the mild cognitive impaired.



Miguel A. Salichs

Miguel A. Salichs is a full professor of the Systems Engineering and Automation Department at Carlos III University of Madrid (UC3M). He received the Electrical Engineering and Ph.D. degrees from Polytechnic University of Madrid. His research interests include autonomous social robots, multimodal human-robot interaction, mind models and cognitive architectures. He was Vicerrector of the UC3M, member of the Policy Committee of the International Federation of Automatic Control (IFAC), Chairman of the Technical Committee on Intelligent Autonomous Vehicles of IFAC, responsible of the Spanish National Research Program on Industrial Design and Production, President of the Spanish Society on Automation and Control (CEA), and the Spanish representative at the European Robotics Research Network (EURON). He is currently Coordinator of the Spanish Robotics Technology Platform (HispaRob), President of the Foundation of the Spanish Society on Automation and Control, and President of Area at the Spanish Research Agency.

Consensus-Based Learning for MAS: Definition, Implementation and Integration in IVEs

C. Carrascosa, F. Enguix, M. Rebollo, J. Rincon *

VRAIn - Valencian Research Institute for Artificial Intelligence, Universitat Politècnica de València, Valencia (Spain)

Received 23 February 2023 | Accepted 4 August 2023 | Published 24 August 2023

Abstract

One of the main advancements in distributed learning may be the idea behind Google's Federated Learning (FL) algorithm. It trains copies of artificial neural networks (ANN) in a distributed way and recombines the weights and biases obtained in a central server. Each unit maintains the privacy of the information since the training datasets are not shared. This idea perfectly fits a Multi-Agent System, where the units learning and sharing the model are agents. FL is a centralized approach, where a server is in charge of receiving, averaging and distributing back the models to the different units making the learning process. In this work, we propose a truly distributed learning process where all the agents have the same role in the system. We suggest using a consensus-based learning algorithm that we call Co-Learning. This process uses a consensus process to share the ANN models each agent learns using its private data and calculates the aggregated model. Co-Learning, as a consensus-based algorithm, calculates the average of the ANN models shared by the agents with their local neighbors. This iterative process converges to the averaged ANN model as a central server does. Apart from the definition of the Co-Learning algorithm, the paper presents its integration in SPADE agents, along with a framework called FIVE allowing to develop Intelligent Virtual Environments for SPADE agents. This framework has been used to test the execution of SPADE agents using Co-Learning algorithm in a simulation of an orange orchard field.

Keywords

Complex Networks, Distributed AI, Multiagent Systems, Neural Networks.

DOI: 10.9781/ijimai.2023.08.004

I. Introduction

This paper follows a research line related to multi-agent learning. So, it extends the work presented by Carrascosa et al. [1], where a new algorithm based on Federated Learning and Consensus in Multi-Agent Systems named CoL was presented. This extension focuses in how this kind of algorithms can be tested in execution in a close to real simulation using a new Intelligent Virtual Environment (IVE) generator.

Multi-agent Learning is currently a hot topic mixing machine learning with distributed systems. It can be found two main different kinds of such systems: the ones where the learning is a specific part of the system carried out by one (or a few) agents of the system (like in the work by Sánchez San Blas et al. [2]) where the deep learning is made by an agent in a complex system dedicated to the automatic detection of illegal swimming pools), and the ones where all agents make the same kind of deep learning process (that is, the learning also uses a distributed approach). In this last kind of system is where the proposed algorithm is classified. The proposed algorithm intends to get the most out of a distributed approach. It tries to mix the learned parameters in each agent with the parameters trained in its local neighbors without knowing the whole system. Moreover, this kind of learning algorithm preserves the privacy of the data used for the learning process by each agent in his local learning.

These features are, in some way, present in other approaches, mainly Federated Learning (FL). The FL algorithm was defined by Google [3]. The main idea behind this algorithm is to take advantage of distributed learning and maintain the privacy of the data used by each node in the learning process. The algorithm uses two different kinds of agents: server and client. The server defines the training model and sends it to all the clients. Then, clients train with their private data and send the model back to the server. Finally, the server aggregates all the models, for example, calculating an averaged one. This global model is sent back to execute the next training iteration. Kairouz et al. [4] analyze deeply the open challenges related to FL algorithms. It should be emphasized that the connection topology among the agents significantly influences the convergence rate in decentralized distributed learning processes. Nonetheless, the FL approach has noteworthy characteristics worth considering. Firstly, it maintains a distributed nature while operating with a centralized framework, implying that the system synchronizes and evolves based on the pace of the slowest agent. Furthermore, it lacks fault tolerance, rendering it vulnerable in scenarios where agents fail to respond or vanish, and it does not accommodate the incorporation of new agents during execution.

These features are of great importance when developing systems that must work in environments with a high probability of communication

* Corresponding author.

E-mail addresses: carrasco@dsic.upv.es (C. Carrascosa), fraenan@inf.upv.es (F. Enguix), mrebollo@dsic.upv.es (M. Rebollo), jrincon@dsic.upv.es (J. Rincon).

failure, where agents communicate sporadically, or when they must deal with disconnection periods to save battery. This situation appears in rural areas, characterized by limited connectivity and where the system may remain isolated without supervision for extended periods. These features can be obtained if, instead of using a centralized approach, a fully distributed one is used, as is the one followed by a consensus algorithm according to Olfati-Saber and Murray [5].

This paper presents a *consensus-based learning algorithm* coined Co-Learning or CoL, trying to take advantage of a completely decentralized approach for an FL-like learning algorithm. Along with presenting the *CoL* algorithm definition and description, an actual implementation using SPADE agents [6] is provided.

SPADE (Smart Python Agent Development Environment) [6] is a framework for developing intelligent agents in Python. The communication layer uses XMPP (Extensible Messaging and Presence Protocol)[1] as an instant messaging protocol.

This platform has been used in different areas, especially in IoT [7]. The CoL implementation in SPADE takes advantage of the *Presence* feature of the XMPP so that it can detect when a neighbor agent decides not to go on being connected or fails its connection, not having to wait for a deadline to acknowledge those failures. There is also some previous work in implementing a *pure* FL algorithm in *SPADE* agents, called *FLaMAS* [8].

In multi-agent systems, communication between agents is essential, and SPADE agents have an integrated message dispatcher that allows communication between them.

The SPADE agent model is based on behaviors. They are tasks that repeat upon a particular time pattern: one-shot, periodic, finite state machines (FSM) or even BDI (Belief Desire Intention) [9] behaviors, which allows reactive and deliberative capabilities in the agent.

The paper also presents a new Intelligent Virtual Environment (IVE) [10] generator, developed to test SPADE agents in a close-to-real-world scenario before deployment. Graphical simulations have always been a way of testing and validating applications (like in the work by Ikidid et al. [11]) where a simulation in ANYLOGIC is used to validate a model to control and fluidize vehicle traffic in a multi-intersection network). Checking qualitatively if a simulation seems to work correctly can save hours of work analyzing boring tables of numbers. The main problem with these simulations is that they usually cost a lot to build or even tune for a specific algorithm.

There is no novelty in proposing just another simulation framework, even if discussing a simulated environment, simply to deploy a Multi-Agent System. Traditionally, simulators that include agency concepts simulate the environment and the agents. That is the case, for instance, of *Netlogo* [12], where agents inhabit a matrix-like environment formed by patches. However, this simulator is limited to four different types of agents, the simulated environment is two-dimensional and does not allow the decoupling of its parts. The configuration of this monolithic system is produced in the same file.

On the other hand, it may be found what is called an IVE (Intelligent Virtual Environment) [10] that is, a virtual environment simulating a real world, inhabited by intelligent agents who may interact and whose behavior can be easily validated.

JaCalIVE (Jason Cartago implemented Intelligent Virtual Environment) [13] can be seen as an example of a framework for developing MAS inhabiting an IVE. This framework is based on MAM5 meta-model [14]. The idea behind it is to define a simulation through such a meta-model, which is compiled into some templates of Jason agents [15] and CaRTago artifacts [16] to be completed by the simulation developer. This framework has a very formal development process, but it is difficult to develop a new simulation or even make changes to an existing one.

It can also be found *MASON* (Multi-Agent Simulator Of Neighborhoods) [17], made purely in the Java programming language and released in 2003. This simulator is mainly oriented toward swarm intelligence and multi-agent systems. In addition, it allows you to choose a discrete or continuous space in the simulations and visualize the result in a two or three-dimensional space. However, achieving a 3D visualization in this simulator is not easy or fast and requires the additional installation of the Java3D libraries and knowledge of Java programming.

Differently, the proposal presented in this paper looks for an easy way of defining and modifying an IVE. This IVE will be developed in Unity[2], and agents will be SPADE agents [6]. This Simulation framework, named *FIVE* [18], allows us to define the environment and incorporate the algorithms to be validated into SPADE agents inhabiting such an environment.

The rest of the paper is structured as follows: in Section II the Co-Learning algorithm is presented. Next, in Section III the implementation of this algorithm in SPADE agents is shown. After that, in Section IV new FIVE framework is presented as a way of easy and fast creation and modification of Intelligent Virtual Environments to test Co-Learning SPADE agents, followed by Section V, where a case study with a virtual environment simulating an orange orchard is presented. The paper ends with some conclusions in Section VI.

II. Co-Learning (CoL) Algorithm

This section presents the model that supports the distributed training of the machine learning model, combined with the consensus process to average the parameters learned by the agents. An interaction topology delimits the ability of the agents to communicate and exchange information.

A. Consensus-Based Multi-Agent Systems

Olfati–Saber and Murray [5] define a consensus process in a Multi-Agent System (MAS) as a problem where the agents reach an agreement about the value of a variable of interest without any intermediate or leader that rules the process. It is an iterative procedure. The agent a_i calculates the new value $x_i(t+1)$ in each iteration, according to Equation (1).

$$x_i(t+1) = x_i(t) + \varepsilon \sum_{a_j \in N_i} \left[x_j(t) - x_i(t)\right] \tag{1}$$

where N_i denotes the neighbors of agent a_i and ε is the learning step: a factor bounded by the maximum degree of the network. The consensus converges to the average of the initial values $\langle x_i(0)\rangle$ whenever $\varepsilon \leqslant \frac{1}{\max d_i}$. This algorithm has been the base for different and multiple applications and other algorithms as, for instance, *Supportive Consensus* [19].

Fig. 1 depicts an example of the evolution of consensus over one of the weights over this simple synthetic network with four agents and initial values $x(0) = \{0.2, 0.4, 0.6, 0.8\}$. The convergence value is $\langle x(0)\rangle = 0.5$.

B. Consensus in Federated Learning

One of the approaches of FL consists of a set of clients that learns the weights of an artificial neural network (ANN) and shares them with a central server, which averages the weights to obtain a global model. Without losing generality, we can consider each weight as an independent variable and execute the consensus process in parallel over all the weights simultaneously.

[1] https://xmpp.org/

[2] http://unity.com

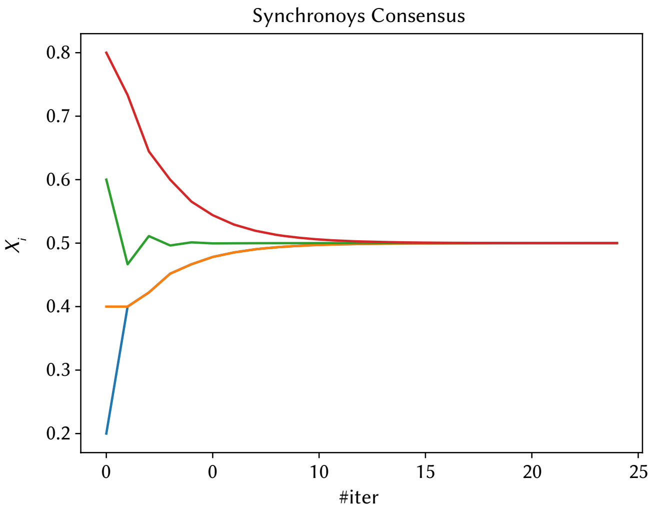


Fig. 1. Consensus evolution in a network with four agents. Initially, $x(0) = \{0.2, 0.4, 0.6, 0.8\}$, so $\langle x(0) \rangle = 0.5$.

Let us define a set of n identical agents A; each one implements identical ANN structures (same blocks, layers, and neurons). The goal is to learn a global model (W, tr) with a set of weights W for a training dataset tr. As the model is common, agents need only to share the set of weights W. The training dataset is divided into n fragments of the same length. The extrapolation of this approach to non-independent and identically distributed (non-IID) datasets is direct by using a weighted consensus variation [20].

The communication among the set of agents is constrained by a topology modeled by an undirected network $G = (A, E)$, where the nodes are the agents of set A. The set of edges E formed by pairs (a_i, a_j), denoting that agent a_i is connected with agent a_j. The neighborhood of agent a_i is denoted with $N_i = \{a_j \in A : (a_i, a_j) \in E\}$.

Each agent keeps an ANN (W_i, tr_i), being W_i a set of weights and biases for each layer of the ANN.

$$W_i = (W_{i,1}, \dots, W_{i,k}) \tag{2}$$

where $W_{i,j} \in \mathbb{R}^{n,m}$ represents the weights (or the bias) learned by agent a_i for the layer j of its ANN. Without losing generality, we can assume that the parameters of the ANN can be reshaped into a conforming representation.

The process follows the adapted Equation (3).

$$W_i(t+1) = W_i(t) + \varepsilon \sum_{a_j \in N_i} [W_j(t) - W_i(t)] \tag{3}$$

C. Algorithm Description

The *Consensus-based Learning Algorithm*, named either *Co-Learning* or *CoL* Algorithm can be described as a set of identical agents learning a model through an ANN, where all the agents share the same ANN structure. This allows sharing the model being learned by each agent with its local neighbors and making a consensus of such model based on the Equation (1). This model is formed by the weights matrices result of the training of the learning process -Equation (2)-. This consensual model is then used for each agent in the next training. An agent a_i following the Co-Learning algorithm (Algorithm 1) first of all will make e epochs of training the algorithm. The result of this training is the set of k matrices at Equation (2), and for each one of them, the next c iterations of the consensus algorithm, following the Equation (1) are made, leading to k new matrices that will be used in the training process again.

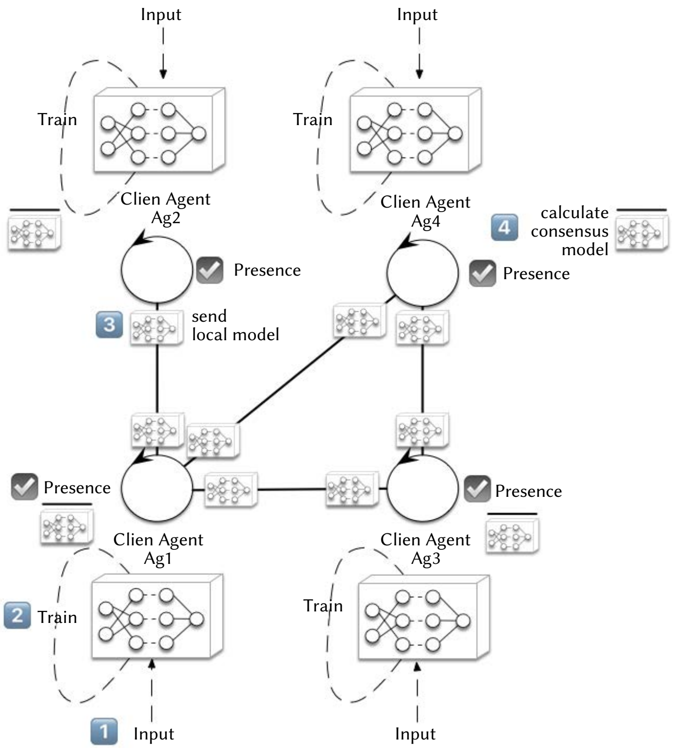


Fig. 2. Scheme of four SPADE agents doing a CoL Algorithm.

The process is executed in parallel as many times as parameters the ANN has. It can be considered a vectorized version of the evolution seen in Fig. 1.

Algorithm 1. Co-Learning (CoL) Algorithm for agent a_i

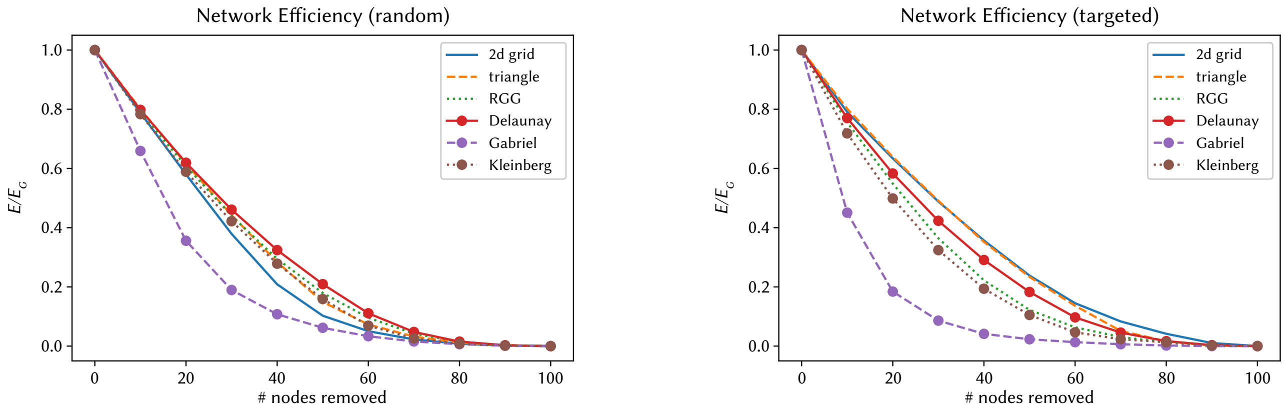
```
while !doomsday do
    for $f \leftarrow 1, e$ do
        $W \leftarrow Train(f)$
    end for
    for $j \leftarrow 1, k$ do
        $X_i(0) \leftarrow W_j$
        for $t \leftarrow 1, c$ do
            $X_i(t+1) \leftarrow X_i(t) + \varepsilon \sum_{a_j \in N_i} [X_j(t) - X_i(t)]$
        end for
    end for
end while
```

D. Network Topology

The underlying network topology does not affect the final consensus value but does the convergence speed. The effect of different network structures has been studied by Carrascosa et al. [1]. Random geometric graphs (RGG) are the most balanced solution between the efficiency in achieving the consensus value and the robustness under deliberate or accidental failures.

In an RGG, agents are located randomly in a square-unit area and linked with neighbors within a determined radius. It's the equivalent of a random graph, considering the spatial location of the agents.

Fig. 3 shows the robustness to agent failures of different network topologies: square and triangular grids, Kleinberg's networks, RGG, Delaunay triangulation, and Gabriel's graph (a simplification of Delaunay one). Comparing random failures and deliberate attacks, RGG and Delaunay have an adequate balance between algorithm



(a) Efficiency under random failures

(b) Efficiency under deliberated attack

Fig. 3. Efficiency of different network topologies.

performance and resilience. Nevertheless, RGG scales better when the size of the network grows.

Therefore, the underlying structure selected to configure the acquaintance's graph is an RGG using a given radius from the initial location of the agents. When no spatial information about the agents is available, we distribute them randomly in a fictitious space.

III. Execution Using SPADE Agents

This section presents the CoL algorithm implementation using *SPADE* agents [21]. Fig. 2 shows the *Co-Learning algorithm* in a network formed by four SPADE agents. Our CoL system is composed of two types of agents, *initialization* and *learning* agents: There is one *initialization* agent in the platform and *n learning* agents.

As its name suggests, the *initialization* agent is the agent in charge of setting up the whole system. It starts with reading a CSV file, which contains all the information related to the construction of the network of agents, indicating to each agent who is in contact. So, each agent will subscribe to the *presence* functionality of its neighbors, provided by the XMPP protocol features. The presence is a feature provided by the XMPP protocol to SPADE agents, enabling an agent to ascertain the status of other agents. This functionality is particularly valuable for determining whether an agent is connected and available for information exchange. The *initialization* agent is a utility agent that is not involved in the consensus process (in fact, the system has been tested adding new SPADE agents to the process during the execution of the system, without using this *initialization* agent).

Learning agents carry out the CoL process, exchanging the ANN model information with the neighbors with mutual subscriptions. The behavior of these *learning* agents is defined as a finite-state machine (FSM) in *SPADE* (See Fig. 4).

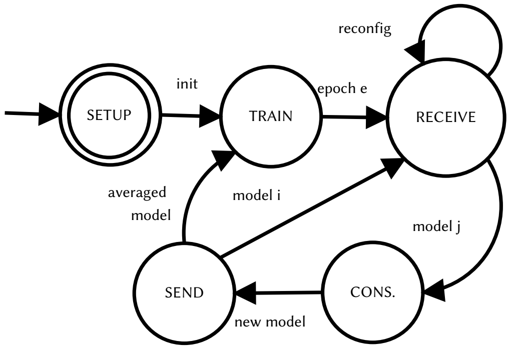


Fig. 4. FSM behavior for the SPADE learning agents doing the *CoL* Algorithm.

The first state is the *SETUP* state. In this state, the FSM that controls the agent is initialized. Then, it will pass to the *TRAINING* state, where it will train the model during e epochs. The next stage is the *RECEIVING* state, in charge of receiving two different kinds of messages: configuration messages and new training weights messages. The first one allows modifying the agent's acquaintances to change the network's structure if necessary. The second one is the messages the agents send to their neighbors during the consensus process to share their model. When the agent has received a message from all its active neighbors sharing their new training weights, it will pass to the *CONSENSUATING* state, where it will calculate a new aggregated model applying the consensus equation. Then, it will progress to the *SENDING* state, sending the latest model to its neighbors. While it is making c iterations of the consensus algorithm, it will go back to the *RECEIVING* state. When it has finished the c consensus iterations, it will go back to the *TRAINING* set, where it will use the new aggregated model to go on training during e epochs.

The *RECEIVE–CONS–SEND* loop finishes when there are no significant differences in the models. Then, the agent can begin a new training iteration if needed or conclude the complete process and use the ANN model.

IV. FIVE Framework

SPADE agents can run over the physical system or on a simulated one without relevant differences. Having available 3D virtual environments as close to reality eases the MAS' development and debugging, testing the agents' behavior in real-world conditions. This section describes the FIVE framework (Flexible Intelligent Virtual Environment developing framework) that will support the agents executing the CoL process.

A. FIVE Architecture

The FIVE framework is composed of three elements:

- The XMPP server.
- The FIVE simulator server, made with Unity[3].
- A set of SPADE agents that will populate the simulated environment.

Each component can transparently run on separate machines (including, of course, each SPADE agent being executed in a different host).

Fig. 5 shows an example of a FIVE simulation deployed into four local networks. The colored rectangles represent different local networks, and the arrows are network sockets. Each intelligent agent

[3] https://unity.com

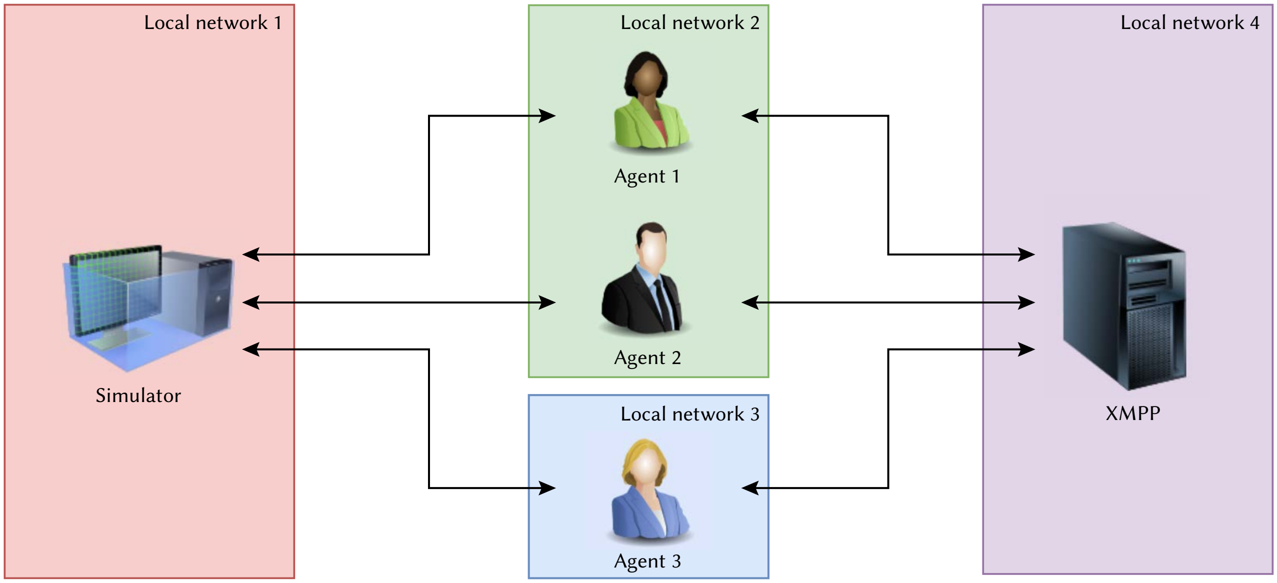


Fig. 5. Example of a FIVE simulation deployed in four different local networks.

represented in the figure runs on a different machine. *Agent 1* and *Agent 2* are on the same local network (*Network 2*). The three agents are connected to both the simulator and the XMPP server.

The FIVE simulator is a new tool made with Unity designed to define IVEs inhabited by SPADE agents. FIVE allows the creation of three-dimensional environments using a built-in text-based map editor. In addition, it will enable the rapid creation of custom agent avatars equipped with sensors, such as a camera.

FIVE agents (based on SPADE) control the virtual avatar in the IVE managed by the simulator. The framework grants network failure toleration: if an agent is disconnected from the FIVE simulator, it can be reconnected easily and resume its activity.

B. Defining a Simulation

FIVE simulations are composed of the environment created by the simulator and the intelligent agents that inhabit it. Defining a simulation is a process that just involves four text files (see Fig. 6). Three define the environment with elements such as terrain, trees, or light conditions, and the last file is used to create the agents.

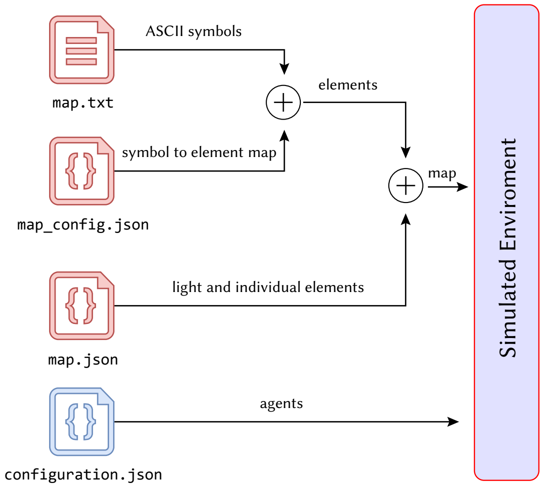


Fig. 6. FIVE simulator environment and agent generation from input files. The first three red files generate the intelligent virtual environment (composed of light objects, agent spawn points, and other elements), and the last blue file is used to generate the agents.

The file named `map.txt` is a text-based map where each ASCII character represents an object in the environment (Fig. 11 contains examples of all these files for the case study). This design decision was made to create the simulations easily and even modify the file through a text-based console.

The second file, `map_config.json`, assigns the map letters in the `map.txt` file to the objects in the simulator. The configurable properties are:

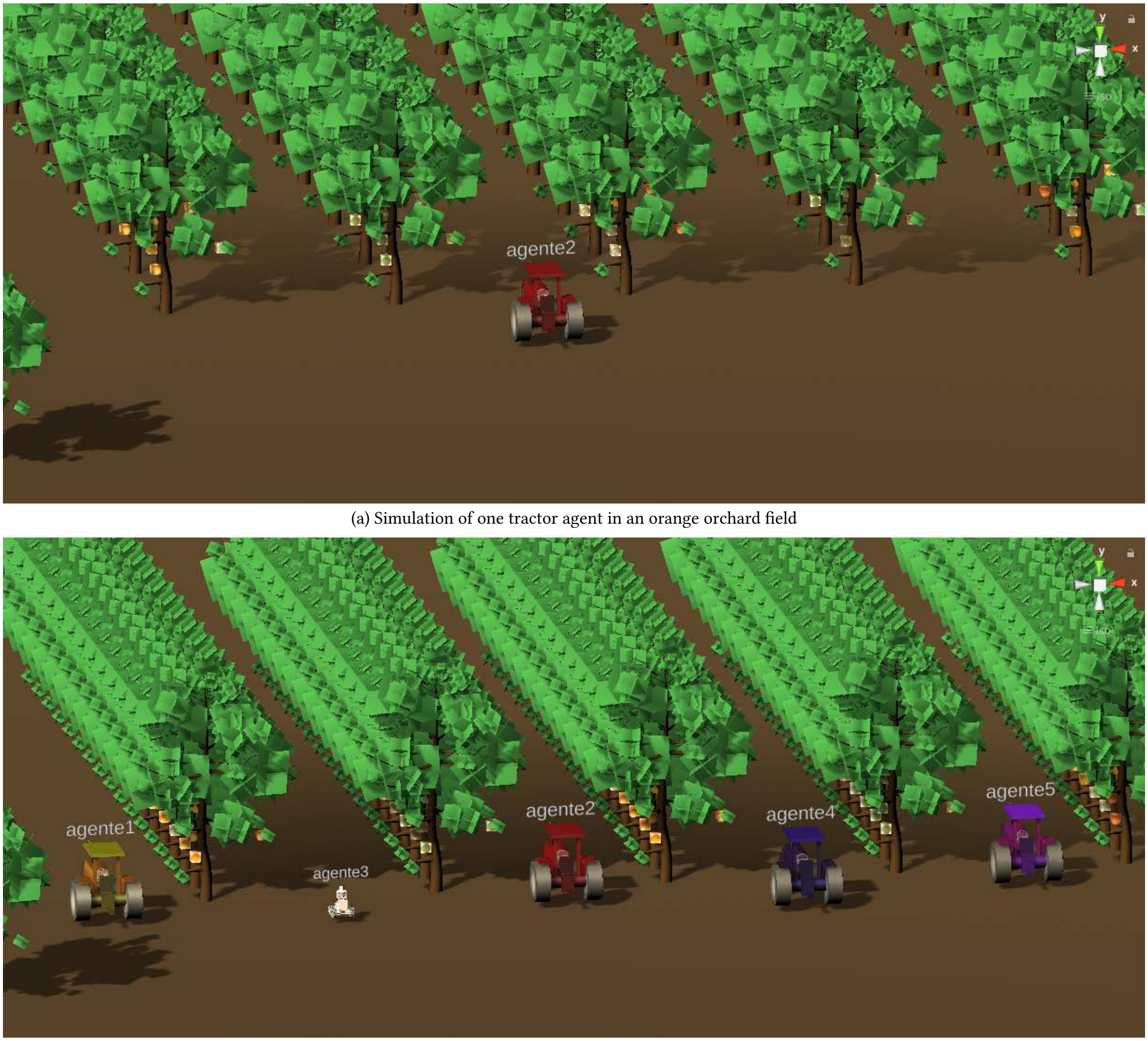
- `origin`: It represents the starting point where the elements will be placed into the simulation as a three-dimensional coordinate.
- `distance`: It represents the separation distance between elements in the different axis.
- `symbolToPrefabMap`: It is a list formed by three elements: the character that represents the element in map.txt file, the reference name of the object that replaces the letter, and an optional path that contain images to represent the object in the simulator

The third file used to generate the environment is `map.json`. The file sets the environmental conditions, such as light objects, and configures individual special elements. For example, if we need a river and a bridge that connects the two sides, the file includes configurable properties for these objects. The file contains two lists: one for objects with light properties and another for objects that do not need them. The main configurable properties are:

- `active`: Flag to create the object or ignore it.
- `objectName`: Internal name of the object.
- `position`: 3d coordinate where the element will be placed.
- `rotation`: Rotation (in degrees) in the three axes.
- `color`: Object with color data, in RGB and an alpha channel for opacity.
- `intensity`: Intensity of light ray.

Besides the files for generating the environment, the `configuration.json` file generates the inhabitant agents. This file includes the definition of all the information needed by the agents, including the FIVE simulator IP address, the avatar of the agents, or the spawn position. The configurable properties for the agents are:

- `name`: Name of the agent.
- `at`: XMPP server direction.



(a) Simulation of one tractor agent in an orange orchard field

(b) Simulation of four tractor agents and one robot agent in an orange orchard field

Fig. 7. Example of a simulation of an orange orchard field and agents. (a) there is only one agent. (b) there are five agents, and the space between trees is three times smaller than in figure (a).

- `imageBufferSize`: Maximum number of images per agent.
- `imageFolderName`: Name of the folder where images are saved. (related to the images perceived by the camera of the agent).
- `enableAgentCollision`: If this value is set to true, this agent will collide with other agents. Otherwise, it won't.
- `prefabName`: Avatar reference name for the agent.
- `position`: Spawn point position that can be referenced by name or by three-dimension coordinate.

FIVE simulator includes a library of existing elements by default, which can be incremented with new imported models. It contains several agent avatars that can be assigned to any inhabitant agent. Additional agent avatars can be added to the simulator through the Unity editor. The same can be said about the remaining objects that can be used to define the IVE.

It is important to underline that agent avatars include a configurable camera component so that the agent can take pictures of the IVE. Users can follow the track of any agent in the IVE in a first-person view. The resolution of these images can be easily configured. The camera component is not exclusive to included avatars; newly designed ones can also incorporate it.

To illustrate the effect of the change in configuration files, Fig. 7 depicts two different scenarios. In Fig. 7a, the distance between trees is nine, and there is only one agent in `configuration.json.` For Fig. 7b, we have reduced the distance to three, with five agents in the correspondent file.

C. Agents Programming

After defining the IVE, the next step is to program the agents' behavior. The FIVE framework includes an inhabitant agent's template, formed by a generic SPADE agent with an FSM behavior that implements the agent's execution cycle for communications with the

IVE. The code is addressed to control the avatars in the environment. The rest of the cognition related to the domain is included in the normal SPADE behaviors. The execution cycle (see Fig. 8) is composed of the following four states:

1. INITIAL STATE: The agent initializes variables to be referenced in other states. It also starts an instance of the *ImageManager* class on a background thread. The *ImageManager* class handles the incoming stream of captures taken by the agent's avatar in the FIVE simulator. It also adds the image data to a shared thread-safe queue for further processing.
2. PERCEPTION STATE: This state captures the image queue, dequeues them, and passes them to the agent behavior so that the images can be used for further process. The pictures are also automatically stored in the file system if desired.
3. COGNITION STATE: This state is where the process of cognition occurs. The agent decides to perform an action based on the information that it has at the moment.
4. ACTION STATE: In this state, the agent sends commands to the agent's avatar in the simulator. An example could be a camera rotation command or a move command.

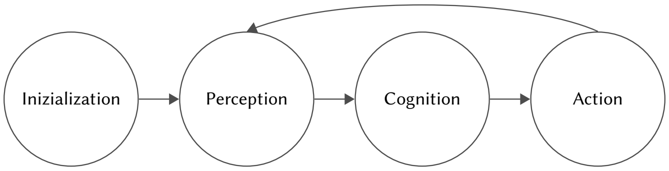


Fig. 8. FIVE agent FSM to control the avatar in the IVE.

Agent programming is done in a file named `entity_shell.py`. This file is an abstraction of the agent behavior explained above. It contains four methods that can be overloaded: `init`, `perception`, `cognition`, and `action`. Each method controls the execution of the agent in the corresponding FSM state.

The agent has access to a `Commander` class which defines an abstraction layer with the FIVE network protocol and contains methods to ease communication with the FIVE simulator. The current commands covered by `Commander` are:

- `create_agent`: It sends an instantiate request to the simulator, and the simulator returns the starting position coordinates to the agent. This command is always sent during the initial state to create the agent's avatar.
- `move_agent`: It sends a command to the simulator to move the agent's avatar to the desired position defined as (x, y, z). The simulator returns the agent the target position if the agent's avatar can reach this position. In the other case, the simulator returns the location where the agent got stuck.
- `fov_camera`: It sends a command to change the field of view value of the camera.
- `move_camera`: It sends a command to move the camera position.
- `rotate_camera`: It sends a command to rotate the camera.
- `take_image`: controls the image capture from the IVE.
- `change_color`: It sends a command to change the color of the agent.

Fig. 9 shows a possible execution interaction between one inhabitant agent and the FIVE simulator. The agent sends a first message to initiate the avatar in the simulation, adjusts the camera, and tries to move across the environment.

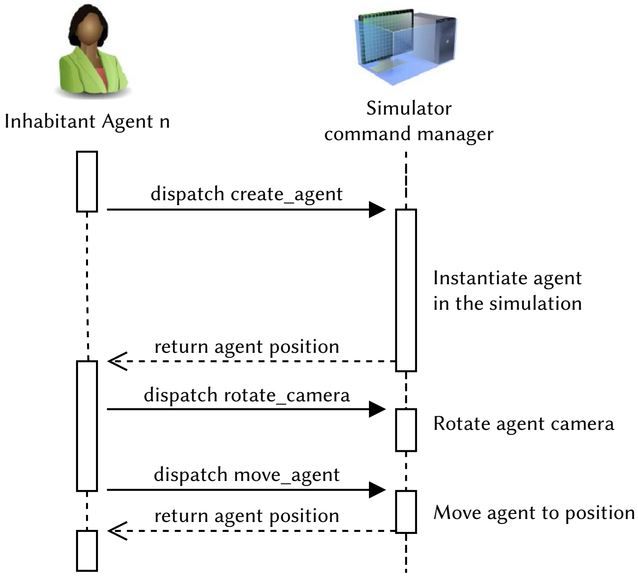


Fig. 9. FIVE agent communication through XMPP messages.

D. Executing a Simulation

With all the previous elements set, the FIVE system is ready for execution. It starts with the FIVE simulator generating the map elements, such as lighting, trees, or walls, and locating agents at their spawn points. First, the FIVE simulator parses the file named `map.json` and places the elements described by the JSON file in the IVE. Then, the FIVE simulator processes the ASCII characters in `map.txt`, situating the corresponding pieces in the IVE according to the letters' definitions. The simulator parses the file map_config.json to get the letters' associations and also sets the origin position for placing the elements and the amount of space between items.

Once the environment is ready, the FIVE simulator listens for incoming requests, handling the recently created sockets in new threads. The simulator provides the starting position coordinates as an answer to any agent sending a `create_agent` command, indicating its entity type information and spawning location data. The agent then initiates a new thread to handle the image socket's data reception to keep synchronized with the avatar and the agent.

Finally, each agent starts the FSM behavior that loops over the `perception`, `cognition` , and `action` states. The simulator process and executes all the commands, reflecting the agent actions in the IVE.

V. Case Study: A Simulation of an Orange Orchard Smart Area

The case study consists of the simulation of an IVE modeling an orange grove smart area. This simulation aims to test the CoL algorithm to train an ANN capable of detecting fruit diseases and what kind of disease it is. Several robots patrol the fruit orchard. Each robot trains its ANN with pictures of the fruits it views. Once the individual models are trained, they are shared and aggregated by consensus with CoL. The result is a model trained with the complete image dataset, even with pictures a particular robot has never seen.

This case uses several of the three-dimensional models available within the FIVE simulator: a tractor robot, a tree, and a white box representing the fruit. The white boxes will have orange textures that will be loaded from `map_config.json` `dataFolder` path, so the trees will show actual oranges hanging on their branches. Fig. 11 depicts some details of the configuration files with the map and object characteristics.

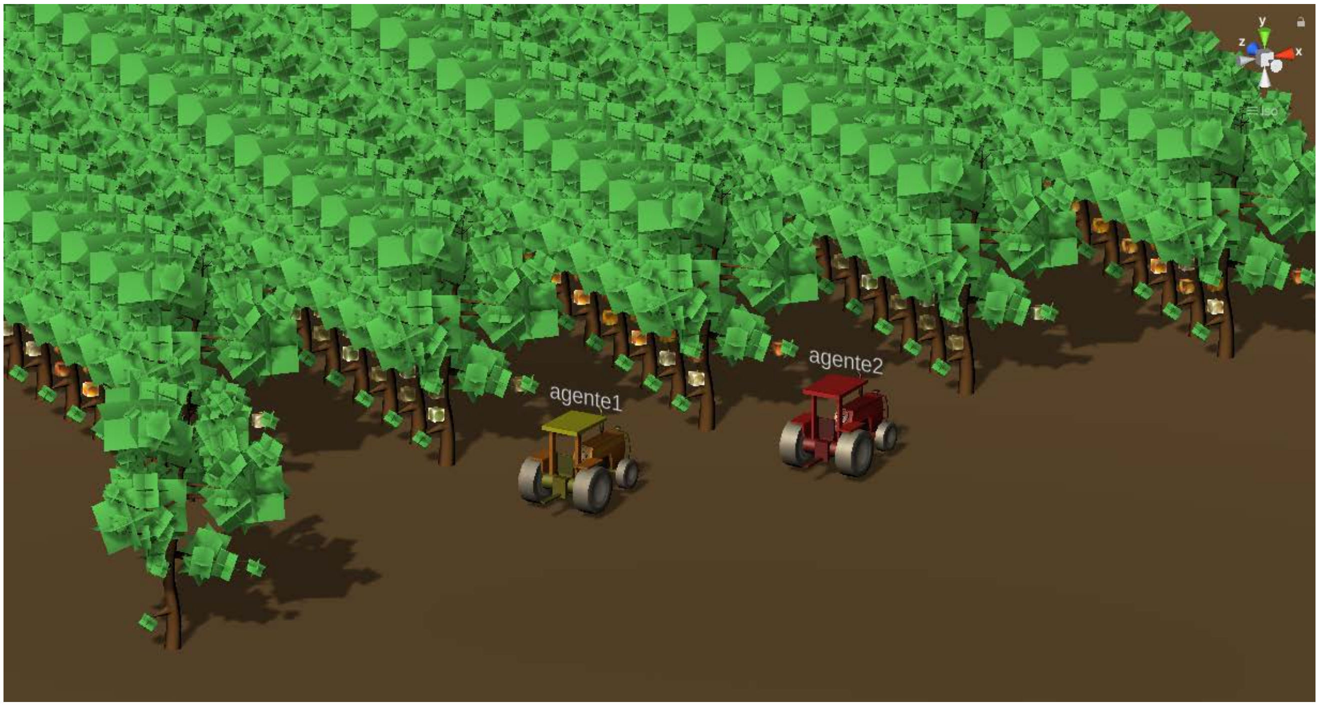


Fig. 10. Agents *agente1* (yellow tractor) and *agente2* (red tractor) patrolling and taking pictures of the oranges in the grove. Notice the "Tree Fruit Variant" trees with random textures of orange fruits applied at runtime, loading the images from the folder path specified in `map_config.json` file.

```
A A
   O G O G O
   O G O G O
   O G O G O
   O G O G O
   O G O G O
   O G O G O
   O G O G O
   O G O G O
   O G O G O
   O G O G O
```

(a) Portion of `map.txt` content.

```
{
   " symbol ": "G",
   " prefabName ": " Tree ␣ Fruit ␣ Variant ",
   " dataFolder ": "C:/ oranges / green "
},
{
   " symbol ": "A",
   " prefabName ": " Spawner "
}
```

(b) Portion of `map_config.json` content.

```
" active ": true ,
" objectName ": " Tree ␣ 1",
" objectPrefabName ": " Tree ",
" position ": {
  "x": -2.6,
  "y": 0.0,
  "z": 0.0
},
" rotation ": {
  "x": 0.0,
  "y": 5.0,
  "z": 0.0
```

(c) Portion of `map.json` content.

```
{
   " name ": " agente1 ",
   "at": " localhost ",
   " password ": " xmppserver ",
   " imageBufferSize ": 3,
   " imageFolderName ": " captures ",
   " enableAgentCollision ": true ,
   " prefabName ": " Tractor ",
   " position ": " Spawner ␣ 1"
}
```

(d) Portion of `configuration.json` content.

Fig. 11. Portions of the content of the four different files involved in defining the FIVE IVE of the case of study.

1. The letters in the map represents: *A* letter is replaced by an agent spawner point, *O* and *G* characters are orange trees. The difference is that *G* trees only have green oranges.
2. Besides identifying *G* and *O* with trees, the file contains in the `dataFolder` fields the paths with the corresponding orange pictures depending on its color. The orange ones might include diseases.
3. The environmental conditions include an isolated with a custom position and rotation in this case.
4. The last file includes the agents' declaration. Its `position` property refers to the name of the (invisible object) spawner where the agent will be created.

When the simulation starts, `map.json` file is parsed and its elements (the light and an isolated tree without oranges) are placed into the environment. Then the other components (agent spawner points and orange trees) from `map.txt` are added to the scene using the `map_config.json` information. Finally, the agents described in `configuration.json` file are spawned in the simulation and walk through the grove field, taking images of the oranges (see Fig. 10).

Although it can be specified in other ways, and even personalized in different ways for each agents, the test made have considered a random network.

A. Disease Identification

To validate the simulator, agents integrate a plant disease classification ANN. The architecture used for the experiments was a Mobilenet V2 [22] with the following hyperparameters definition: the agents make one epoch in their training step before changing to the receiving state in the FSM machine. The models of all agents are identical, having undergone training using data augmentation and fine-tuning, employing the following set of hyperparameters: Global Epochs:1; Local Epochs: 10; Local Batch Size: 10; Learning Rate: 0.001; SGD momentum: 0.5; Number of Each Kind of Kernel: 9; Number of Filters for Conv Nets: 32; Max Pooling: Yes; Network: CNN; Transfer learning (Tl): Yes or No.

This network was trained using the dataset presented in port [23], which has four classes Blackspot, Canker, Fresh, and Grenning. The dataset is divided into 80% for training, 10% for validation and 10% for testing. The training set contains 207 images for Blackspot, 202 images for Canker, 389 images for Fresh, and 370 images for Grenning. The testing set contains 139 images for Blackspot, 149 images for Canker, 165 images for Fresh, and 177 images for Grenning. Fig. 12 shows some pictures extracted from the dataset used to perform the training. The dataset images have been distributed along the different orange trees in the simulation, and each agent is able to access only a subset of trees as they are distributed along the different parts of the orange orchard. So, they are using different parts of the dataset.

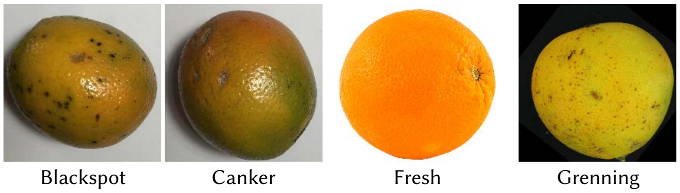


Fig. 12. Four sample images, one image of each citrus dataset class.

Fig. 13 shows the accumulated accuracy and loss obtained in the training process and the confusion matrix is presented and elaborated in the "Execution using SPADE agents" section of the original article, where a convergence analysis of the CoL algorithm have been conducted [1]. After training the network using the CoL process, the obtained model was integrated into the `Cognition` method available for the inhabitant agents and used for testing the model against the testing part of the dataset commented above.

As commented above, these agents include, by default, a camera for capturing images. The camera was adjusted using the `Commander` API, modifying its position and field-of-view via commands to focus on the fruit images as the tractor robot moved along the grove. These fruits were images of fruits loaded according to the dataset path indicated in the `map_config.json` file. Executing the agents would allow validating the values we obtained when training the network.

B. Modifying the Field Configuration

In this section, we are going to make modifications to the case of study in order to illustrate how simple it is to change a simulation in the FIVE framework. The modification consists of dividing the trees into five classes. As we have more agents capable of identifying diseases of the orange grove, we will obtain faster identification. Each agent will be spawned in a different column. Therefore, it will only be necessary for everyone to go through their column once to obtain captures of all the trees in the orchard. We are also going to modify the environmental conditions so that the captures are taken at night, checking the identification precision under poor light conditions.

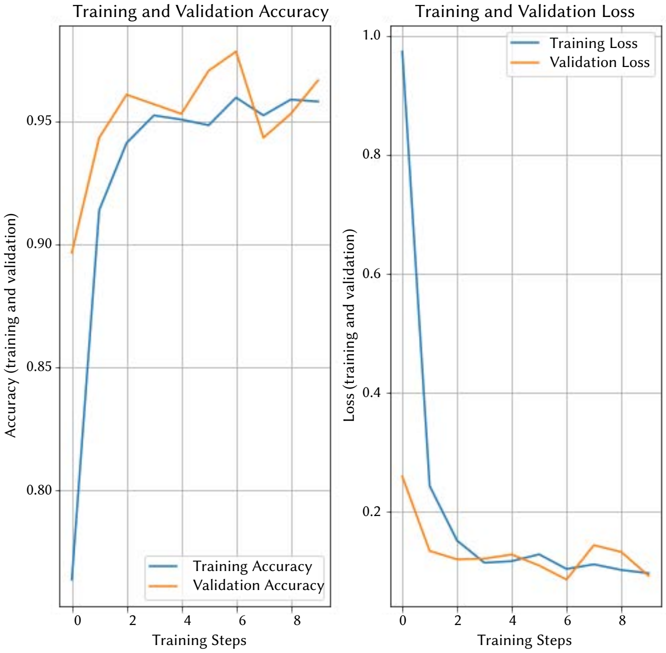


Fig. 13. Accumulated Train Accuracy and Accumulated Train Loss.

First, we have to create five folders: one folder for each class. Each folder will contain four images rendered as a texture and randomly applied to the oranges in the tree to which the folder class belongs. To achieve this, we must modify the `map.txt` and `map_config.json` files. In `map.txt`, we add three more *A* letters to create the new agents, and we have to define a character for each tree class. For example, we can use *B* for black spot, *C* for Canker, *G* for Greening, *M* for Melanose, and *H* for Healthy. Then, in map_config.json file, we match the characters with the elements they represent, as letter *G* is defined in Fig. 11b. Finally, we must modify the `dataFolder` property with the folder where the images are to load the textures.

Next, we must update the `map.json` file to modify the environmental conditions. Our desired light condition is moonlight, so we can change the intensity and color of the light used in map.json without writing a single line of code.

Finally, we have to modify configuration.json file and add three more agents as *agente1* is defined in Fig. 11d. We can change the `name` property of the new agents to *agente3*, *agente4*, and *agente5*. We can also set the initial position of them in *Spawner 3*, *Spawner 4*, and *Spawner 5* generation points. Lastly, it has to be indicated the neighbors of the new agents generated.

Once we have defined all our modifications in the four files involved, we can start the simulation process, and the result is shown in Fig. 14. As a result, we have a completely new environment to test whenever the ANNs trained in good light conditions are valid or if they need some retraining process to adjust the parameters to the new scenario.

C. FIVE Loading Time Test

The last experiments measure the load time FIVE simulator needs to load a complete scene populated by elements that use the FIVE system to apply textures from images at runtime.

The dimension of the images for the textures is 224 × 224 pixels, randomly chosen from five classes located in folders with four images each, adding a total of twenty different pictures. The execution platform is a laptop without an external graphics card and with the following components: an integrated graphics model *Intel Iris Plus Graphics*, an *Intel Core i7 processor 1035G7*, 16 GB of RAM, a motherboard *ASUSTeK X421JAY* and a storage device *NVMe Intel SSDPEKNW01*.

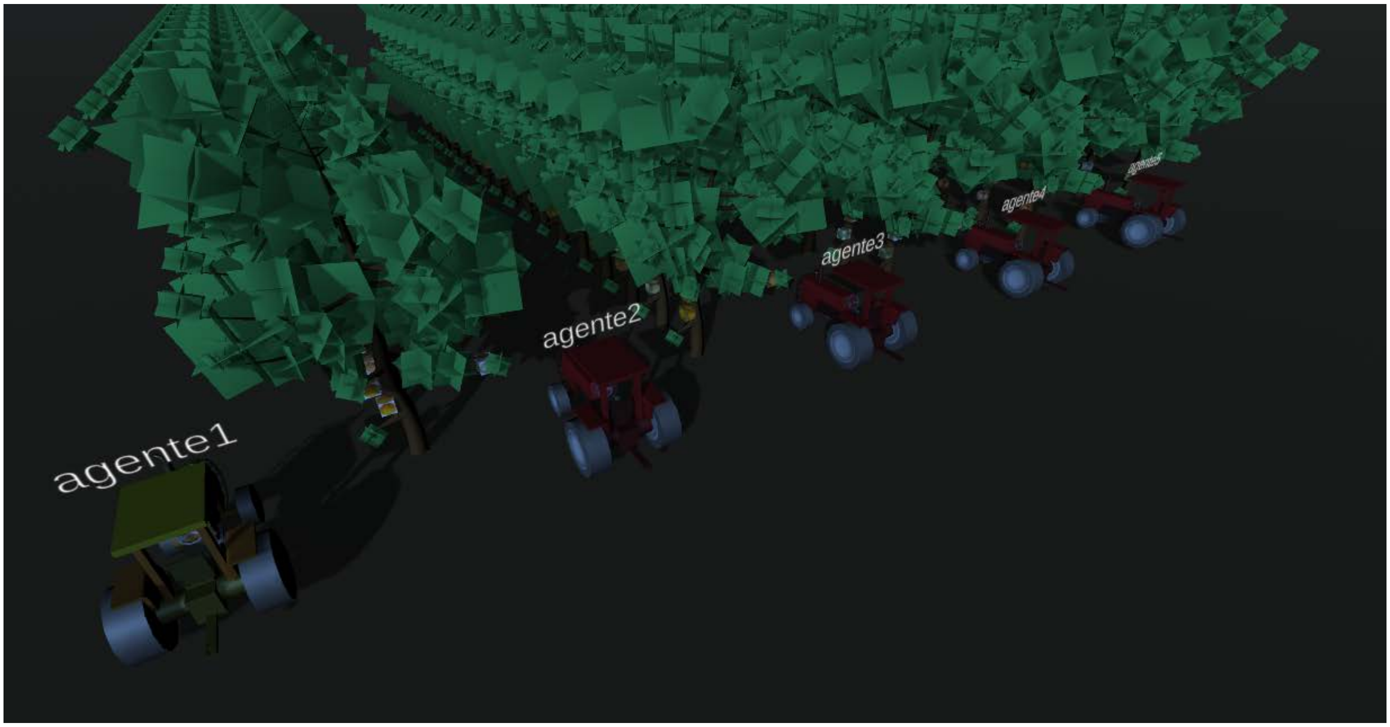


Fig. 14. Five agents in the modified case of study with nocturnal environmental conditions.

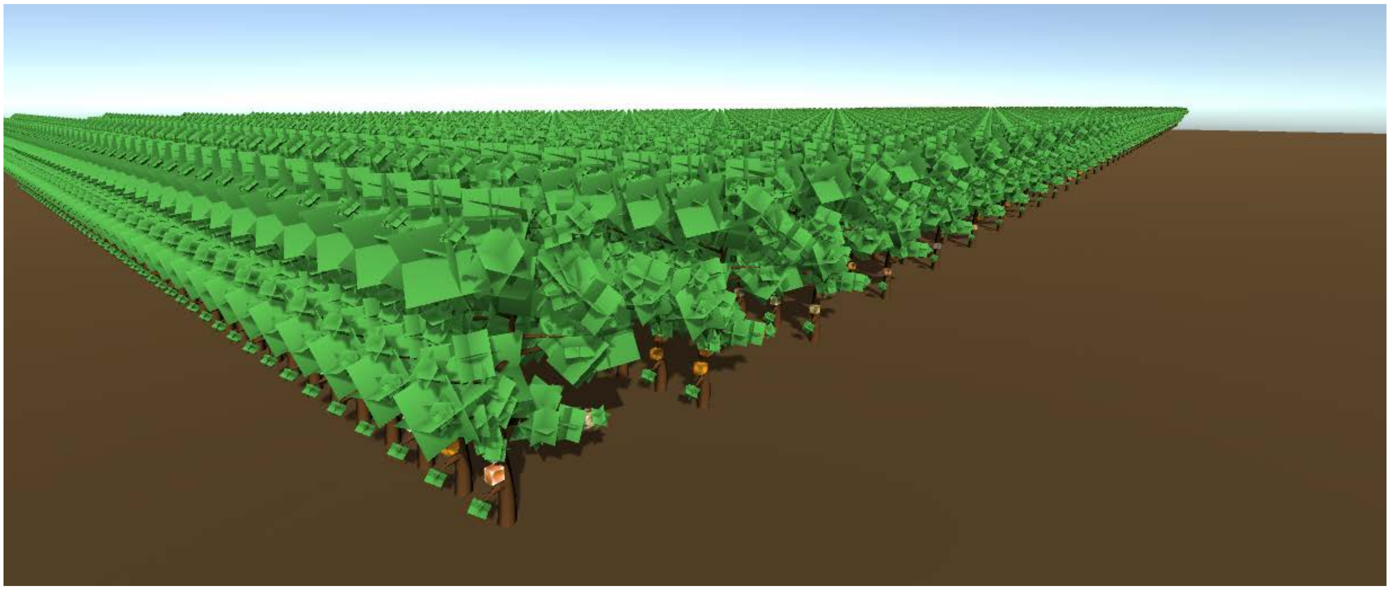


Fig. 15. Modified case of study with daylight environment conditions and two thousand five hundred orange trees.

Fig. 16 shows a graph that illustrates the time (in seconds) it has taken to load the entire scene, populating it with a light object, the terrain, and a variable number of fruit trees. The loading time of the whole scene has been measured, not just the texture loading process.

In conclusion, we can see that FIVE is ready to load complex environments quickly and effectively. The reason is that FIVE uses optimization techniques that allow us to simulate scenes with a large number of different elements.

VI. Conclusions

We have presented a Consensus-based Learning algorithm (CoL) that takes advantage of distributed learning based on the idea behind federated learning of sharing a model between a set of agents. This advantage is based on complementing individual models the agents train with their aggregation. By doing this, all agents may benefit from the training completed by the rest of the agents. The agents share the parameters of the models but not the training data. Therefore, privacy is maintained during the training. As we use a consensus-like algorithm for the model's aggregation, we have some other advantages as the adaptation to variations in the agent set, allowing agents to abandon and enter during the execution. The paper shows the implementation of CoL algorithm in *SPADE* agents.

RGG topology improves the performance of the convergence of consensus since the average path lengths are shorter than the rest of the networks and is a pretty robust topology under random or deliberate failures. Therefore, we propose its use as the underlying structure for the MAS.

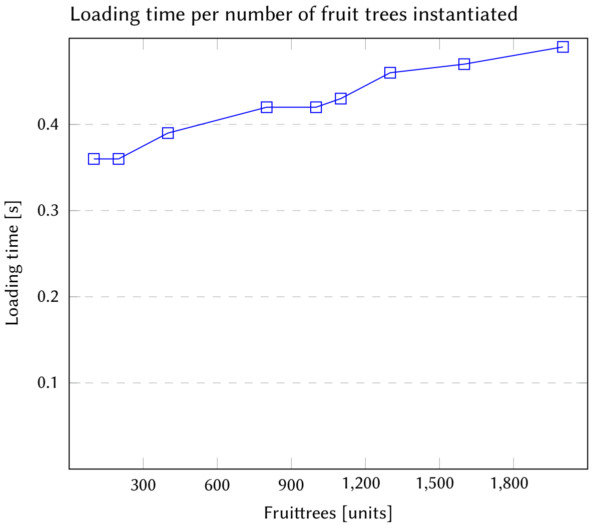


Fig. 16. Graph showing the time it took to load the scene composed of fruit trees, with four fruits each, and loading their textures at runtime from images.

Moreover, we have presented a new framework called FIVE that allows the easy creation and modification of IVEs inhabited by SPADE agents. This framework has been used to test CoL in SPADE agents through an orange orchad simulation.

As part of our future work, we are dedicated to enhancing the communication between agents in the CoL process. This includes optimizing message transmission, both in terms of quantity and size. Additionally, we are actively exploring the generation of simulated maps, where satellite images are leveraged to create them automatically. Lastly, we are delving into the possibility of introducing semantic coalitions among agents. This entails agents that share similar meanings in the data they handle, engaging in more frequent information exchange with each other compared to other agents in the network.

Acknowledgment

This work has been developed thanks to the funding of projects:

- Grant PID2021-123673OB-C31 funded by MCIN/AEI/ 10.13039/501100011033 and by "ERDF A way of making Europe"
- PROMETEO CIPROM/2021/077
- TED2021-131295B-C32
- Ayudas del Vicerrectorado de Investigacion de la UPV (PAID-PD-22)

References

[1] C. Carrascosa, J. Rincón, M. Rebollo, "Co-learning: Consensus-based learning for multi-agent systems," in *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation. The PAAMS Collection*, 2022, pp. 63–75.

[2] H. Sánchez San Blas, A. Carmona Balea, A. Sales, L. Augusto Silva, G. Villarrubia González, "A platform for swimming pool detection and legal verification using a multi-agent system and remote image sensing," *International Journal of Interactive Multimedia and Artificial Intelligence*, 2023, pp. 1-13, doi: 10.9781/ijimai.2023.01.002.

[3] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson, B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, 2017, pp. 1273–1282, PMLR.

[4] P. Kairouz, H. McMahan, B. Avent, A. Bellet, M. Bennis, A. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., "Advances and open problems in federated learning," *Foundations and Trends in ML*, vol. 14, no. 1–2, pp. 1–210, 2021.

[5] R. Olfati-Saber, R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE TAC*, vol. 49, no. 9, pp. 1520–1533, 2004.

[6] J. Palanca, A. Terrasa, V. Julian, C. Carrascosa, "SPADE 3: Supporting the new generation of multi-agent systems," *IEEE Access*, vol. 8, pp. 182537–182549, 2020, doi: 10.1109/ACCESS.2020.3027357.

[7] J. Palanca, J. Rincon, V. Julian, C. Carrascosa, A. Terrasa, "Developing iot artifacts in a mas platform," *Electronics*, vol. 11, no. 4, p. 655, 2022.

[8] J. Rincon, V. Julian, C. Carrascosa, "Flamas: Federated learning based on a spade mas," *Applied Sciences*, vol. 12, no. 7, pp. 1–14, 2022, doi: 10.3390/app12073701.

[9] M. Bratman, *Intention, Plans, and Practical Reason*. Cambridge: Cambridge, MA: Harvard University Press, 1987.

[10] M. Luck, R. Aylett, "Applying artificial intelligence to virtual reality: Intelligent virtual environments," *Applied artificial intelligence*, vol. 14, no. 1, pp. 3–32, 2000.

[11] A. Ikidid, E. F. Abdelaziz, M. Sadgal, "Multi-agent and fuzzy inference-based framework for traffic light optimization," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, no. 2, pp. 88-97, 2023, doi: 10.9781/ijimai.2021.12.002.

[12] U. Wilensky, "Netlogo (and netlogo user manual)," *Center for connected learning and computer-based modeling, Northwestern University*. http://ccl. northwestern. edu/netlogo, 1999.

[13] J. Rincon, E. Garcia, V. Julian, C. Carrascosa, "The jacalive framework for mas in ive: A case study in evolving modular robotics," *Neurocomputing*, vol. 275, pp. 608–617, 2018.

[14] A. Barella, A. Ricci, O. Boissier, C. Carrascosa, "Mam5: multi-agent model for intelligent virtual environments," in *10th european workshop on multi-agent systems* (*EUMAS 2012*), 2012, pp. 16–30.

[15] R. H. Bordini, J. F. Hübner, M. Wooldridge, *Programming multi-agent systems in AgentSpeak using Jason*. John Wiley & Sons, 2007.

[16] A. Ricci, M. Viroli, A. Omicini, "Cartago: A framework for prototyping artifact-based environments in mas," in *International Workshop on Environments for Multi-Agent Systems*, 2006, pp. 67–86, Springer.

[17] S. Luke, G. C. Balan, L. Panait, C. Cioffi-Revilla, S. Paus, "Mason: A java multi-agent simulation library," in *Proceedings of Agent 2003 Conference on Challenges in Social Simulation*, vol. 9, 2003.

[18] F. Enguix Andrés, *Desarrollo de un generador de simulaciones en Unity 3D para sistemas multi-agente basados en SPADE*. PhD dissertation, Universitat Politècnica de València, 2022.

[19] A. Palomares, M. Rebollo, C. Carrascosa, "Supportive consensus," *PLOS ONE*, vol. 15, no. 12, pp. 1–30, 2020.

[20] F. Pedroche, M. Rebollo, C. Carrascosa, A. Palomares, "Convergence of weighted-average consensus for undirected graphs," *International Journal of Complex Systems in Science*, vol. 4, no. 1, pp. 13–16, 2014.

[21] J. Palanca, A. Terrasa, V. Julian, C. Carrascosa, "Spade 3: Supporting the new generation of multi-agent systems," *IEEE Access*, vol. 8, pp. 182537–182549, 2020, doi: 10.1109/ACCESS.2020.3027357.

[22] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, L. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," *CoRR*, vol. abs/1801.04381, 2018.

[23] M. C. Silva, J. C. F. da Silva, R. A. R. Oliveira, "Idissc: Edge-computing-based intelligent diagnosis support system for citrus inspection.," in *ICEIS* (1), 2021, pp. 685–692.

Carlos Carrascosa

Dr. Carlos Carrascosa was born in Valencia (Spain) and received the M.S. degree in Computer Science from the Universidad Politécnica de Valencia (UPV), in 1995. He obtained his Ph.D. in the Departamento de Sistemas Informáticos y Computación at UPV and is currently a Lecturer involved in teaching several AI-related subjects at the UPV. He is member of the VRAIn (Valencian Research Institute for Artificial Intelligence) where he develops his research that include MAS, Federated Learning, consensus in MAS, IVEs, social emotions, serious games, information retrieval, and real-time systems.

Francisco Enguix

Francisco Enguix Andrés was born in Valencia (Spain). He is pursuing a Master's degree in Artificial Intelligence, Pattern Recognition and Digital Imaging at Polytechnic University of Valencia (UPV) while he works at the Valencian Institute for Research in Artificial Intelligence (VRAIN). He holds a degree in Computer Science from the Polytechnic University of Valencia (UPV) since 2022.

Miguel Rebollo

Dr. Miguel Rebollo received his PhD. in Artificial Intelligence from the Universitat Politècnica de València (Spain) in (2004), and Dr. in Complex Systems from the Universidad Politècnica de Madrid (Spain) in 2019. He is a member of the Valencian Research group for Artificial Intelligence (vRAIN). He works as Associate Professor at the Universitat Politècnica de València. His research interests involve complex intelligent adaptive systems, multi-agent systems, chaos and non-linear systems, and social network analysis.

Jaime Andrés Rincón Arango

Jaime Andrés Rincón Arango is a postdoctoral researcher at the Valencian Institute for Research in Artificial Intelligence (VRAIN) of the Polytechnic University of Valencia. He holds a degree in Biomedical Engineering from the Universidad Manuela Beltrán (Colombia), a Master's degree in Artificial Intelligence from the Universidad Politécnica de Valencia and a PhD in Computer Science from the Universidad Politécnica de Valencia. His main research activities focus on IoT, IoMT, Cognitive Assistants, assistive robotics for the elderly and Edge AI. He is author or co-author of more than 50 articles in specialized journals and national and international conferences.

Development of an Intelligent Classifier Model for Denial of Service Attack Detection

Álvaro Michelena[1]*, Jose Aveleira-Mata[2], Esteban Jove[1], Héctor Alaiz-Moretón[3], Héctor Quintián[1], José Luis Calvo-Rolle[1]

[1] University of A Coruña, CTC, CITIC, Department of Industrial Engineering, Ferrol, A Coruña, (Spain)
[2] University of León, RIASC: Research Institute of Applied Sciences in Cybersecurity, León, (Spain)
[3] University of León, Department of Electrical and Systems Engineering, León, (Spain)

Received 23 February 2023 | Accepted 31 July 2023 | Published 21 August 2023

Abstract

The prevalence of Internet of Things (IoT) systems deployment is increasing across various domains, from residential to industrial settings. These systems are typically characterized by their modest computational requirements and use of lightweight communication protocols, such as MQTT. However, the rising adoption of IoT technology has also led to the emergence of novel attacks, increasing the susceptibility of these systems to compromise. Among the different attacks that can affect the main IoT protocols are Denial of Service attacks (DoS). In this scenario, this paper evaluates the performance of six supervised classification techniques (Decision Trees, Multi-layer Perceptron, Random Forest, Support Vector Machine, Fisher Linear Discriminant and Bernoulli and Gaussian Naive Bayes) combined with the Principal Component Analysis (PCA) feature extraction method for detecting DoS attacks in MQTT networks. For this purpose, a real dataset containing all the traffic generated in the network and many attacks executed has been used. The results obtained with several models have achieved performances above 99% AUC.

Keywords

Cybersecurity, DoS Attacks, Feature Extraction, MQTT, Soft Computing, Supervised Classifiers.

DOI: 10.9781/ijimai.2023.08.003

I. Introduction

IoT (Internet of Things) allows daily objects to acquire new functionalities, such as gathering information from the environment or performing actions in the environment through actuators. Thanks to internet connectivity, these devices can collect, analyze, and share data between objects, software applications, and cloud platforms. Concepts such as smart cities [1] and Industry 4.0 [2] have emerged thanks to healthcare devices, industrial sensors, and actuators connected to the Internet.

Recent market studies have predicted that the number of connected devices will be more than 70% of total internet connections, with the number growing by 180% in the next four years [3].

IoT systems present new cybersecurity challenges due to the heterogeneous growth in the number of devices and linked services. Operating in resource-constrained environments, such as networks with low transfer rates due to interference, low power consumption, and small embedded processors, requires the use of simple protocols and devices, which may limit security aspects [4], [5].

The different protocols can be represented like a layered structure, where each of them provides a different functionality [6], being the most widely used architecture the three-layer topology. Considering the studies on the protocols used in IoT environments [7], [8], they can be classified according to Table I.

TABLE I. IoT Protocol Classification in Three Layers

Protocols	Layers
XMPP, MQTT, CoAP, Web-Socket, HTTP REST	Application
UDP, TCP, 6LoWPAN	Network
LoRa, IEEE 802.15 (BLE, Bluetooth, ZigBee), IEEE 802.11(Wi-Fi)	Physical

Malicious actors can exploit a diverse range of attack vectors, based on the special behaviours of this kind of environment. As a result, there is a growing interest in cybersecurity topics research around IoT. In the review addressed by Lu & Xu [9], a clear upward trend in research on "IoT cybersecurity" is shown.

Attackers usually exploit vulnerabilities of specific IoT protocols embedded in TCP/IP networks [10]. One of the most common attacks is a denial of service (DoS) which consists of the attacker saturating the network with a large volume of traffic until the system cannot provide [11] service. One of the most famous attacks that have been performed on the Internet was the "Mirai" botnet, developed on September 2016. It performed a DDoS attack, based on a distributed denial of service over "DynDNS" servers, being one of the largest DNS service providers systems. "Mirai" attack generated 1.2 terabits of malicious traffic, forcing to set of "DynDNS" servers, the out of service during several hours, which caused the fall of widespread of internet services such as Twitter, Netflix, Reddit, and GitHub [12]. A more recent botnet attack was "dark_nexus" which dated in 2020 compromised around 1370 devices. Bitdefender analysis report [13] shows how "dark_nexus" works, with a behaviour very similar to Mirai.

* Corresponding author.

E-mail address: alvaro.michelena@udc.es

A. IoT Cybersecurity Solutions

This subsection addresses state of the art to show the most popular solutions for protecting IoT environments.

The research work conducted by Idriss et al. [14], delves into various options for implementing cybersecurity in IoT systems, being the most notable of them, the implementation of a hardware module that allows adding randomness to the encryption in a more lightweight way than other methods, calling PUF (Physical Unclonable Functions) based lightweight authentication. Amanlou et al. [15] proposes a lightweight authentication system for IoT systems using the MQTT protocol, a temporary key exchange algorithm FCDHE, and the shared key authentication (PSK) algorithm. This combination provides mutual authentication between IoT network devices thanks to an authentication scheme known as ECDHE-PSK. This implementation would also improve IoT cybersecurity using this protocol. However, the systems deployed previously must be modified.

In the last few years, new IoT cybersecurity approaches have been published. Zhu & Deng [16], include IoT security situation classification based on support vector machines and security situation awareness based on Markov game model. Choudhary & Pahuja [17], present a new technique called Steering Convention for Vitality Effective Systems (SC-VFS) that improves vitality proficiency and ensures the safety of sensitive information in remote sensor networks, with a focus on detecting doppelganger attacks in IoT-based intelligent health applications. Berjón et al. [18], introduce the SCIFI-II system, which simplifies the development of applications in IoT contexts by allowing the distribution of events between event brokers and designing components that are decoupled from the event brokers.

To address cybersecurity without modifying existing systems, implementing Intrusion Detection Systems (IDS) is the main solution since they can analyse the traffic generated by the environment, without intervening in its configuration. There are several types of intrusion detection systems, depending on the paradigm applied in their detection module, being theses rule-based IDS or anomaly-based IDS [19]. The anomaly-based IDS paradigm observes network traffic features to detect attacks by identifying altered behaviour within the network.

Anomaly detection systems (IDS) are an effective solution for implementing attack detection in IoT systems. They are highly versatile in detecting new types of attacks and can adapt to new protocols. Anomaly IDS systems utilize classification models created with soft computing techniques, such as machine and deep learning, supported by neural networks [20]. The implementation of these procedures requires training models using high quality datasets [21].

B. Objectives

Based on the state of the art addressed previously, this paper aims to develop a functional IDS with an intelligent model for detecting DoS attacks on the MQTT protocol. The model will be constructed thanks to applying soft computing techniques based on machine learning techniques.

To achieve a functional IDS, several tasks are addressed. These tasks are described throughout the paper as follows:

- Study the data sets available to develop the intelligent model that applied the soft computing techniques chosen (Section II).
- Collect a new MQTT dataset (How this dataset has been constructed will be addressed in Section III) because no MQTT datasets exist with normal and DOS traffic for applying machine learning methods.
- Chose and test a set of machine learning methods for application to the previously defined dataset, to achieve the best model for deployment in the IDS (Sections IV and V).

II. Related Works

In order to implement a set of machine learning techniques for getting a functional model that will be inserted in an IDS, it is necessary to work with a specific dataset. This dataset consists of labeled traffic frames, each one tagged as standard/normal network traffic or traffic with hostile purposes. Thanks to the models obtained after a training process, the IoT MQTT behaviour is modeled as well as the recognition of the most important features for understanding this behaviour.

Using general purposed datasets collected from TCP/IP networks can be a solution for modeling attacks such as botnets, without focusing on the special characteristics of IoT systems [22]. To obtain anomaly-based IDS capable of detecting DoS attacks, well-known datasets are used. Some of them were created like over general purposed networks (non IoT networks) such as the NSL-KDD dataset [23], which an enhanced version of the KDD99 dataset was developed in 1993. Some research works address the use of different artificial intelligence techniques for modeling traffic and detecting distributed denial of service DDoS attacks, caused by IoT system botnets [24], [25]. Liu et al. [26], also use the NSL-KDD dataset for getting the model that will be included in the IDS, in this case, Kontiki is the software utilized for simulating IoT environments where CoAP (Constrained Application Protocol) works.

MQTT is typically utilized to connect small devices with restricted bandwidth in IoT [27] and Industry 4.0 environments [28]. MQTT is a publish/subscribe protocol designed for lightweight machine-to-machine (M2M) communications, being ideal for connecting small devices to networks with low bandwidth. MQTT architecture follows a star topology with a central server node called a Broker. The communication is based on topics. Clients can create and publish topics, while others that want to receive information from that topic, can subscribe to it. The broker side handles all the load of the overall system. This operation can be seen in detail in Fig. 1.

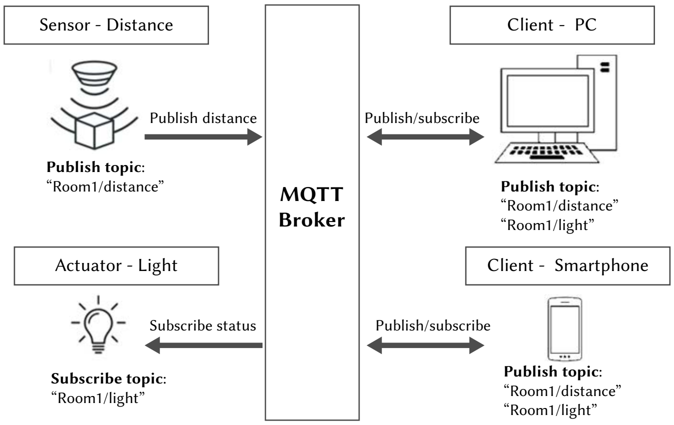

Sensor - Distance
Publish distance
Publish topic: "Room1/distance"
MQTT Broker
Client - PC
Publish/subscribe
Publish topic: "Room1/distance" "Room1/light"
Actuator - Light
Subscribe status
Subscribe topic: "Room1/light"
Client - Smartphone
Publish/subscribe
Publish topic: "Room1/distance" "Room1/light"

Fig. 1. MQTT environment.

MQTT does not specify any networking or routing techniques; it uses TCP as a transport protocol and TLS/SSL for security. IoT application protocols, such as MQTT, can be supported by transport layer security (TLS), but there are no mechanisms in place to protect IoT devices from denial of service (DoS), being this susceptible to this kind of attack. Several datasets have recently been created that focus on attacks on MQTT systems. For example, "MQTT-iot-ids2020" [29] was generated using a simulated MQTT architecture that consists of twelve sensors sending random messages, a server that manages the connections called "broker", a simulated camera, and an attacker. On top of this environment, the attacker performs network scanning and brute force attacks to decrypt access credentials.

The "TON_IoT" dataset [30] focuses on modern IoT systems using the MQTT protocol. It was generated in a simulated environment with the NSX-VMware platform [31] where network scanning process and DoS attacks are performed on the MQTT environment.

III. Case Study

A Denial of Service (DoS) attack involves flooding a network with a high traffic volume to the point where the system cannot provide its intended services. This attack particularly affects IoT systems, since they have limited computational capacity and most protocols they use are for processing information in real time [32].

The previously described datasets use simulated environments and traffic that is collected by only considering the frames of the MQTT protocol. This paper presents the development of a dataset that aggregates all traffic from a real-world environment, utilizing an IoT system with the MQTT protocol. Notably, any denial of service attacks against the broker within this system is labeled. Therefore, an MQTT environment is developed to simulate real traffic thanks to a broker programmed in "node.js" with the "Aedes" library [33]. It uses an actuator with a relay, a distance sensor, and two clients: a smartphone and a computer. All the traffic generated in this environment, including the interactive Internet traffic, is captured by a router with the "OpenWRT OS" installed.

Several DoS attacks are performed on the environment, taking into account the vulnerabilities of the protocol. An attacker scans the network with a search engine like "Shodan" through the well-known port 1883 [23]. Thanks to this, it is possible to find out which servers use this protocol as a broker, being this the vulnerable part of the MQTT system, due to this centralizes all control of the system.

The attacks are performed with a tool developed for performance testing called "Malaria MQTT" [34]. This tool sends many messages to the broker, simulating 1000 clients, sending 1000 messages per second with a size of 100 characters. Thanks to this, it is not possible for the broker to respond to all of these messages, generating a service failure in the IoT environment.

To generate the dataset, all the traffic in the test environment developed is captured, standard internet browsing traffic and traffic generated by the IoT environment. The router registers all the traffic for generating a PCAP file. The set of PCAP files contains a lot of information and many fields. In this way it is simplified by a dissecting procedure. With this purpose, a tool developed for the authors was designed [35]. The dissecting tool works as follows:

- The frames in a pcap file must be organized to analyze a DoS attack effectively. During an attack, a large number of frames may be generated in a short period of time, and the capture tool (such as tcpump with OpenWRT) may overlap several frames with the same timestamp. To obtain useful information about the attack, it is necessary to separate these overlapping frames based on their timestamps.
- The frames are dissected by taking some fields common to all the frames. These common fields are chosen, taking as an example the AWID dataset, which is from 802.11 protocol. All fields that make up the MQTT protocol are included, resulting in 65 fields for each frame.
- To properly label each frame as either part of an attack or normal traffic, it is necessary to consider the timestamp of when each attack begins and ends. Each frame should be tagged based on this information, allowing for a clear distinction between attack frames and normal traffic.

The resulting dataset contains all traffic generated by the described environment, capturing both the normal operation traffic and the traffic under a DOS attack on the MQTT protocol. The dataset comprises a CSV file in which 65 fields delineate each captured frame. It compiles a total of 94,625 frames, 45,513 of which are labeled as "under attack" in the "type" field, while 49,112 are labeled as "normal". This dataset is currently accessible online [36].

IV. Soft Computing Techniques Used

Two stages are implemented to detect DoS attacks in MQTT networks with a functional IDS based on an intelligent model. The first one reduces the dataset dimensionality, while in the second one, a set of classification methods are implemented, choosing the best one.

Therefore, this section is divided into two subsections. Section IV.A will describe the feature extraction method employed, while Section IV.B will define the six different classification techniques implemented.

A. Feature Extraction Method

As discussed in Section III, the working dataset contains a total of 65 variables. This large number of features can lead to a high computational cost in the model training process and a certain mathematical complexity in the classifiers. Therefore, in these scenarios, it is very common to use dimensionality reduction techniques, based on feature extraction, to minimize the number of variables in the dataset in order to reduce the computational cost and obtain simpler classifiers with good performance. Nowadays, there is a wide variety of techniques and algorithms for dimensionality reduction, being Principal Component Analysis (PCA) one of the most common.

1. Principal Component Analysis

Principal Component Analysis (PCA) is an unsupervised multivariate statistical approach developed by Pearson [37] and is generally used for dimensional reduction. The variation of a multivariate dataset is described by this technique as a set of uncorrelated variables corresponding to linear combinations of the original parameters. In general, the principal purpose of this strategy is to generate a new set of orthogonal axes that maximize data variance, avoiding the loss of information. This is accomplished by computing the eigenvalues of the correlation matrix. The initial set can then be linearly translated into lower dimension space using the eigenvectors [38]. Fig. 2 shows an example in $\mathbb{R}^2$ of obtaining the principal components.

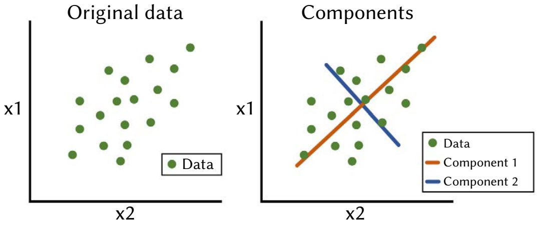


Fig. 2. PCA example.

B. Classification Methods

This subsection describes briefly the six supervised classification techniques implemented in this research.

1. Decision Trees

One of the simplest and most widely used supervised machine learning techniques are decision trees (DT). This method is based on generating a model with a hierarchical tree structure with a root node, branches, decision nodes, and response nodes, also known as leaves [39].

The model starts at the root node, where one of the dataset variables is evaluated. Then, according to the variable value, one of the output branches is selected to re-evaluate the data in a decision node. This process is repeated until the data reaches a response node where the sample is classified with the value associated with the leaf node. In general, the decision tree divides the data according to the value of its variables, so it is essential to find the optimal division boundaries. For this purpose, this method calculates each variable's entropy, or Gini index, to know its impurity degree. By estimating this value, the information gain value can be determined by comparing the impurity of the data set before and after the node splitting. Since the decision tree has a hierarchical structure, the tree is built from top to bottom using the variables with the highest information gain at the nodes of the first stages.

On the other hand, the decision trees algorithm generates models that are easy to understand and interpret; however, this technique cannot achieve good performance in complex problems since it generates large and complex trees that tend to cause overfitting.

2. Multi-Layer Perceptron

Artificial neural networks are one of the most widely used techniques in the field of soft computing. This method uses artificial neurons linked in layers to generate a structure of interconnected neurons that emulates the functioning of a human brain [40]. In this way, neural networks consist of an input layer, one or more hidden layers, and an output layer. Each of these layers is composed of one or more artificial neurons. These neurons sum the input values weighted by weights related to each input, and an independent value, also known as bias. Then, an activation function is applied to this value to obtain the neuron's output result.

Information flows through the network's hidden layers from the input to the output layer. In contrast, the training process is executed from the output layer to the input layer, applying a method known as backpropagation. The training process calculates the necessary gradients to optimize and adjust each network connection's weights.

Different network architectures can be developed depending on the configuration of the layers and the connections of layers and neurons. However, one of the simplest and most commonly used structures is the Multilayer Perceptron neural network (MLP), which is characterized by each neuron being connected to all the neurons of the next layer.

3. Random Forest

Random Forest (RF) is a well-known supervised machine learning technique commonly applied in classification and regression tasks based on implementing a certain number of decision trees [41].

Its performance is based on hiring a certain number of decision trees to generate a more accurate and robust model. Each random forest tree is different since it is trained with different random subsets selected from the training data. The Bootstrap Aggregation, or Bagging, is used to obtain the data subsets. This technique generates as many subsets as decision trees used in the model.

Finally, with each tree trained, Random Forest uses each decision tree to classify the input data. The classification of all the trees is then analyzed, and the most common prediction is taken as the model's output classification.

4. Support Vector Machine

Other well-known supervised techniques are the Support Vector Machines (SVM) developed by Cortes and Vapnik [42]. These methods are a group of machine learning algorithms often used for classification and regression tasks. The main objective of SVMs is to achieve a hyperplane that maximizes the minimum distance, known as the margin, between the hyperplane and the nearest samples of each class. This margin is used to determine a boundary for classifying new data samples.

The above SVMs definition assumes that a linear boundary can separate the classes. However, most real-world datasets are not linearly separable. To solve this problem, SVMs use data transformations, $\langle x_i, x_j \rangle \rightarrow \langle \phi(x_i), \phi(x_j) \rangle$, for mapping the data into a higher dimensional space, where a linear boundary can separate it. The specific transformation implemented, $\phi(x)$, depends on the kernel function selected.

5. Fisher Linear Discriminant

The Linear Discriminant Analysis (LDA), or Fisher Linear Discriminant Analysis, is a supervised classification machine learning technique developed by R.A. Fisher [43].

The main goal of the Fisher Linear Discriminant is to find the best linear combination of features that separates different training data classes as much as possible. Therefore, LDA searches out the hyperplane where the means of each class are as far apart as possible and the classes have the least variance in their data. The objective function, Equation (1) defined as $J(\theta)$ is maximized in the optimization process.

$$J(\theta) = \frac{(\mu_1 - \mu_2)^2}{\hat{s}_1^2 + \hat{s}_2^2} \tag{1}$$

where μ_1 and μ_2 are the mean value of class 1 and 2 respectively, and $\hat{s}_1$ and $\hat{s}_2$ correspond to the within-class variance 1 and 2.

6. Naive Bayes

Naive Bayes, also known as Naive Bayesian (NB), are straightforward machine learning methods, frequently used for classification issues, that are based on the Bayes statistical theorem. Additionally, these approaches presuppose that given the class, data properties are conditionally independent [44]. Although this assumption is generally excessively strong, Naive Bayes performance still produces outcomes that are very competitive and computationally efficient.

Under this technique, different algorithms can be applied. In the current research, Bernoulli and Gaussian methods have been tested.

Bernoulli Naive Bayes: Each feature is thought to correlate to a binary value. In this model, the probability is obtained using Equation (2).

$$P(x_i \mid c) = P(x_i \mid c)b_i + (1 - b_i)(1 - p(x_i \mid c)) \tag{2}$$

To use this approach, all data features must be binary; if a feature contains any other type of data, a binarization process is carried out.

Gaussian Naive Bayes: The numerical attribute values in Gaussian NB have a normal distribution and are shown concerning the mean and standard deviation. Equation (3) is used in this approach to determine the probability of the features.

$$P(x_i \mid c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} exp\left(-\frac{(x_i - \mu_c)^2}{2\sigma_c^2}\right) \tag{3}$$

where σ is the standard deviation and μ the mean value.

V. Experiments and Results

The present section describes the setup of the experiments and the results obtained.

A. Experiments Setup

This section provides the experiment configurations, including the tools and metrics used to measure and compare the performance of each classifier. The experiments were implemented using Python and several libraries such as Scikit-learn, Pandas, Numpy, TensorFlow and Keras.

To configure the experiments, the fundamental stages of machine learning problems summarized in Fig. 3, were followed. Each of these stages is described in detail below.

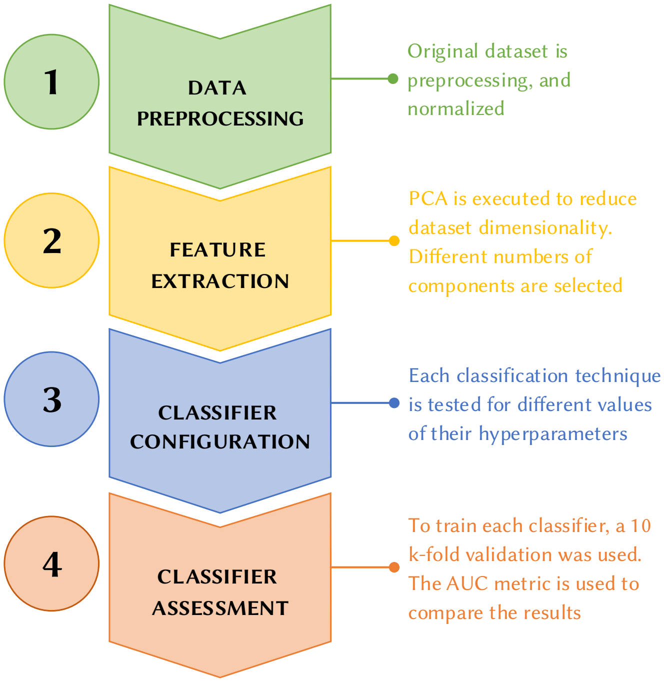

1
DATA PREPROCESSING
Original dataset is preprocessing, and normalized
2
FEATURE EXTRACTION
PCA is executed to reduce dataset dimensionality. Different numbers of components are selected
3
CLASSIFIER CONFIGURATION
Each classification technique is tested for different values of their hyperparameters
4
CLASSIFIER ASSESSMENT
To train each classifier, a 10 k-fold validation was used. The AUC metric is used to compare the results

Fig. 3. Experiment setup scheme.

1. Data Preprocessing

The first step was the preprocessing of the study case dataset. Once the data was analyzed, samples with missing data and constant variables for all the samples were removed from the dataset. On the other hand, the non-numerical variables were codified to numerical features, and finally, the data were normalized using the z-score method, with a mean value of 0 and a standard deviation of 1.

2. Feature Extraction

After preparing the dataset, the PCA technique was employed to reduce the number of features and choose the most important dataset variables. The number of principal components to retain was determined by analyzing the variance explained by each component, focusing on those that explain a significant amount of variance. In order to obtain the best classifier, both in terms of performance and computational cost efficiency, in this research, the experiments were carried out taking into account the different number of components.

3. Classifier Configuration

Each of the supervised classification techniques presented in Section IV has been tested for different configurations. Table II shows the hyperparameters that have been configured for each algorithm as well as the values that have been implemented. Each of the model's hyperparameters is briefly described below.

- Decision Trees (DT): for this technique, decision trees have been evaluated for different maximum depth parameters, from 5 to 50 layers of the tree with intervals of 5. In addition, it has also been tested, not indicating a maximum depth value ("None"). This way, the nodes are expanded until all leaves contain less than two samples.
- Multi-layer Perceptron (MLP): MLP neural networks have been analyzed for different network structures, considering the number of hidden layers, the number of neurons in the hidden layers, and the dropout percentage. The dropout corresponds to the middle layers used to control the regularization of the neural network and avoid overfitting problems. The following values have been taken into account for each parameter:
  - Number of hidden layers: 1, 2, and 3 hidden layers.
  - Number of neurons in hidden layers: 5, 10, 15, and 20 neurons per layer.
  - Dropout: 0 and 20%.

  It is important to note that the *ReLu* function was used as activation function in the neurons of the hidden layers and *Softmax* in the output layer. This configuration is commonly used in classification tasks with neural networks.
- Random Forest (RF): the parameter to be determined in this technique is the number of decision trees that conform the model. In this case, the algorithm performance was evaluated for models of 10 to 100 trees with increments of 10 trees.
- Support Vector Machines (SVM): in this case, different configurations of the Support Vector Machine have been tested by modifying the algorithm kernel, which indicates the transformation function, and the data regularisation factor, *C*. The strength of the regularisation is inversely proportional to *C*. The values used in these hyperparameters are:
  - Kernel: linear, polynomial, rbf (Radial Basis Function) and sigmoid.
  - Data regularization *C*: 0.001, 0.01, 0.1 and 1.
- Fisher Linear Discriminant (LDA): the performance of this technique has been tested for three different algorithms solvers (least squares, *lsqr*, singular value decomposition, svd and eigenvalue decomposition, *eigen*).
- Naive Bayes (NB): as already mentioned in IV.B.6 the performance of Bernoulli and Gaussian Naive Bayes have been evaluated.

4. Classifier Assessment

For the training process of each model, k-fold cross-validation with a k value of 10 has been used. In addition, the Area Under the receiving operating Curve (AUC) has been considered as the evaluation metric, which is widely used in classification tasks. The relationship between true positive and false positive rates is established by this parameter,

TABLE II.Configurations Tested

Evaluated technique	Evaluated configuration	Tested values
Decision Trees	Maximum depth	5:5:50 and None
Multi-layer Perceptron	Number of hidden layes Neurons in hidden layers Dropout (%)	1:1:3 5:5:20 0, 10
Random Forest	Number of trees	10:10:100
Support Vector Machine	Data regularization Kernel	0.001, 0.01, 0.1, 1 linear, poly, rbf, sigmoid
Fisher linear discriminant	Solver	svd, lsqr, eigen
Naive bayes	Algorithm	Bernoulli, Gaussian

which boasts two key benefits. Firstly, it offers a unified evaluation of classifier performance, and secondly, it remains unaffected by variations in class distribution.

On the other hand, the computational cost of each classifier implemented has also been measured. For this purpose, the average training time of each configuration of the models has been considered. In this sense, it must be taken into account that the experiments have been executed on a computer with an Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz and a RAM memory of 8GB.

B. Results

The results derived from the experimental setup outlined above are shown in this section. First, to determine the number of components to reduce the initial dataset, an initial Principal Component Analysis was executed to identify the components and their respective percentage of variance explained. Fig. 4 shows the results obtained in bar graph format.

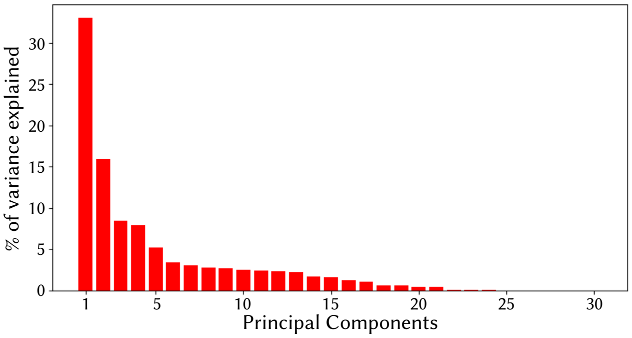


Fig. 4. PCA initial analysis.

Based on the achieved results, three different component selections will be considered for the experiments for evaluating the performance of combining dimensional reduction with the above-described classification techniques in terms of classification accuracy and computational cost. The three component selection criteria are as follows:

- Components with a percentage of explained variation greater than 10%: in this case the first 2 components are selected.
- Components with a percentage of variance explained greater than 5%: in this case the first 5 components are selected.
- Components with a percentage of explained variation greater than 0.01%: in this case the first 24 components are selected.

After selecting the different numbers of components to be used in each experiment, we trained and evaluated each technique's performance using the proposed configurations.

Before presenting the results, since the models were tuned and evaluated using k-fold cross-validation, it is essential to highlight that all the tables in this section depict the average AUC and training time values.

Table III presents the results obtained using decision trees. As can be seen, this technique achieved excellent results, exceeding 99% in terms of AUC, with the different configurations tested. Furthermore, it is noticeable that using fewer components significantly reduces the training time, with a slight loss in classifier performance, lower than 0.3% in terms of AUC. This technique has very low training times, less than 1 second in some of its configurations.

On the other hand, Table IV shows the results obtained with the Multi-Layer Perceptron neural networks (MLP). This technique exhibits high performance, reaching more than 99.8% of AUC in some configurations. In this case, reducing the number of components also reduces the classifier's performance. Comparing the results obtained, a reduction of more than 1% in terms of AUC can be produced using 24 or 2 components. On the other hand, the training time is not affected by the number of components, i.e., the number of neurons in the network's input layer. The network dimension, determined by the number of hidden layers and neurons per layer, is the main factor affecting computational cost. Additionally, it is observed that using dropout in the network does not improve classifier performance and significantly increases the training time, as it involves adding a new layer (the regularisation layer). Generally speaking, this technique presents a higher computational cost than decision trees.

Table V presents the performance of the Random Forest method for its different configurations. This technique achieves very good classifiers, with an AUC of over 99% in all configurations tested. Regarding the results, it can be observed that using a greater number of trees does not significantly improve the model's performance. For example, comparing the 100-tree model with the 10-tree model showed a difference of less than 0.1%. Similarly, the classifier's performance does not deteriorate significantly when using fewer components, achieving a reduction of 0.2% AUC when comparing models trained with 24 components to models adjusted with 2. However, models with fewer trees combined with a reduced number of selected components minimize the computational cost measured in training time, reducing it by more than 70% in some cases.

On the other hand, Table VI shows the performance of support vector machines. With this technique, very different results were obtained among the evaluated configurations. In general, it can be observed that using a reduced number of components greatly affects the classifier's performance, with a loss of more than 20% of AUC in many cases. Moreover, the best results are obtained with the highest value of the hyperparameter C, which implies low data regularisation. For this technique, the best model obtained was the one that uses the polynomial kernel with $C = 1$, which reaches a 98.32% AUC considering 24 components. Finally, highlight that reducing the number of components used does not reduce the training time. Compared to the other techniques, except for MLP, the computational cost of this technique is much higher.

The performance results of Fisher's Linear Discriminant Analysis are presented in Table VII. It can be observed that changing the algorithm's solver does not affect the classifier's performance, and this hyperparameter only influences the training time. In this regard, the svd (Singular value decomposition) method is the most computationally expensive compared to the other solvers tested. Additionally, when analyzing the impact of the number of components on the classifier's performance, it is evident that reducing the number of components significantly compromises the classifier's performance, lowering the AUC value and the training time. For this technique, the optimal classifier is obtained using the lsqr (Least squares solution) algorithm and components, which achieves over 89% AUC with an average training time of 0.159 seconds.

Finally, the Gaussian and Bernoulli naive Bayes were tested and the results are presented in Table VIII. With this technique, the Gaussian model fits better to the problem posed and performs better than the Bernoulli algorithm. On the other hand, it can be observed how considering a greater number of components improves the AUC result of the classifier and increases the training time.

Fig. 5 summarizes the best AUC results obtained for each of the techniques and the different number of components. This graph shows how using a greater number of components improves the results measured by the AUC metric and how the best classifiers are obtained with the Decision Trees, Multi-layer Perceptron, and Random Forest.

TABLE III. Decision Trees Results

Model setup \ PCA	2 components		5 components		24 components	
Nº of trees	AUC (%)	T. time (s)	AUC (%)	T. time (s)	AUC (%)	T. time (s)
5	98.27	0.064	98.43	0.135	98.83	0.705
10	99.00	0.098	99.02	0.224	99.24	1.263
15	99.09	0.112	99.19	0.262	99.30	1.679
20	99.09	0.116	99.24	0.273	99.33	2.013
25	**99.10**	0.116	99.25	0.278	99.35	2.042
30	99.09	0.115	99.28	0.276	99.33	2.038
35	99.10	0.118	99.27	0.276	99.34	2.038
40	99.10	0.117	**99.29**	0.275	**99.35**	2.037
45	99.09	0.117	99.27	0.275	99.34	2.038
50	99.09	0.117	99.27	0.275	99.35	2.038
None	99.09	0.117	99.28	0.280	99.33	2.039

TABLE IV. Multilayer Perceptron Results

Model setup \ PCA			2 components		5 components		24 components	
Nº of hidden layers	Nº of neurons	Dropout (%)	AUC (%)	T. time (s)	AUC (%)	T. time (s)	AUC (%)	T. time (s)
1	5	0	96.22	20.755	99.11	20.714	99.57	20.786
1	5	10	96.04	21.794	97.33	21.933	99.45	22.043
1	10	0	98.20	21.259	99.27	21.116	99.68	21.405
1	10	10	98.18	22.446	99.28	22.560	99.68	22.884
1	15	0	98.25	21.496	99.30	21.328	99.75	21.579
1	15	10	98.22	22.680	99.29	22.810	99.70	23.044
1	20	0	98.41	22.077	99.35	21.936	99.76	22.032
1	20	10	98.33	24.207	99.39	23.131	99.74	23.330
2	5	0	93.35	22.652	92.48	22.663	99.58	22.250
2	5	10	96.39	24.766	99.07	27.514	99.46	24.297
2	10	0	98.37	22.977	99.35	23.039	99.73	22.941
2	10	10	96.50	25.435	99.37	25.471	99.68	25.243
2	15	0	98.47	23.459	99.42	23.508	99.73	23.286
2	15	10	98.50	25.942	99.41	26.003	99.76	25.722
2	20	0	98.62	24.099	99.47	23.861	99.79	23.591
2	20	10	98.51	26.379	99.42	26.391	99.78	26.056
3	5	0	98.12	24.411	93.58	24.170	99.59	23.799
3	5	10	95.40	27.229	99.13	26.911	99.49	26.760
3	10	0	98.43	25.194	99.43	24.801	99.73	24.457
3	10	10	98.22	28.644	99.34	27.994	99.73	27.849
3	15	0	98.65	25.787	99.53	25.205	99.79	25.111
3	15	10	98.52	29.346	99.45	28.693	99.76	28.716
3	20	0	98.79	26.494	99.48	25.795	99.82	25.287
3	20	10	98.52	30.360	99.47	29.480	99.80	29.156

TABLE V. Random Forest Results

Model setup \ PCA	2 components		5 components		24 components	
Nº of trees	AUC (%)	T. time (s)	AUC (%)	T. time (s)	AUC (%)	T. time (s)
10	99.19	0.594	99.29	0.718	99.36	2.052
20	99.20	1.211	**99.33**	1.539	99.39	4.221
30	99.19	1.839	99.31	2.013	99.38	6.209
40	99.19	2.435	99.32	2.665	99.42	6.655
50	99.18	3.248	99.32	3.278	99.40	8.994
60	99.20	3.385	99.32	3.933	99.42	11.621
70	99.19	4.021	99.30	4.610	99.41	12.580
80	99.19	4.160	99.32	5.281	99.41	12.646
90	**99.21**	3.922	99.31	6.216	**99.44**	14.138
100	99.21	4.393	99.32	6.599	99.41	15.604

TABLE VI. Support Vector Machine Results

Model setup \ PCA		2 components		5 components		24 components	
Kernel	Reg.	AUC (%)	T. time (s)	AUC (%)	T. time (s)	AUC (%)	T. time (s)
linear	1	73.50	37.442	84.47	52.600	90.00	34.126
linear	0.1	73.50	25.434	79.90	28.180	89.97	23.385
linear	0.01	73.50	20.351	74.66	21.244	90.07	23.605
linear	0.001	73.50	19.763	74.64	21.188	89.20	28.774
poly	1	73.69	29.088	90.72	17.263	**98.32**	16.225
poly	0.1	73.69	49.309	75.52	19.876	91.04	22.407
poly	0.01	73.70	26.095	74.87	25.077	75.56	29.913
poly	0.001	73.71	21.021	74.86	25.688	75.55	37.652
rbf	1	**88.33**	26.963	**91.04**	18.143	98.26	16.359
rbf	0.1	73.73	30.471	90.27	25.654	98.27	21.541
rbf	0.01	73.50	27.578	74.65	30.738	90.46	43.467
rbf	0.001	73.50	30.304	74.65	39.303	73.29	62.536
sigmoid	1	71.15	27.657	71.16	26.914	85.77	46.046
sigmoid	0.1	73.03	27.566	73.48	29.337	88.95	49.293
sigmoid	0.01	78.27	40.877	79.53	38.644	82.70	55.067
sigmoid	0.001	73.23	40.016	74.61	49.282	73.49	58.673

TABLE VII. Fisher Linear Discriminant Results

Model setup \ PCA	2 components		5 components		24 components	
Solver	AUC (%)	T. time (s)	AUC (%)	T. time (s)	AUC (%)	T. time (s)
svd	73.51	0.041	74.52	0.049	89.27	0.236
lsqr	73.51	0.038	**74.52**	0.040	**89.27**	0.159
eigen	**73.51**	0.037	74.52	0.042	89.27	0.178

TABLE VIII. Naive Bayes Results

Model setup \ PCA	2 components		5 components		24 components	
Algorithm	AUC (%)	T. time (s)	AUC (%)	T. time (s)	AUC (%)	T. time (s)
Bernoulli	73.51	0.026	75.11	0.022	75.19	0.047
Gaussian	**80.37**	0.023	**78.80**	0.021	**83.55**	0.054

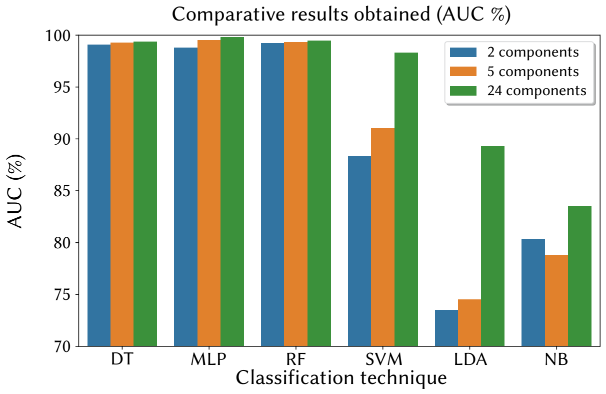


Fig. 5. Comparison of results.

VI. Conclusions and Future Work

This research analyses the performance of six supervised classification techniques in combination with the PCA dimensional reduction method to detect DoS attacks in data networks working with the MQTT protocol. The obtained results have been highly promising, reaching AUC values higher than 95% except for the LDA and Naive Bayes methods that have achieved, for their best configuration, a maximum of 89.27% and 83.55% AUC, respectively.

Considering only the classifier performance, MLP neural networks have been shown to detect better DoS attacks reaching 99.82% AUC for the network with 3 hidden layers, 20 neurons per layer, and without dropout layers. However, the computational cost of this technique, with a mean average training time of 25 seconds, is significantly higher than other methods that have also demonstrated excellent performance, such as, for instance, Decision Trees, with a maximum of 99.35% AUC and training times between 0.1 and 2 seconds, or Random Forest with more than 99% AUC and training times between 0.6 and 15 seconds depending on the selected configuration. SVMs also performed well in many configurations with values above 98% AUC but with training times above 30 seconds. Therefore, considering a computational performance-cost relationship, it can be concluded that decision trees are the best technique.

On the other hand, comparing the results obtained by using a different number of components, it was observed that a significant reduction in the number of components can worsen the classifier's performance and reduce the model's training times. For Decision Trees, a maximum loss of 0.3% AUC is quite optimal when compared to the substantial reduction in training time, often exceeding 90%in some cases. This aspect is also similarly reflected in the Random

Forest models, in which component reduction greatly reduces the computational cost with minimal loss of classifier performance. However, in SVMs and MLP neural networks, using a smaller number of components does not reduce the training time and worsens the performance of these techniques.

Thanks to the high performance of the models achieved, these can be deployed in an IDS for detecting anomalous network behaviours, preventing attacks.

In future works, we will study the performance of other supervised and unsupervised classification techniques and other feature extraction methods to compare their performance against the proposal shown in this paper. Additionally, it will also be considered to test the performance of our proposal for detecting Denial of Service attacks in other types of IoT protocols, such as CoAP and LoRa, among others. On the other hand, the possibility of detecting other types of attacks in this protocol will also be studied. Finally, the development of an intelligent hybrid system capable of detecting different attacks in different IoT network protocols will be analyzed, making it possible to standardize and offer a handy tool for the field of cybersecurity.

Acknowledgment

Álvaro Michelena's research was supported by the Spanish Ministry of Universities (https://www.universidades.gob.es/), under the "Formación de Profesorado Universitario" grant with reference FPU21/00932. Spanish National Cybersecurity Institute (INCIBE) and developed Research Institute of Applied Sciences in Cybersecurity (RIASC). CITIC, as a Research Center of the University System of Galicia, is funded by Consellería de Educación, Universidade e Formación Profesional of the Xunta de Galicia through the European Regional Development Fund (ERDF) and the Secretaría Xeral de Universidades (Ref. ED431G 2019/01).

References

[1] T. M. Ghazal, M. K. Hasan, M. T. Alshurideh, H. M. Alzoubi, M. Ahmad, S. S. Akbar, B. Al Kurdi, I. A. Akour, "Iot for smart cities: Machine learning approaches in smart healthcare—a review," *Future Internet*, vol. 13, no. 8, 2021, doi: 10.3390/fi13080218.

[2] P. K. Malik, R. Sharma, R. Singh, A. Gehlot, S. C. Satapathy, W. S. Alnumay, D. Pelusi, U. Ghosh, J. Nayak, "Industrial internet of things and its applications in industry 4.0: State of the art," *Computer Communications*, vol. 166, pp. 125–139, 1 2021, doi: 10.1016/j.comcom.2020.11.016.

[3] M. Rothmuller, S. Barker, "Iot the internet of transformation 2020," *Juniper Research, Basingstoke, UK, Whitepaper*, 2020.

[4] M. Ahmad, T. Younis, M. A. Habib, R. Ashraf, S. H. Ahmed, "A review of current security issues in internet of things," *Recent Trends and Advances in Wireless and IoT-enabled Networks*, pp. 11–23, 2019, doi: 10.1007/978-3-319-99966-2.

[5] M. H. Khalid, M. Murtaza, M. Habbal, "Study of security and privacy issues in internet of things," *CITISIA 2020 - IEEE Conference on Innovative Technologies in Intelligent Systems and Industrial Applications, Proceedings*, 11 2020, doi: 10.1109/CITISIA50690.2020.9371828.

[6] B. Kepçeoğlu, A. Murzaeva, S. Demirci, "Performing energy consuming attacks on iot devices," in *2019 27th Telecommunications Forum (TELFOR)*, 2019, pp. 1–4.

[7] J. Granjal, E. Monteiro, J. S. Silva, "Security for the internet of things: A survey of existing protocols and open research issues," *IEEE Communications Surveys and Tutorials*, vol. 17, pp. 1294–1312, 2015, doi: 10.1109/COMST.2015.2388550.

[8] R. Yugha, S. Chithra, "A survey on technologies and security protocols: Reference for future generation iot," *Journal of Network and Computer Applications*, vol. 169, p. 102763, 11 2020, doi: 10.1016/j.jnca.2020.102763.

[9] Y. Lu, L. D. Xu, "Internet of things (iot) cybersecurity research: A review of current research topics," *IEEE Internet of Things Journal*, vol. 6, pp. 2103–2115, 4 2019, doi: 10.1109/JIOT.2018.2869847.

[10] J. Tournier, F. Lesueur, F. L. Mouël, L. Guyon, H. Ben-Hassine, "A survey of iot protocols and their security issues through the lens of a generic iot stack," *Internet of Things*, vol. 16, p. 100264, 12 2021, doi: 10.1016/J.IOT.2020.100264.

[11] E. Džaferović, A. Sokol, A. A. Almisreb, S. M. Norzeli, "Dos and ddos vulnerability of iot: a review," *Sustainable Engineering and Innovation*, vol. 1, no. 1, pp. 43–48, 2019.

[12] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, *et al.*, "Understanding the mirai botnet," in *26th USENIX security symposium (USENIX Security 17)*, 2017, pp. 1093–1110.

[13] M. H. Khalid, M. Murtaza, M. Habbal, "Study of security and privacy issues in internet of things," in *2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA)*, 2020, pp. 1–5, IEEE.

[14] T. A. Idriss, H. A. Idriss, M. A. Bayoumi, "A lightweight puf-based authentication protocol using secret pattern recognition for constrained iot devices," *IEEE Access*, vol. 9, pp. 80546–80558, 2021, doi: 10.1109/ACCESS.2021.3084903.

[15] S. Amanlou, M. K. Hasan, K. A. A. Bakar, "Lightweight and secure authentication scheme for iot network based on publish–subscribe fog computing model," *Computer Networks*, vol. 199, p. 108465, 11 2021, doi: 10.1016/J.COMNET.2021.108465.

[16] X. Zhu, H. Deng, "A security situation awareness approach for iot software chain based on markov game model," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, pp. 59–65, 2022, doi: 10.9781/ijimai.2022.08.002.

[17] D. Choudhary, R. Pahuja, "Improvement in quality of service against doppelganger attacks for connected network," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, pp. 51–58, 2022, doi: 10.9781/ijimai.2022.08.003.

[18] R. Berjón, M. Mateos, M. E. Beato, A. F. García, "An event mesh for event driven iot applications," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, pp. 54–59, 2022, doi: 10.9781/ijimai.2022.09.003.

[19] H. J. Liao, C. H. R. Lin, Y. C. Lin, K. Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, pp. 16–24, 1 2013, doi: 10.1016/J.JNCA.2012.09.004.

[20] L. Aversano, M. L. Bernardi, M. Cimitile, R. Pecori, "A systematic review on deep learning approaches for iot security," *Computer Science Review*, vol. 40, p. 100389, 2021.

[21] A. Khraisat, A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, pp. 1–27, dec 2021, doi: 10.1186/s42400-021-00077-7.

[22] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, p. e4150, 2021.

[23] S. Andy, B. Rahardjo, B. Hanindhito, "Attack scenarios and security analysis of mqtt communication protocol in iot system," in *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2017, pp. 1–6.

[24] D. H. Deshmukh, T. Ghorpade, P. Padiya, "Intrusion detection system by improved preprocessing methods and naïve bayes classifier using nsl-kdd 99 dataset," in *2014 International Conference on Electronics and Communication Systems (ICECS)*, 2014, pp. 1–7.

[25] M. Esmaeili, S. H. Goki, B. H. K. Masjidi, M. Sameh, H. Gharagozlou, A. S. Mohammed, "Ml-ddosnet: Iot intrusion detection based on denial-of-service attacks using machine learning methods and nsl- kdd," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–16, 8 2022, doi: 10.1155/2022/8481452.

[26] J. Liu, B. Kantarci, C. Adams, "Machine Learning- Driven Intrusion Detection for Contiki-NG-Based IoT Networks Exposed to NSL-KDD Dataset," in *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, New York, NY, USA, 2020, ACM.

[27] P. Sethi, S. R. Sarangi, "Internet of things: architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, vol. 2017, 2017, doi: 10.1155/2017/9324035.

[28] K. Ramamoorthy, S. Karthikeyan, T. Chelladurai, "An investigation on

industrial internet of things for mission critical things in industry 4 . 0 2 . literature review," *Seybold Report*, vol. 15, pp. 3294–3300, 2020.

[29] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, X. Bellekens, "Machine learning based iot intrusion detection system: An mqtt case study (mqtt-iot-ids2020 dataset)," in *International Networking Conference*, 2020, pp. 73–84, Springer.

[30] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, A. N. Anwar, "Ton-iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862.

[31] VMware, "Vmware nsx data center datasheet." [Online]. Available: https://kb.vmware.

[32] J. Deogirikar, A. Vidhate, "Security attacks in iot: A survey," *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*, pp. 32–37, 2017, doi: 10.1109/I-SMAC.2017.8058363.

[33] "GitHub - moscajs/aedes: Barebone MQTT broker that can run on any stream server, the node way." [Online]. Available: https://github.com/moscajs/aedes.

[34] K. Palsson, "mqtt-malaria @ github.com," 2018. [Online]. Available: https://github.com/remakeelectric/mqtt-malaria.

[35] J. Aveleira-Mata, H. Alaiz-Moreton, "Functional prototype for intrusion detection system oriented to intelligent iot models," in *International Symposium on Ambient Intelligence*, 2019, pp. 179–186, Springer.

[36] "MQTT Dataset LE-229-18," 2019. [Online]. Available: https://joseaveleira.es/dataset.

[37] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, vol. 2, no. 11, pp. 559–572, 1901.

[38] H. Abdi, L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[39] L. Rokach, O. Maimon, "Decision trees," in *Data mining and knowledge discovery handbook*, Springer, 2005, pp. 165–192.

[40] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, 2018.

[41] A. Cutler, D. R. Cutler, J. R. Stevens, "Random forests," in *Ensemble machine learning*, Springer, 2012, pp. 157– 175.

[42] C. Cortes, V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[43] J. Yang, Z. Jin, J.-y. Yang, D. Zhang, A. F. Frangi, "Essence of kernel fisher discriminant: Kpca plus lda," *Pattern Recognition*, vol. 37, no. 10, pp. 2097–2100, 2004.

[44] I. Rish, *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, 2001, pp. 41–46.

Álvaro Michelena

Álvaro Michelena is a Ph.D. student in Computational Science at the University of A Coruña, Spain, su. He received a M.S. in Industrial Computing and Robotics from the University of A Coruña in 2021. He has worked for a year and a half as a Research Assistant at the Centre for Information and Communications Technology Research (CITIC) of the University of A Coruña where he has collaborated on different research projects. His main research areas are related to applying intelligent techniques for anomaly detection and system modeling.

Jose Aveleira-Mata

Jose Aveleira-Mata is a Ph.D. student on Production and Computing Engineering at the University of Leon, Spain. His research interests include Internet of Things, Cloud Computing, Wireless Sensor Networks and Network Security. He has several papers published in international conferences, as well as scientific publications in JCR journals, on topics related to cybersecurity.

Esteban Jove

Esteban Jove received a M.S. degree in Industrial Engineering from the University of Leon in 2014. After two years working in the automotive industry, he joined the University of A Coruña, Spain, where he has been Assistant Professor of Power Electronics in the Faculty of Engineering since 2016. He received his Ph.D at the University of La Laguna in 2020, and his research has been focused on the use of intelligent techniques for nonlinear systems modelling and anomaly detection using one-class techniques.

Héctor Alaiz-Moretón

Héctor Alaiz-Moretón received his degree in Computer Science, performing the final Project at Dublin Institute of Technology, in 2003. He received his PhD in Information Technologies in 2008 (University of Leon). He has worked as a lecturer since 2005 at the School of Engineering at the University of Leon. His research interests include knowledge engineering, machine and deep learning, networks communication, and security. He has several works published in international conferences, as well as books, more than 90 scientific publications between JCR papers, Lecture Notes and Scientific Workshops. He has been a member of scientific committees in conferences. He has headed several PhD Thesis and research competitive projects. Actually, he is the vice main of RIASC (Institute of Applied Sciences to Cybersecurity).

Héctor Quintián

Héctor Quintián is currently Assistant Professor at University of A Coruña (UDC). Along his academic career, he has published many papers in national and international scientific journals and several book chapters. As far as research activity is concerned, it is worth highlighting his publication activity; over the last 10 years, 62 research papers in journals indexed with relative quality index, all of them in the JCR. Around 70% of them have been published in journals located in the first two quartiles of their categories. He has published a total of 65 contributions in conferences of which 80% correspond to international conferences, most of them indexed at the CORE ranking and at the GII-GRIN-SCIE (GGS) Conference Rating. In addition, it has organized a large number of scientific conferences in various editions (40), all of them of recognized international prestige, His main research lines are focused on artificial intelligence, and not supervised learning developing several algorithms with application to industrial modelling systems.

José Luis Calvo-Rolle

José Luis Calvo-Rolle received M.S. and Ph.D. degrees in Industrial Engineering from the University of Leon in 2004 and 2007, respectively. He is Full Professor in the Systems Engineering and Automation Area, of the Industrial Engineering Department, University of A Coruña. Currently, he is the director of that department and the head of the Environmental Radioactivity Laboratory. In addition, he coordinates the Cybernetic Science and Technology Research Group. His main research areas are focused on the application of intelligent techniques and systems for optimization, diagnosis, modeling and control.

A Survey on Demand-Responsive Transportation for Rural and Interurban Mobility

Pasqual Martí[1], Jaume Jordán[1], Angélica González Arrieta[2], Vicente Julian[1,3] *

[1] Valencian Research Institute for Artificial Intelligence (VRAIN), UniversitatPolitècnica de València, Valencia (Spain)
[2] Department of Computer Science and Automation, University of Salamanca,Salamanca (Spain)
[3] Valencian Graduate School and Research Network of Artificial Intelligence, Universitat Politècnica de València, Valencia (Spain)

Received 26 February 2023 | Accepted 24 July 2023 | Published 31 July 2023

Abstract

Rural areas have been marginalized when it comes to flexible, quality transportation research. This review article brings together papers that discuss, analyze, model, or experiment with demand-responsive transportation systems applied to rural settlements and interurban transportation, discussing their general feasibility as well as the most successful configurations. For that, demand-responsive transportation is characterized and the techniques used for modeling and optimization are described. Then, a classification of the relevant publications is presented, splitting the contributions into analytical and experimental works. The results of the classification lead to a discussion that states open issues within the topic: replacement of public transportation with demand-responsive solutions, disconnection between theoretical and experimental works, user-centered design and its impact on adoption rate, and a lack of innovation regarding artificial intelligence implementation on the proposed systems.

Keywords

Artificial Intelligence, Demand-Responsive, Intelligent Transportation, Optimization, Rural Areas.

DOI: 10.9781/ijimai.2023.07.010

I. Introduction

Access to public transportation (PT) should be generalized, as its name implies. Rural communities are often marginalized, with citizens only accessing low-quality PT. Some of the characteristics associated with rural PT are old vehicles, long and infrequent routes, and inconvenient stops. Therefore, it is common to observe higher ownership of personal motor vehicles in rural settlements (2 per household versus 1 in cities) [1].

The demand for transportation in rural areas differs from that in urban areas. It is characterized by more scattered transport requests, both in time and space, which makes the economic viability of higher-quality services more difficult. Consequently, with this shape of demand, it seems difficult to justify deploying a transport that continuously offers service, with or without passengers. Because of that, the on-demand transportation paradigm shows potential for reducing costs while increasing service quality in rural areas.

Demand-responsive transportation (DRT) systems offer displacement services adapted to the needs of their users. Initially conceived as a mobility option for impaired people and inhabitants of isolated areas [2], this mode of transport is again attracting PT providers' interest thanks to technological advances that allow users to be connected most of the time. DRT systems count on two main characteristics: on-demand mobility and adaptable flexibility. According to the specific configuration, DRT can resemble transportation ranging from high-capacity interurban buses to dial-a-ride urban taxis [3]. Thus, given a use case, it is necessary to analyze which implementation best fits the needs of the potential customers. In practice, however, implemented DRT services have a relatively high failure rate, caused by high economic costs [4], [5] and low customer acceptance, among others. In addition, the success of a concrete DRT deployment depends on the characteristics of the area it services, its population density, demand, and current transportation trends. The implementation of demand-responsive mobility has been highly studied in recent years, although mostly applied to urban contexts [6].

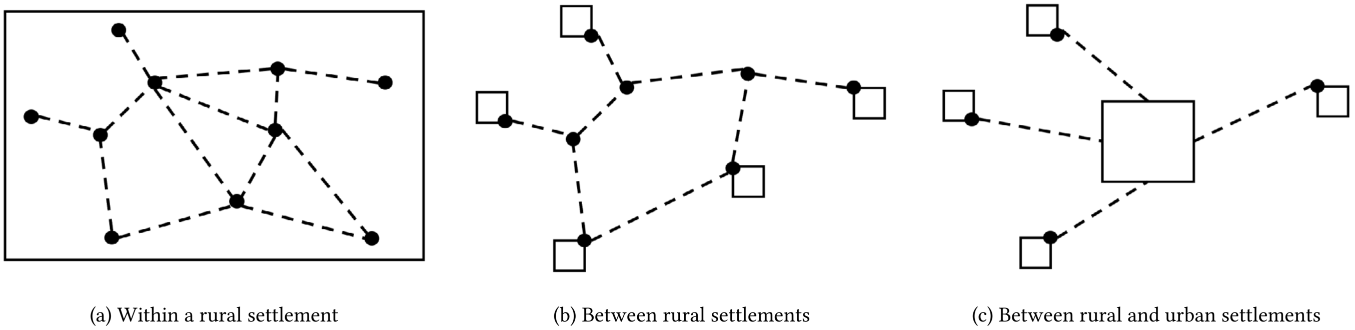
In this review article, we bring together papers that discuss, analyze, model, or experiment with DRT systems applied to rural areas and interurban transportation, with the intention of discussing their general feasibility as well as the most successful configurations. Political authorities from different parts of the world have shown their interest in the improvement of rural transport with a sustainable perspective. The Spanish government, for example, has presented within its "mobility strategy" the Rural Mobility Roundtable[1] where it highlights, among others, the importance of demand-driven transport and the creation of dynamic routes to work towards the goal of generalized access to PT in rural areas.

The rest of the paper is structured as follows. Section II describes DRT systems and their components, introducing the challenges its

* Corresponding author.

E-mail addresses: pasmargi@vrain.upv.es (P. Martí), jjordan@dsic.upv.es (J. Jordán), angelica@usal.es (A. González Arrieta), vinglada@dsic.upv.es (V. Julian).

[1] https://esmovilidad.mitma.es/mesa-de-movilidad-rural (Accessed on 01/12/2022)


(a) Within a rural settlement

(b) Between rural settlements

(c) Between rural and urban settlements

Fig. 1. Observed operational patterns for rural demand-responsive transportation systems. Boxes indicate rural/urban settlements. Black dots represent stops. Dashed lines represent demand-responsive lines. Pictures (a) and (b) are cases of many-to-many transportation, while (c) represents a many-to-one model.

implementation involves. Section III classifies the state-of-the-art work dividing it into analyses and proposals and summarizing each of the cited works. Section IV discusses the results and insights of the reviewed publications, with a particular fixation on the observed open issues. Finally, Section V concludes the review article by summarizing the state of DRT research and stating the main takeaway points of the present work.

II. Definitions and Problem Description

This section describes DRT and provides the necessary definitions for the posterior classification of rural-DRT publications. First, we characterize demand-responsive systems according to their configuration. Then, the modeling and optimization techniques that are classically applied to works in the area are commented. Finally, some insight is given regarding the optimization perspective that different DRT researchers follow.

A. Demand-Responsive Transportation Characteristics

DRT systems have a series of standard elements present in all of them. Different authors apply different labels to those elements. For the current work, we have followed the terminology described in this survey [7].

In a DRT system, a *service* is the departure of a vehicle to serve the transportation requests it has assigned. One service is generally tied to a concrete area or line assigned to the transport. In contrast, a *route* is the specific path the vehicle follows connecting all the pickups and drop-offs. A route does not necessarily include all existing stops in a line or area. Customers are picked up and dropped off in a predefined set of *stops* within the serviced area or line. Alternatively, a *door-to-door* service can be offered, in which any user-specified location within a particular area may act as a stop. This type of mobility is thought to be *shared*; i.e.: multiple customers are served by the same vehicle. Typical vehicle choices for demand-responsive services include a taxi-like car with a capacity of 4 passengers, mini-vans with 9 to 12 seats, and mini-buses or buses with 20 to 30 seats, respectively.

Many operational patterns exist for DRT. Specifically, for rural-DRT, we find the following: transportation within rural settlements, transportation between rural settlements, and transportation between rural and urban settlements. In practice, these cases can be reduced to two systems: *many-to-many*, with a set of multiple origins and destination locations, and *many-to-one*, where origin and destination locations share a unique pick-up or drop-off point. The last type is usually the so-called feeder line, where a flexible transportation service is used to move passengers to a different, less accessible service (for instance, communications from rural settlements to an airport). Fig. 1 shows a schematic representation of the commented used cases.

If the customer is required to send a *request* to access transport the service is provided *on-demand*. The time between sending a request and the customer's pick up is the *lead time*, and it is used to adapt the fleet operation or *planning* to include such a request. In a stop-based operation, the customer will be assigned a stop from which it will be picked up. On-demand systems can operate in real-time, accepting last-minute bookings, or with a hybrid approach, accepting bookings in advance too. DRT systems which are not on-demand are also possible. These systems consider current demand or demand predictions for service planning but do not require requests to run.

The period of time for which the DRT service is planned and optimized is referred to as *planning horizon*. The duration of planning horizons is usually a whole day. In addition, the operator may plan for a few hours to adapt to high/low demand periods. According to the influence of the demand data on the service planning, the system will be *fully-flexible* if routes are planned from scratch according to current demand or *semi-flexible* if a predetermined plan exists but vehicles are allowed to modify it influenced by demand.

B. Modeling and Optimization Techniques

Once the specific type of DRT system has been chosen, it must be modeled and tested to check its performance and adjust its attributes. We discuss below the different steps this involves, citing relevant research and the methods their authors employ. Please be aware that not every paper cited in this section explores rural-DRT.

Most DRT works are set in a specific settlement or area. In general, the main transportation network (roads, highways) of the area is mirrored thanks to services like Open Street Map (openstreetmap.org) or Open Sourcing Routing Machine (OSRM, project-osrm.org) [8]. Ideally, the actual organization of the area, its types of districts, population, or socio-economic reality, among others, should also be considered. Authors in [9] describe a seven-step analysis method for the optimization of any transportation system, based on reproducing the features of the currently implemented transport service (that would potentially be replaced). Alternatively, some works employ grid-like modelings of the area where the system will run [10]. The actual routing of each fleet vehicle represents one of the main challenges of DRT services, as it must be performed in real time. Innovative heuristic algorithms [11], [12] aid in this respect.

Demand modeling is also crucial. Passenger demand has two main aspects: (1) frequency and intensity and (2) shape (location of origin-destination pairs). Demand attributes can be extracted from datasets of different transportation modes and extrapolated, as in [13], where taxi data is used. Moreover, real data of pilot DRT services [14], [15] can be reproduced when available. However, the most observed technique is the use of synthetic demand data that can be generated statistically [10], based on socio-demographic information [16], via surveys [8], [15], [17] or generated in a (semi-)random [18] way according to the properties of the reproduced area (population, age, occupation, vehicle ownership). Finally, if traffic intensity data is available, it is useful to include it in the model, although not as relevant for rural areas with respect to city-centered studies, since the former tend to have lower intensity.

The operation of the DRT system requires automated planning and scheduling of vehicle services. At the same time, these tasks need information on the time and traveled kilometers that any detour would imply, which makes routing algorithms also necessary. In addition, since it is common to find online systems that accept real-time requests, the computation time for detours and new request insertions must be kept low. The use of multimodal planning [8] is common to solve the scheduling of vehicle services. Moreover, some simulation platforms, such as MATSim [19] include their own implementations of the algorithms mentioned above. These implementations usually employ (meta)heuristic techniques [16] that optimize vehicle-passenger assignments (insertion heuristics [20], for instance) or vehicle routing in a short computational time. Besides that, other less exploited techniques such as automated negotiation could be used to decide assignments from a decentralized perspective [21].

Finally, to observe the system's dynamics and its operation and adjust its attributes, it is necessary to simulate it. This can be performed through mathematical modeling [9] provided detailed data is available. However, a more popular way of achieving this is through multi-agent simulation (MAS). Among the observed choices we find NetLogo [22], used in [23], the already mentioned MATSim, and even custom simulators [8], [24].

C. Optimization Goals

The main goal of people transportation services is to supply the displacement needs of its users. Ideally, the operation of the service shall be performed by optimizing three factors: (1) the economic viability of the service; (2) the customer's experience (or quality of service); and (3) the environmental impact of the service. These three factors are translated into scopes when it comes to transportation research, and thus we can find works that asses one (only operator perspective [25]), or many of them from a multi-objective perspective (passenger and operator perspectives [26]). Optimizing customer experience implies reducing passenger travel times, whereas economic viability is ensured by reducing operational costs. Finally, optimizing sustainability requires reducing vehicle traveled kilometers (VTK).

The greatest challenge of demand-responsive transportation systems is finding the equilibrium among the above factors to offer a competitively-priced, economically viable, and flexible mobility alternative to private cars and traditional public transportation. For the case of rural-DRT, economic viability is especially difficult, taking into account the relatively low demand.

III. State of the Art Classification

This section presents a classification of the relevant literature found while researching the topic. Given the heterogeneity observed among the articles, they have been grouped by two criteria. On the one hand, the first group encapsulates studies, surveys, and analyses on the implantation of DRT solutions for rural areas. On the other hand, the second group presents papers that include at least an explicit DRT system proposal and experimentation to evaluate it. Both types of work offer reflections and insights into the viable application of on-demand mobility to areas with scattered populations and low demand.

A. Literature Retrieval and Overview

The Google Scholar and Scopus search engine were used to retrieve articles and book chapters relevant to the topic. The results were filtered applying the following rules: 1) the term "demand-responsive" had to be present in the title, abstract, or keywords of the publication, and 2) at least one of the terms "rural", "rural area" or "interurban" had to be present in the title, abstract, or keywords of the publication. Using the above criteria, the first search yielded 34 articles. Of these, 9 were discarded because the algorithms or systems they described did not fit the rural perspective of our review paper. The keywords "rural" and "interurban" could be present in the abstracts, but that did not guarantee that the characteristics of the systems researched by the authors matched those of rural or interurban mobility. Therefore, only papers that explicitly modeled low demand with scattered residents or assessed a rural interurban scenario were retained. Once filtered, the batch of relevant publications had a relatively small size of 25 publications. The fact is that rural-DRT solutions are less explored than their urban counterparts, probably because of factors such as scarce data availability and a lack of general interest until recent times.

In addition to the few publications, the degree of detail regarding the DRT systems described in them varied considerably. In general, all authors describe at least the operation of the basic components of any transportation system. However, just a minority explicitly state their system's constraints, the objective function(s), or the technology employed to build their proposals. Finally, it is worth mentioning that each proposal is tailored to the rural area it serves, which also differs for each work.

Given the described situation, we have chosen to summarize the publications on this topic one by one, giving as much relevant detail for each of them as possible. Nevertheless, two main classification criteria have been applied to divide the publications: analytical works, discussing challenges and studying the implementation of DRT in a specific context (Section III.B); and experimental works that explicitly model, implement and simulate a DRT system (Section III.C).

B. Analyses and Surveys

This subsection groups the state-of-the-art literature which assesses the challenges, potential benefits and contributions of implementing DRT for rural mobility. Most of the cited works develop their analyses around a main topic, which is shared among some, but present their own methods and conclusions. Following, we present the contributions grouped by the main topic they discuss.

1. Success and Failure of DRT Systems

One of the most historically studied topics in DRT history is the success and failure of deployed systems. Works in this line give important insight that PT providers must consider when designing a system. Enoch et al. assess the failure of DRT systems in [5]. The authors concluded that DRT projects are often not realistically costed or designed with a full understanding of the market they are to serve. A pattern was observed in which providers offered too flexible a service, including costly technological systems, when they may not be needed. In contrast, the authors recommend an incremental approach as a more sensible option. Compared to conventional PT operations, DRT requires more marketing effort and skills, but above all, it requires new skills in working in partnership. The failure in partnerships is where the root of DRT failure is often found.

G. Curry and N. Fournier [4] review DRT and Micro-Transit implementations to assess their performance. High failure rates stand out in their findings. 50% of the systems last less than 7 years, 40% last less than 3 years, and about a quarter fail within 2 years. In the UK, 67% of DRT services have failed, and in Australasia, 54%. The results indicate that simpler operations (e.g., many-to-few or route deviation) had lower failure rates compared to more complex many-to-many services. The authors develop a cost analysis that shows a strong and definitive link between DRT failure and higher service costs.

2. Replacing Classic PT With DRT

Many analyses focus on a particular rural settlement and aim to replace or optimize the currently implemented means of PT. Ryley et al. [2] investigate the contributions of DRT to sustainable PT. Their study surveys the public of both urban (Rochdale, Manchester) and rural (Melton Mowbray, Leicester) areas of the UK. Six DRT service variants are explored using mixed logit models; from those, a rural hopper service linking a number of rural settlements to a market town fits our research. Regarding that system, authors find the in-vehicle time of passengers is longer than normal, as the alternative to the DRT service is private motorized vehicles. Longer times are mainly caused by the dispersion of the served population, and the need for door-to-door as opposed to stop-based services, necessary due to the predominantly elderly and/or mobility-impaired users.

Coutinho et al. [15] assess replacing a fixed public bus line with a DRT system to service the rural surroundings of Amsterdam, the Netherlands. Their analysis focused on indicators such as distances, ridership, costs, greenhouse gas (GHG) emissions and the population's perception of DRT. Their results expect a drop in ridership which is compensated by mileage and operating time-frame reductions. There is better overall efficiency with DRT compared to the fixed service. The number of traveled kilometers, operational costs and GHG emissions per passenger were smaller.

C.-G. Roh and J. Kim [27] analyze and propose an optimization for six small bus routes in the rural city of Yangsan-si, South Korea. Geographic Information Systems (GIS) were used to compare and review the planned routes and operation status of each route, while improved DRT operation methods were studied based on these operations patterns. A more suitable DRT small bus operation model for each route was proposed as a conclusion.

3. DRT Systems' Adoption Rate

The adoption rate of newly deployed DRT systems is tightly related to their success. Some authors center their assessments on this topic. Wang et al. [28] discuss the DRT adoption rate in the rural area of Lincolnshire, England. The authors argue that car ownership, the aging population, and cuts in public spending threaten the traditional public bus services that operate in rural settlements. DRT, however, faces a series of challenges for its successful implementation. Through the analysis of various factors, it is determined that people with disabilities, those traveling for work, and those who live in less densely populated areas are more likely to travel by DRT. In addition, a gender-based analysis reveals females have a higher propensity to use DR services compared to males below retirement age. However, the trend vanishes upon reaching retirement age. This, for the authors, indicates an emerging market potential from the retired male market segment, and thus service providers should design their systems considering it.

Anburuvel et al. [29] run a survey to explore the willingness to accept a DRT service for the spatially scattered population of a rural region of Sri Lanka. The survey pointed towards economic attributes (income and vehicle ownership), sociocultural attributes (age, gender, and education), and mobility needs (travel frequency and access distance/cost) as the primary factors which decided the choice of a transport mode, thus begin more relevant in the decision of the deployment of a new service.

Schasché et al. [30] elaborate a review on the conflicting expectations and weak user acceptance of rural-DRT systems. Their paper creates an overview of the development in the research field, focusing particularly on user-oriented research, detects conflicting performance expectations towards DRT services that complicate their success, and identifies discrepancies between perception and empirical design studies. The findings suggest a need for more focus on rural areas when attempting to reduce the use of private combustion engine vehicles in favor of public transport and successfully establish DRT services as well as further research into specific user groups. The main take-away points are the following: In rural areas, personal factors such as age, gender, and private car access are found to be of stronger influence on user acceptance than in urban areas. Service-related factors like time reliability and booking methods have a higher impact on rural transport mode decisions than in urban settings. Finally, knowledge of DRT service and information provision also appears more influential for users in sparsely populated regions.

4. Reviews on Smart and Sustainable Mobility for Rural Areas

Some of the most useful theoretical contributions come from those works that group relevant publications, much like the present paper. The perspectives and criteria for the grouping are what differentiated one review on a concrete topic from another. Agriesti et al. [31] aim to build the case for a renewed research effort about smart mobility in low-density areas. The authors perform a wide surveying effort across Estonian municipalities, focusing on the outputs from rural and small suburban centers. The results report the main mobility challenges across the region and what hindering factors are preventing envisioned solutions. Tracking social behavior, changing travel patterns, and social inclusion stand out among these challenges. Technology implementation is also identified as a key priority, particularly regarding traffic management and planning practices.

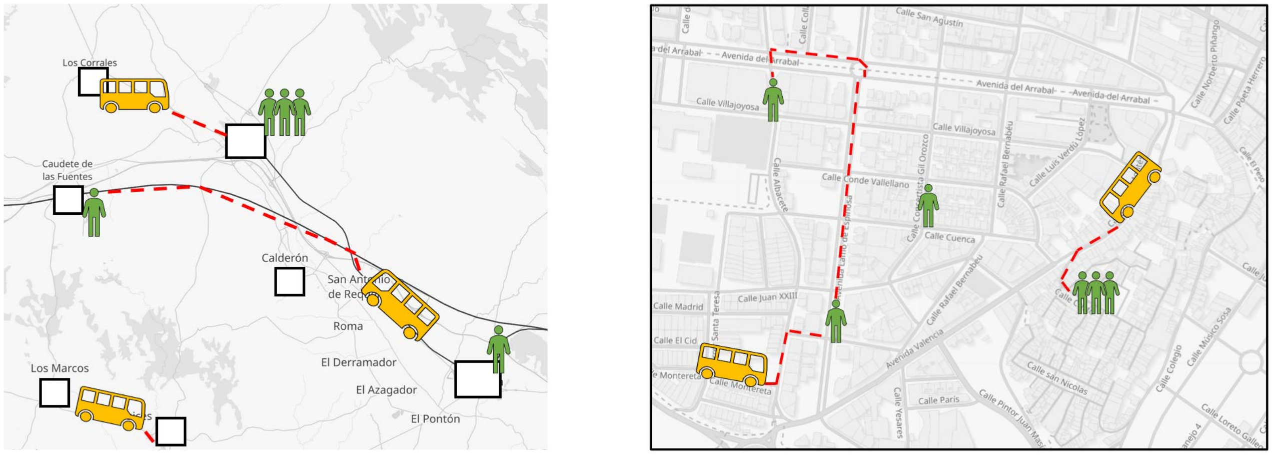
Poltimäe et al. [32] present a review of papers dealing with inclusive and sustainable mobility systems for rural areas. After analyzing many proposals, the authors group them into four categories: semi-flexible DRT, flexible door-to-door DRT, car-sharing, and ride-sharing. The main conclusion of their study is that single mobility solutions are rarely applicable to all rural travelers. Therefore, the future lies in multimodal mobility, considering that strong spatial and temporal synergies exist when combining different solutions. Success factors for sustainable rural transportation are identified, among which accessible and easily understandable information on routing, booking, and ticketing systems, as well as cooperation, shared values and trust between various parties, stand out. Finally, the importance of integrating the needs of various user groups for implementing environmentally, socially, and economically sustainable mobility solutions in rural areas is emphasized.

5. Other Analytical Contributions

Given the strong relationship between transportation systems and the area they service, some authors focus their surveys and proposals on specific topics which are relevant in their case. Abdullah et al. [33] assess the service quality of two DRT bus services operating in Lahore, Pakistan, through a questionnaire. The surveyed data reflected service attributes and bus ambiance as significant predictors of overall customer satisfaction.

F. Heinitz [34] approaches the improvement of rural mobility through incentives for private vehicle drivers to share their vehicle with other passengers for a concrete journey. The author builds a framework that defines steps to take when considering the introduction of DRT elements to a rural mobility scenario. His case study, set in the Schmalkalden-Meiningen area, Germany, takes into account German legislation. The author's conclusions show he understands as a mistake the proposal of a whole DRT solution from scratch for a certain rural area. Instead, he bets on modal integration and the development of high-adoption ridesharing among citizens, as private vehicles are the best approach to the mobility patterns of rural inhabitants.

F. Cavallaro and S. Nocera [35] study the novel concept of integrating passenger and freight transportation in flexible-route vehicles for rural areas. The developed case study is centered in the municipality



(a) Interurban stop-based operation

(b) Within settlement door-to-door operation

Fig. 2. Graphic representations of demand-responsive transportation systems operating with different configurations. Passenger demand is depicted by green human icons, whereas vehicles are portrayed by yellow buses. Vehicle routes are indicated with red dashed lines. Picture (a) reproduces an interurban operation, where settlements, indicated with white boxes, act as stops to travel from/to. Picture (b) depicts a door-to-door operation within a rural settlements, in which passengers can ask for a ride from any location within the town.

of Misano Adriatico, Italy. The performance of the service is evaluated through a selection of financial, operational, environmental, and social key performance indicators. The results of the analysis revealed a reduction in kilometers traveled, fuel consumption, and air pollutants, together with an increase in the area covered by the service, an increase in potential daily deliveries (for freight transport), and an increase in the occupancy rates of vehicles (for passengers).

C. Proposals and Experimental Work

This subsection groups the state-of-the-art literature which explicitly describes either a complete DRT system or some crucial part of it, including proposals that seek to optimize the system's operation or that simply test a particular approach for modeling, scheduling, or simulation.

Two main criteria have been used to divide the publications according to the system's proposed features. On the one hand, systems following many-to-many locations' operational patterns are separated from those using a many-to-one scheme. On the other hand, within each operational pattern, systems are split into those with fully-flexible routing and scheduling and those with semi-flexible ones.

Fig. 2 illustrates different DRT configurations that were found among the proposals analyzed in this section.

1. Many-To-Many Operational Pattern

Fully-flexible scheduling

Among the analyzed works that implement and validate concrete proposals, a few aim to enhance a commonly used technique or define approaches that deviate from the norm. Van Engelen et al. [18] propose an enhancement to insertion heuristics by including demand anticipation. Their algorithm is tested over the Tata Steel IJmuiden area in the Netherlands. The demand forecast is considered when a new request arrives in the system and is used to filter the number of fleet vehicles that can serve it. Generally, a vehicle will have enough free seats to serve passengers (demand) at the next stop on its route. Demand forecasting is applied to decide the probability that the next stop will have more demand than what the system currently considers. A vehicle may be rerouted to a stop with an expected demand greater than its current seat availability if the operator has "low confidence" in the demand forecast; this implies taking a risk. Conversely, when there is high confidence in the prediction, vehicles with a higher number of available seats than the current demand are rerouted, thus making room for the estimated demand as well. The authors compare their method to traditional insertion heuristics. The results show that by combining their proposal with empty vehicle rerouting 98% of the baseline rejected requests are eliminated, and travel and waiting times are reduced by up to 10 and 46%, respectively.

K. Viergutz and C. Schmidt [16] propose a case study on the rural town of Colditz, Germany, comparing conventional public transportation against DR services. The conventional transportation consisted of a bus line, whereas for the DRT two proposals were tested. Both DR proposals were on-demand, many-to-many, and fully-flexible. However, one of them operated stop-based with 5 automobiles and the other door-to-door with 10 vans. Their system declared constraints on the number of fleet vehicles, vehicle capacity, the maximum waiting and passenger travel time, and walking distance to the nearest stop. The scheduling of services was performed by a heuristic algorithm that allocates the nearest idle vehicle to each new request. The authors used surveyed and statistical data to reproduce realistic demand for the experimentation. Then, multi-agent simulations were run for each fleet configuration. Their findings revealed that, for the stop-based scenario, the number of passengers increases compared to conventional PT, but also does the fleet necessary to keep a good level of service (four vehicles vs one). Moreover, dynamic, real-time vehicle assignment requires hard-and software, which involves additional expenses to already financially limited rural PT providers. An excess of dynamism in PT (absence of lines and timetables), according to the authors, may be a strain on customers, leaving them at the mercy of their technical capabilities for managing booking applications. The work concludes that ultra-flexible DRT services are not the panacea for the rural PT sector, especially not in the case of a free-floating, DR, door-to-door service. Economically speaking, the authors remark on the importance of autonomous vehicles for a more efficient DRT.

Dytckov et al. [8] explore by means of simulation the benefits of replacing existing bus lines in the rural area of Lolland, Denmark, with a DRT system that better fits the low mobility demand. Authors build their own microsimulator joining together many open-source tools: a multimodal travel planner for scheduling (OpenTripPlanner), a library for solving vehicle routing problems (jsprit), OSRM to prepare data for the routing solver, and finally a custom event-driven simulator. Their proposal consists of an on-demand, fully-flexible, many-to-many, stop-based DRT system served by eight-seat minibusses. During the experimentation, constraints on request lead time, time window, trip time, driving time, and vehicle capacity are defined and modified. In addition, authors consider penalties for rejected requests and for the dispatching of new vehicles. The main assumption of their study is that transportation demand does not change when changing from buses to a DRT system. The simulation results show the potential to reduce costs and CO_2 emissions.

Morrison and T. Hanson [36] explore the concept of volunteer driver programs (VDPs) to replicate a door-to-door DRT service in rural areas. A rule-based system was developed to describe the operation of a VDP. The system was calibrated and validated with one year of New Brunswick (Canada) Volunteer Driving database data. Then, the multi-agent simulator Netlogo was used to implement and study a simple agent-based VDP. The system operation was simulated and stressed through many scenarios that posed challenges. Finally, VDPs were understood as a viable solution, although the authors remark on the need for additional research regarding actor (users, drivers, dispatchers) interactions.

Matsuhita et al. [37] propose two methods for promoting tourism use of a demand transportation system operated in the rural town of Aizumisato, Fukushima Prefecture, Japan. These proposed methods are a hybrid operation of both conventional on-demand transportation and scheduled transportation which is compatible with Google Map route search and the posting of times and routes using virtual stops. The effect of the proposals is studied utilizing the SUMO microscopic traffic simulator. The results show that the proposed system can operate on time without any problems, although the waiting time for passengers increases compared to the current method. The average maximum number of passengers that can be picked up and dropped off within 30 minutes is 12.3, which means that the system can operate with an increase of about four passengers compared to its current maximum capacity during peak hours.

Semi-flexible scheduling

Bruzzone et al. [38] explore the implementation of a DRT solution for the rural town of Velenje, Slovenia, given the poor performance of its current transit system. The researchers surveyed a focus group to establish the faults of the current transportation and the citizen's attitude towards on-demand mobility and cycling. The authors had the parallel objective of moving demand away from private motorized transports. Their final proposal combines two new DR bus lines and an electric bike-sharing system (e-BBS). The main DR line offers a semi-flexible, many-to-many service with a scheduled route and several on-demand stops; meanwhile, the secondary line operates in a fully-flexible manner, feeding the main line with a many-to-one[2] service. The e-BBS has two roles; feeding both DRT bus lines and offering accessible transportation for short displacements within the town's neighborhoods. Cost analysis reveals the proposal would achieve better service quality with the same financing the current public transportation is getting, reaching a higher percentage of the population.

Li et al. [39] propose a method for transit scheduling of DRT systems based on optimizing urban and rural transportation stops. Their method clusters passenger reservation demand through a DK-means clustering algorithm, identifying later fixed and alternative stops for the transportation system. Then, a genetic simulated annealing algorithm is proposed to build the bus schedule, obtaining a flexible-route DRT service that promotes urban-rural connections. Their proposal is validated in the northern area of Yongcheng City, Henan Province, China. Comparing their final model against the existing regional flexible buses, results show the optimized bus scheduling reduced the operating cost by 9.5% compared with that of regional flexible buses while reducing the running time by 9%. In addition, the authors compare their final proposal to that obtained merely after the DK-means clustering of stops and observed a 4.5% reduction in operational costs and 5% reduction in run times, thus proving the genetic simulated annealing step crucial to improve the service further.

2. Many-To-One Operational Pattern

Fully-flexible scheduling

Vehicle dispatching (from the current stop to the following one) in DR services is generally computed as a function of time, ensuring early service to boarded customers and waiting at stops only when there is enough slack time. Marković et al. [25] propose a threshold policy to dispatch vehicles according to the number of onboard passengers. For the experimentation, a flexible, one-to-many, door-based DR service is implemented, transporting the customers from a terminal to their homes. The authors adjust their proposal through numerical simulations set in a rural context, with demands ranging from 21 to 30 passengers per hour. They aim to find the threshold that reduces hourly costs as well as the adequate fleet size. The results indicate that the optimal threshold is a function of time-varying demand and thus must be adjusted for different times of the day. In contrast, the fleet size must be adjusted accordingly.

J. Bischoff and M. Maciejewski [20] propose an optimization for the operation of a DR fleet based on balancing vehicles according to the expected trip demand. Their method ensures that the spatial availability of vehicles follows the spatial distribution of demand in the (near) future. To test their proposal, the authors implement a feeder service that connects inhabitants of rural areas to other high-capacity means of transportation. The system operation is simulated with MATSim. The passenger-vehicle allocation is done through insertion heuristics where, given a request, each feasible insertion point is assessed and the best one is chosen. The balancing of the fleet is done as follows: First, rebalanceable (with enough slack time) and soon-to-be-idle vehicles are grouped. Then, the amount of demand per zone is estimated according to historical data. With that, the surplus (extra) vehicles in each zone are computed, and vehicles are dispatched from routes with a positive surplus to those with a negative one. Such dispatching aims to incur the shortest possible movement of the empty dispatched vehicle. The results show that customer waiting times can be cut up to 30% with no increase in VTK, meaning the rebalancing improves service quality at barely any monetary cost.

Schlüter et al. [17] assess the impact an autonomous DRT system would have in the specific case of linking an urban and a rural area. Specifically, their work is centered in the city of Bremerhaven, Germany, and its surrounding rural settlements of Lengen, Schiffdorf and Loxstedt. This constitutes a fairly wide area, leading the authors to two different assessments, centering one of them in the rural area. For that, an on-demand, door-to-door, many-to-one, fully-flexible service is established. As for the implementation, authors use the multi-agent simulator MATSim [40] with DRT modules. The road network is created with Open Source Routing Machine (OSRM), reproducing the real one. The system optimizes the operation through insertion heuristics, and the demand is generated following population statistics and surveys. The experimentation studies the replacement of the MIT (motorized individual transport). Results show that at least 1800 vehicles with a capacity of 6 passengers are necessary to provide a service rate of above 95%. Passenger waiting time values are below 13 minutes in this manner and decrease with an increasing number of DRT vehicles. The average travel time of the agents increases by around 66% when switching from a car-based scenario to pure DRT. Their results distill the following assessments: the number of vehicles can be reduced by more than 90%. By that, several negative side

[2] Bruzzone et al. use the term few-to-one, which would be a variation of a many-to-one operation with a relatively small number of origin stops.

effects such as congestion, noise, fragmentation, or land sealing can be mitigated, allowing new perspectives for urban planning and regional management. The replacement of human drivers with an autonomous driving system leads to a significant reduction in operational costs. However, the authors state that without the use of fully automated driving systems, DRT cannot compete economically. Finally, the limitations of this work come from the available data, which does not provide sufficient depth, the exclusion of public transportation from the simulated baseline framework, and the replacement of the entire MIT of a region, which is a radical theoretical approach. Authors remark that the adoption rate of new mobility, such as DRT systems, and the acceptance of fully automated vehicles determine the realistic percentage of MIT that can be replaced.

Calabrò et al. [10] explore the benefits of DRT feeder services with respect to a fixed-route (FR) service. Even though their experimentation takes place in a virtual road network, feeder services are one of the go-to DR modes in rural settlements, and thus we consider them relevant for the present review. The authors model a stop-based, many-to-one, fully-flexible, on-demand service. Their implementation employs basic insertion heuristics and a demand generation based on Poisson distributions. The system operated on a node-joint network. The simulations reveal that DRT is preferred in peripheral areas where the space between stops is high and during off-peak demand periods. In contrast, FR service performs better during peak demands. The recommendation for a transport operator is, therefore, to switch services according to the demand.

Semi-flexible scheduling

Lakatos et al. [9] explore the substitution of a regular bus line operating between 11 "dead-end" villages in rural Hungary. They describe a seven-step analysis method for the optimization of any transportation system. Such a method attaches particular importance to the characteristics of the current transport service (the one that would potentially be replaced). Their study is conducted through mathematical modeling fed by surveyed data. The study proposes three different DRT solutions. All proposals are on-demand and stop-based but vary in operational pattern and flexibility. Their first system (1) completely replaces all bus connections with a DRT service, modeling a many-to-many, semi-flexible operation. The second one (2) aims to replace only the detours that the bus has to do from the main line with a DRT service, keeping the regularly scheduled bus service just along the main line, describing a many-to-one, semi-flexible operation. Finally, the last proposal (3) introduces DRT just as an extra service connecting settlements with the present main route, therefore establishing a feeder for the main bus service. In this case, the operation would be many-to-one and fully-flexible. After analyzing all three proposals, the main bus line is kept for four of the settlements and the connection among all of them, whereas the other seven villages implement a DRT service, with one minibus each, connecting the stops within them to the main line. This configuration feeds the main line and avoids bus detours. The new configuration's cost does not exceed that of the traditional transportation system but increases the level of service with better frequencies and more connections. The authors emphasize the importance of developing policies with the public services for the viability of the rural-DRT system as well as the limitations of their method, which mainly considers ridership as an influencing factor.

IV. Discussion

The cited works have been summarized in a series of tables. Table I gathers the works from Section III.B whereas Table II collects those described in Section III.C.

A. Summary of Results

Observing Table I, certain topics stand out as the most investigated. DRT systems' failure as a general public transportation service has been widely studied. Such a topic is closely related to the adoption rate these systems have once deployed; the number of users that switch from their current transportation alternatives to the new DRT system. In addition, many authors aim to replace or improve the current PT of a rural area with a DRT solution. This is also the case for most of the assessed system proposals. Regarding the observed challenges for a viable and successful DRT system deployment, these can be grouped into economic challenges: unrealistic or excessively flexible operation, lack of partnerships, and poor adoption rate; and social challenges: scattered population, disparity among technological skills, low income, different social behaviors and travel patterns, and high ownership of MIT. Both analytical (Table I) and experimental works (Table II) acknowledge the potential of DRT to improve service quality and thus passenger satisfaction, and reduce vehicle mileage and operating hours, thus reducing the system's environmental impact too. Besides that, a series of factors increment the chances of a successful deployment of DRT: semi-flexible operations, user-focused design, user-group research, partnerships with public and private institutions, and the integration of different modes of transportation.

Regarding experimental works, Table II shows the most popular trend in terms of DRT systems' configuration: a many-to-many operational pattern with a fully-flexible routing, servicing a series of stops with an on-demand shared mid-capacity vehicle. The proposals mainly aim to replace or improve the operation of the current means of PT in a concrete rural area. In some cases, a new system is proposed from scratch to serve a specific unfulfilled displacement need. Among the observed used cases, most of them serve a series of locations freely, whereas a minority propose feeder systems that bring passengers to a higher-capacity, less flexible transportation network. Finally, it is usual that authors aim to optimize, at least, the passenger's perspective. Most of them also include an operator perspective, which is closely related to the economic viability of the service. Finally, a minority explicitly comments on the environmental improvements their system brings.

B. Open Issues

Following, the open issues and key insights distilled from our classification are discussed, providing a basis for reflection on the challenges and indicating possible solutions and recommendation.

1. Replacement and Optimization of Existing PT With DRT

The reviewed literature shows the difference among authors' insights regarding the performance of their proposed systems. For a fair assessment of a DRT proposal's performance, we must consider the context in which the system is proposed and thus its intended goals. The metrics that the authors will give importance to in their research depend on those goals. For instance, when it comes to public transportation optimization, usual metrics are passenger waiting and traveling time, vehicle traveled kilometers (VTK), and greenhouse gas (GHG) emissions. If a DRT system is proposed to replace or complement the current public transportation system, the research will focus on reducing passenger waiting and traveling time, VTK and GHG emissions. In contrast, a DRT service may be planned to introduce public transportation in an area where there are no mobility alternatives besides motorized individual transports (MIT). In such a case, the research will focus on the level of adoption rate of the new service and the reduction of MIT in favor of public transportation. Most of the cited works propose a partial or a complete replacement of the traditional means of transportation already implemented in a chosen rural area in favor of a new DRT solution. Those aiming for total replacement usually keep elements of the old transportation system (such as stops) in the DRT service. This approach eases the

TABLE I. Cited Survey and Analysis Works Classified by Main Topic, Data-Gathering Method, and Identified Challenges and Potentials. Acronyms: DRT (Demand-Responsive Transportation), MIT (Motorized Individual Transport), PT (Public Transportation), VTK (Vehicle Traveled Kilometers)

	Topic	Method	Challenges	Potentials
[5]	Success and failure of DRT systems	Analysis of failure factors	Unrealistic design, excessive flexibility, lack of partnership, high service costs	Simpler operations (in pattern and flexibility)
[4]		Review of DRT database		
[2]	Replacement/ optimization of public transportation with DRT	Citizen survey	Financial viability, institutional barriers	Mileage reduction, operating time-frame reduction, improved passenger load
[15]		Historical overview of DRT systems	Population's perspective, drop in ridership	
[27]		Modeling	Populations' aging and decline	
[28]	Adoption rate	Factor analysis	Ageing population, cuts in public expense	Market for commuters and retired population User-focused deployment of services Specific user group research
[29]		Citizen survey	Scattered population, low income, high vehicle ownership	
[30]		Literature review	Disparity among perception and empirical design	
[31]	Smart, sustainable mobility for rural areas	Citizen survey	Social behav. tracking, changing travel patterns, technology implementation	Multimodal mobility Cooperation among parties User group integration
[32]		Literature review	Mobility solutions tied to specific travelers	
[33]	Service quality	Questionnaire	High costs, institutional barriers	Customer satisfaction given by vehicle ambiance
[34]	Incentivized shared mobility	Modeling	Excess of MIT in rural areas, uneven travel patterns	Modal integration, citizen cooperation
[35]	Passenger-freight transportation	Modeling	Limited resources to guarantee access to main territorial hubs, underutilized PT	Higher area of service, higher occupancy, reduction in VTK

TABLE II. Cited Experimental Works Classified by Operational (Op.) Pattern, Route Flexibility, Stop Configuration, Booking Necessity, Fleet Size and Capacity, and Optimization Perspective (Persp.). Acronyms: E-BBS (Electric Bike-Sharing System)

	Op. pattern	Flexibility	Stops	Booking	<# vehicles>x<# seats>s	Optimization persp.
[18]	many-to-many	fully-flexible	stop-based	on-demand	100x5s	passenger
[16] (1)			stop-based	on-demand	5vx4s	operator passenger
[16] (2)			door-to-door	on-demand	10x6-14s	
[8]			stop-based	on-demand	29x8s, 19x8s	operator passenger environment
[36]			door-to-door	on-demand	4s (private cars)	passenger
[37]			door-to-door	on-demand	1x9s + 2x4s	passenger
[38] (1)		semi-flexible	stop-based	not needed	1 bus + e-BBS	passenger
[39]			stop-based	not needed	1x20s	operator passenger
[9] (1)			stop-based	on-demand	11x8s	operator passenger
[38] (2)	many-to-one	fully-flexible	stop-based	on-demand	1 bus + e-BBS	passenger
[25]			door-to-door	not needed	6x10s	operator
[20]			door-to-door	on-demand	100x4s	operator passenger
[17]			door-to-door	on-demand	1800x6s	operator passenger
[10]			stop-based	on-demand	3x20s, 5x8s, 10x4s, 20x2s	passenger
[9] (3)			stop-based	on-demand	1x50s + 11x8s	operator passenger
[9] (2)		semi-flexible	stop-based	on-demand	1x50s + 7x8s	operator passenger

comparison between new and previous transportation systems. However, it also facilitates results with lower VTK and, therefore, GHG emissions, as generally with DRT some of the stops along a vehicle line are optional. If the DRT is implemented as a door-to-door service, VTK and GHG may increase with respect to the existing means of PT, and thus an improvement in service quality through time reduction and the servicing of a wider area gain more relevance.

The substitution and improvement of preserved elements of the current PT of an area should also be assessed when aiming to improve its operation. As in [39], demand distribution and stop location can be studied and modified to fit the new proposal's characteristics better.

2. Disconnection Between Analysis and Proposals

From a general perspective, comparing the potentials that DRT offers (Table I) with the most popular system configurations (Table II), there appears to be a disconnection between theoretical and practical works. Even though surveys conclude on the benefits of simpler, less flexible operations and the inclusion of multimodality, the proposals present mostly fully-flexible services, and only some of them [38] consider a different transportation mode (electric bike-sharing) to complement DRT. Some authors [4], [5] agree that an excess of dynamism in demand-responsive operation can be too economically costly for the system's long-term sustainability, especially when the level of demand does not justify such a level of dynamism. The general conclusion of analytical works seems to favor semi-flexible systems, with elements from scheduled transportation (non-flexible) combined with on-demand, dynamic operation.

As a relatively new field, DRT lacks standardized systems, leading to a plethora of proposals, each with its own unique "name". Despite the abundance of ideas, a closer examination reveals that most systems are strikingly similar, varying only in minor details. Furthermore, there are few works that delve into the attributes of these models. Although it is expected to explore various algorithms and techniques in a field with many open issues like DRT, authors should focus on the specific contributions their algorithms and system models bring to passengers, operators, and drivers. It is crucial to adjust configurable components such as stops, assignments, and vehicle capacity to suit the specific real-world use case of the system.

Authors in [32], [34] comment on the importance of the integration of different modes of transportation to truly match the rural area inhabitants' mobility requirements. In addition, partnerships between the transportation provider and other entities have been identified as a factor contributing to DRT success. One of the many ways multimodal transportation and partnerships can be promoted is through mobility hubs [41], physical locations where different modes of transportation are integrated. Mobility hubs provide travelers with options for transfers between various transport systems in order to facilitate the exchange from one mode of travel to another. Moreover, they can also include amenities like shops and restaurants, making them attractive places to visit while traveling. Given the high percentage of failed DRT systems, we consider the implementation of mobility hubs must be studied together with the topic at hand.

3. User-Centered Design and Adoption Rate

It seems evident that a transportation system has to adapt to the area it serves. Elements such as routes, stops and vehicles take into account the geography and spatial-temporal demand of the area. However, when it comes to classic transport systems, the way they operate remains the same regardless of where they are implemented. In the case of DRT, generalist solutions have no place, even less so in rural areas. Their necessary flexibility, combined with the low and distributed demand, forces an operation tailored to the reality of the system's potential users.

How well a system is adapted to its potential users determines the number of final users it will have. This is even more evident for systems that compete with other alternatives, such as transportation systems. Therefore, user-centered design is closely related to the final DRT system's adoption rate. The adoption rate of DRT is one of the key issues leading to its failure. The number of passengers that may switch from existing PT or MIT to DRT depends on the service quality and the ease of interaction with the service. The latter concept refers to the booking of services, which is generally done through a call center, web, or smartphone application. Because of all the aforementioned, when simulating a DRT operation, the demand intensity must be adapted accordingly and not simply copied from the existing PT or MIT displacements. In addition, by including findings on human behavior in such simulations, further research could simulate the estimated depth and speed of user transition from their preferred transportation method to the new DRT solution.

Works such as [28]–[30], [32] conclude on the importance of adapting the design, operation, and deployment of DRT solutions to specific user groups. The displacement requirements of potential users should be at the center of the development of a mobility system. In rural areas, where demand is low, and the gap between users is widening, it is especially crucial to consider their characteristics, such as social and travel patterns and technological skills. However, it would be unrealistic to propose a system that adapts to each and every one of its users. Because of that, user-group research is advisable to determine the best operation for the system. Moreover, we consider hybrid operations that adapt to different user groups in various periods of the day as a potential solution to increase a system's adoption rate.

4. Artificial Intelligence for Rural-DRT

Regarding rural-DRT research, we can establish a baseline of commonly discussed topics and commonly applied technologies for modeling and simulation. Regarding the latter, most proposals are modeled through mathematical or agent-based approaches. The demand for the system's validation is synthetically generated according to surveys and population, vehicle ownership, and other relevant statistics from the serviced area. The system counts with routing algorithms and insertion heuristics to assign passengers to vehicles and schedule the service. Finally, numerical or agent-based simulations are run according to the modeling, and conclusions about the proposal are drawn.

Recently, rural areas have attracted the interest of artificial intelligence researchers, in order to apply in them the type of techniques which are already being developed for smart cities [42], [43]. Still, there is a noticeable lack of innovation regarding rural-specific transportation. Certain aspects of transportation research, such as autonomous vehicles [44], enjoy a high level of popularity and therefore a high level of articles. For the case of DRT, most of the proposed systems do not implement new algorithms for allocating the demand or scheduling operations. On the contrary, the authors assess the viability of specific proposals. The few improvements for the classic algorithms that have been reviewed present general optimizations and do not consider the characteristics of the rural demand to further improve the system. Because of that, we wish to highlight those contributions which innovate regarding research topics.

The works in [34]–[36] present unexplored topics which tie their proposals to specific characteristics of the serviced area. These topics are incentive-driven shared mobility, integrated passenger-freight transportation, and volunteer driving programs, respectively. In addition, some authors innovate with the optimization techniques applied to their systems. In [18], demand anticipation is used to improve the classic insertion heuristic. In [25], the authors propose a dispatch policy based on a threshold of passengers onboard a vehicle.

In [20], vehicles are rebalanced based on expected demand. Finally, the authors in [39] employ generally unused techniques for their proposal: DK-means to group stops and a genetic algorithm (global optimization) combined with simulated annealing (local optimization) to define the system's operation. These works, regardless of their relevance, bring freshness to the field of research and, as analyzed in this paper, follow the line necessary to apply real solutions that work in concrete rural areas.

The DRT paradigm facilitates resource savings and transport adaptability. Hand in hand with artificial intelligence (AI), the potential for improving rural mobility increases considerably. Machine learning and pattern recognition techniques can be used for demand prediction and generation, both historically and in real time. This, in turn, may optimize vehicle deployment and passenger balancing. AI can also identify and group potential customers of a future DRT service according to their social behavior and travel patterns. Regarding the adoption rate of DRT, AI can be implemented to analyze data about the needs of rural populations and identify ways to increase the demand for transportation services, such as gamification: creating incentives for people to use public transportation or offering discounts to those who carpool. Additionally, as mentioned throughout the paper, heuristic optimization can improve transportation conditions, identify the best routes, and create more efficient routes that reduce the amount of time and money spent on displacement. There are myriad approaches that can be leveraged to the topic at hand, from agent negotiation to evolutionary computation, and most are worth exploring to build original solutions for a research field in need of innovation.

V. Conclusions

This survey has reviewed relevant works that assess the viability and potential of improvement that the DRT paradigm can bring to rural mobility. Such a task included the description of transportation problems, the characterization of DRT, and the enumeration of the techniques that computer science brings to implement and experiment with transportation systems. Both analytical and experimental works have been described and classified. Finally, the open issues of the matter, gathered from the reviewing process, have been discussed.

The main takeaway points of the present work are the following. Practical research needs to be more in touch with its theoretical counterpart. Works that apply the knowledge of transportation research must favor the approaches which are economically viable. The problem of low adoption rate and implementation that does not adapt to the potential users of the rural area has to be considered in every step of the formulation of the transportation system. PT providers must understand those issues and adapt their expected ridership amount accordingly. It is smart to begin with a somewhat less flexible operation and increase the flexibility if factors such as demand justify it. Finally, one should always keep in mind the potential of multimodal transportation; study the application area to try and create partnerships with other actors that facilitate the transition to the new transportation method.

From the point of viewof computer science research, there is a need for rural-specific works that use the deployment area's features to find innovative and creative optimization solutions. There are a series of unexplored algorithms that could bring new perspectives to synthetic data generation, mobility modeling, and simulation.

The present research inspires two logical follow-up works. On the one hand, the results of this work could be applied to the definition of a framework describing the series of steps that both PT providers and researchers in the area should follow when considering the design and implementation of a DRT system, giving the necessary importance to user-centered design, multimodality, and innovation in modeling and optimization. On the other, we would like to take advantage of the latest advances in AI to study the best way to implement and improve rural-DRT.

Regarding the latter, we have plans to develop a general framework for transportation fleet optimization. Employing agent-based modeling to reproduce public transportation and other types of fleets, and integrating different algorithms to optimize aspects such as task allocation and vehicle coordination from both a centralized and a distributed perspective. A few examples of algorithms we have been researching would include insertion heuristics, distributed negotiation and task allocation through auctions, or distributed planning of the fleet's operations [45]. With the aforementioned ideas, a first approach on demand-responsive systems can be found in [46].

Machine learning techniques are also a powerful tool to innovate in the improvement of the operational area and further optimize transportation operations. For instance, in [47] demand-prediction models are developed to test and optimize a public bus service. Finally, we would use massive multi-agent simulation techniques, such as those illustrated in [48], to validate the different systems and identify potential partnerships with other means of transport or actors in the rural area.

Acknowledgment

This work is partially supported by grant PID2021-123673OB-C31 funded by MCIN/AEI/ 10.13039/501100011033 and by "ERDF A way of making Europe". Pasqual Martí is supported by grant ACIF/2021/259 funded by the "Conselleria de Innovación, Universidades, Ciencia y Sociedad Digital de la Generalitat Valenciana". Jaume Jordán is supported by grant IJC2020-045683-I funded by MCIN/AEI/ 10.13039/501100011033 and by "European Union NextGenerationEU/PRTR".

References

[1] R. Choudhary, V. Vasudevan, "Study of vehicle ownership for urban and rural households in india," *Journal of Transport Geography*, vol. 58, pp. 52–58, 2017, doi: https://doi.org/10.1016/j.jtrangeo.2016.11.006.

[2] T. J. Ryley, P. A. Stanley, M. P. Enoch, A. M. Zanni, M. A. Quddus, "Investigating the contribution of demand responsive transport to a sustainable local public transport system," *Research in Transportation Economics*, vol. 48, pp. 364–372, 2014.

[3] S. C. Ho,W. Szeto, Y.-H. Kuo, J. M. Leung, M. Petering, T.W. Tou, "Asurvey of dial-a-ride problems: Literature review and recent developments," *Transportation Research Part B: Methodological*, vol. 111, pp. 395–421, 2018.

[4] G. Currie, N. Fournier, "Why most drt/micro-transits fail–what the survivors tell us about progress," *Research in Transportation Economics*, vol. 83, p. 100895, 2020.

[5] M. Enoch, S. Potter, G. Parkhurst, M. Smith, "Why do demand responsive transport systems fail?," in *Transportation Research Board 85th Annual Meeting*, Washington DC, USA, 22-26 Jan 2006.

[6] L. Butler, T. Yigitcanlar, A. Paz, "Smart urban mobility innovations: A comprehensive review and evaluation," *IEEE Access*, vol. 8, pp. 196034–196049, 2020, doi: 10.1109/ACCESS.2020.3034596.

[7] P. Vansteenwegen, L. Melis, D. Aktaş, B. D. G. Montenegro, F. S. Vieira, K. Sörensen, "A survey on demand-responsive public bus systems," *Transportation Research Part C: Emerging Technologies*, vol. 137, p. 103573, 2022.

[8] S. Dytckov, J. A. Persson, F. Lorig, P. Davidsson, "Potential benefits of demand responsive transport in rural areas: A simulation study in lolland, denmark," *Sustainability*, vol. 14, no. 6, 2022.

[9] A. Lakatos, J. Tóth, P. Mándoki, "Demand responsive transport service of 'dead-end villages' in interurban traffic," *Sustainability*, vol. 12, no. 9, 2020.

[10] G. Calabrò, M. Le Pira, N. Giuffrida, G. Inturri, M. Ignaccolo, G. Correia, "Fixed-route vs demand- responsive transport feeder services: An exploratory study using an agent-based model," *J. of Advanced Transportation*, vol. 2022, 2022.

[11] H. B. Demir, E. Pekel Özmen, S. Esnaf, "Time- windowed vehicle

routing problem: Tabu search algorithm approach," *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 11, p. 179–189, Oct. 2022, doi: 10.14201/adcaij.27533.

[12] E. Osaba, F. Diaz, "Design and implementation of a combinatorial optimization multi-population meta-heuristic for solving vehicle routing problems," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, pp. 89–90, 12/2016 2016, doi: 10.9781/ijimai.2016.4213.

[13] M. Hyland, H. S. Mahmassani, "Operational benefits and challenges of shared-ride automated mobility-on- demand services," *Transportation Research Part A: Policy and Practice*, vol. 134, pp. 251–270, 2020.

[14] S. Vallée, A. Oulamara, W. R. Cherif-Khettaf, "Maximizing the number of served requests in an online shared transport system by solving a dynamic darp," in *Computational Logistics*, Cham, 2017, pp. 64– 78, Springer International Publishing.

[15] F. M. Coutinho, N. van Oort, Z. Christoforou, M. J. Alonso-González, O. Cats, S. Hoogendoorn, "Impacts of replacing a fixed public transport line by a demand responsive transport system: Case study of a rural area in amsterdam," *Research in Transportation Economics*, vol. 83, p. 100910, 2020.

[16] K. Viergutz, C. Schmidt, "Demand responsive - vs. conventional public transportation: A matsim study about the rural town of colditz, germany," *Procedia Computer Science*, vol. 151, pp. 69–76, 2019.

[17] J. Schlüter, A. Bossert, P. Rössy, M. Kersting, "Impact assessment of autonomous demand responsive transport as a link between urban and rural areas," *Research in Transportation Business Management*, vol. 39, p. 100613, 2021.

[18] M. van Engelen, O. Cats, H. Post, K. Aardal, "Enhancing flexible transport services with demand- anticipatory insertion heuristics," *Transportation Research Part E: Logistics and Transportation Review*, vol. 110, pp. 110–121, 2018.

[19] M. Balmer, M. Rieser, K. Meister, D. Charypar, N. Lefebvre, K. Nagel, "Matsim-t: Architecture and simulation times," in *Multi-agent systems for traffic and transportation engineering*, IGI Global, 2009, pp. 57–78.

[20] J. Bischoff, M. Maciejewski, "Proactive empty vehicle rebalancing for demand responsive transport services," *Procedia Computer Science*, vol. 170, pp. 739– 744, 2020.

[21] C. Bertelle, M. Nabaa, D. Olivier, P. Tranouez, "A decentralised approach for the transportation on demand problem," in *From System Complexity to Emergent Properties*, Springer, 2009, pp. 281–289.

[22] S. Tisue, U. Wilensky, "Netlogo: A simple environment for modeling complexity," in *Int. conference on complex systems*, vol. 21, 2004, pp. 16–21, Boston, MA.

[23] G. Inturri, N. Giuffrida, M. Ignaccolo, M. Le Pira, A. Pluchino, A. Rapisarda, *Testing Demand Responsive Shared Transport Services via Agent-Based Simulations*, pp. 313–320. Cham: Springer International Publishing, 2018.

[24] J. Palanca, A. Terrasa, C. Carrascosa, V. Julián, "Simfleet: a new transport fleet simulator based on mas," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, 2019, pp. 257–264, Springer.

[25] N. Marković, M. E. Kim, E. Kim, S. Milinković, "A threshold policy for dispatching vehicles in demand-responsive transit systems," *Promet - Trafficamp;Transportation*, vol. 31, pp. 387–395, Aug. 2019.

[26] S. Liyanage, H. Dia, "An agent-based simulation approach for evaluating the performance of on- demand bus services," *Sustainability*, vol. 12, no. 10, 2020.

[27] C.-G. Roh, J. Kim, "What are more efficient transportation services in a rural area? a case study in yangsan city, south korea," *International journal of environmental research and public health*, vol. 19, no. 18, p. 11263, 2022.

[28] C. Wang, M. Quddus, M. Enoch, T. Ryley, L. Davison, "Exploring the propensity to travel by demand responsive transport in the rural area of lincolnshire in england," *Case Studies on Transport Policy*, vol. 3, no. 2, pp. 129–136, 2015.

[29] A. Anburuvel, W. Perera, R. Randeniya, "A demand responsive public transport for a spatially scattered population in a developing country," *Case Studies on Transport Policy*, vol. 10, no. 1, pp. 187–197, 2022.

[30] S. E. Schasché, R. G. Sposato, N. Hampl, "The dilemma of demand-responsive transport services in rural areas: Conflicting expectations and weak user acceptance," *Transport Policy*, vol. 126, pp. 43–54, 2022.

[31] S. A. M. Agriesti, R.-M. Soe, M. A. Saif, "Framework for connecting the mobility challenges in low density areas to smart mobility solutions: the case study of estonian municipalities," *European Transport Research Review*, vol. 14, no. 1, p. 32, 2022.

[32] H. Poltimäe, M. Rehema, J. Raun, A. Poom, "In search of sustainable and inclusive mobility solutions for rural areas," *European transport research review*, vol. 14, no. 1, p. 13, 2022.

[33] M. Abdullah, N. Ali, S. A. H. Shah, M. A. Javid, T. Campisi, "Service quality assessment of app-based demand-responsive public transit services in lahore, pakistan," *Applied Sciences*, vol. 11, no. 4, p. 1911, 2021.

[34] F. Heinitz, "Sustainable development assessment of incentive-driven shared on-demand mobility systems in rural settings," *European Transport Research Review*, vol. 14, no. 1, p. 38, 2022.

[35] F. Cavallaro, S. Nocera, "Flexible-route integrated passenger–freight transport in rural areas," *Transportation Research Part A: Policy and Practice*, vol. 169, p. 103604, 2023.

[36] R. Morrison, T. Hanson, "Exploring agent-based modelling for car-based volunteer driver program planning," *Transportation research record*, vol. 2676, no. 11, pp. 520–532, 2022.

[37] S. Matsuhita, S. Yumita, T. Nagaosa, "A proposal and performance evaluation of utilization methods for tourism of a demand-responsive transport system at a rural town," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 2920–2925, IEEE.

[38] F. Bruzzone, M. Scorrano, S. Nocera, "The combination of e-bike-sharing and demand-responsive transport systems in rural areas: A case study of velenje," *Research in Transportation Business & Management*, vol. 40, p. 100570, 2021.

[39] P. Li, L. Jiang, S. Zhang, X. Jiang, "Demand response transit scheduling research based on urban and rural transportation station optimization," *Sustainability*, vol. 14, no. 20, p. 13328, 2022.

[40] A. Horni, K. Nagel, K. Axhausen Eds., *Multi-Agent Transport Simulation MATSim*. London: Ubiquity Press, Aug 2016.

[41] T. Rongen, T. Tillema, J. Arts, M. J. Alonso-González, J.-J. Witte, "An analysis of the mobility hub concept in the netherlands: Historical lessons for its implementation," *Journal of Transport Geography*, vol. 104, p. 103419, 2022.

[42] G. Mariammal, A. Suruliandi, S. P. Raja, E. Poongothai, "An empirical evaluation of machine learning techniques for crop prediction," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. In Press, pp. 1–9, 12/2022 9998, doi: 10.9781/ijimai.2022.12.004.

[43] V. K. Solanki, M. Venkaesan, S. Katiyar, "Conceptual model for smart cities: Irrigation and highway lamps using iot," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, pp. 28–33, 03/2017 2017, doi: 10.9781/ijimai.2017.435.

[44] M. Qader Kheder, M. Aree Ali, "Iot-based vision techniques in autonomous driving: A review," *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 11, p. 367–394, Jan. 2023.

[45] P. Martí, J. Jordán, V. Julian, "Best-response planning for urban fleet coordination," *Neural Computing and Applications*, pp. 1–20, 2023.

[46] P. Martí, J. Jordán, F. De la Prieta, H. Billhardt, V. Julian, "Demand-responsive shared transportation: A self- interested proposal," *Electronics*, vol. 11, no. 1, 2022, doi: 10.3390/electronics11010078.

[47] A. Ibáñez, J. Jordán, V. Julian, "Improving public transportation efficiency through accurate bus passenger demand," in *Highlights in Practical Applications of Agents, Multi-Agent Systems, and Cognitive Mimetics. The PAAMS Collection*, Cham, 2023, pp. 18–29, Springer Nature Switzerland.

[48] P. Martí, J. Llopis, V. Julian, P. Novais, J. Jordán, "Validating state-wide charging station network through agent-based simulation," in *Highlights in Practical Applications of Agents, Multi-Agent Systems, and Cognitive Mimetics. The PAAMS Collection*, Cham, 2023, pp. 158–169, Springer Nature Switzerland.

Pasqual Martí

Pasqual Martí is a Ph.D. student and researcher at the Valencian Research Institute for Artificial Intelligence (VRAIN) of the Universitat Politècnica de València. He finished his degree studies in Computer Engineering in 2019 and continued his education with a master's degree in Artificial Intelligence, which finished in 2020. During the development of his master, he began his career in research through a scholarship, participating in 2 R&D projects related to Multi-Agent Systems and Urban Fleets. From such work, he has published several articles in indexed journals and international conferences.

Jaume Jordán

Jaume Jordán is a postdoctoral researcher at the Valencian Research Institute for Artificial Intelligence (VRAIN) of the Universitat Politècnica de València. He finished his Ph.D. in 2017 with a grade of Outstanding Cum Laude and international mention. He has participated in more than 10 R&D projects (1 as principal investigator) related to Artificial Intelligence, particularly in the lines that integrate Multi-Agent Systems, Planning, Game Theory, Genetic Algorithms, and Urban Fleets. He more than 35 works published in relevant international conferences and JCR journals, and he collaborated as an organizer, committee, editor, and reviewer of different conferences and journals.

Angélica González Arrieta

Received a Ph.D. in Computer Science from the University of Salamanca in 2000. She is currently a Lecturer at Salamanca's University Department of Computer Science and has attended several Masters courses. She is also a UNED professor and tutor (Universidad Española de Educación a Distancia, Spain's Open University). Previously, she carried out relevant administrative tasks, such as Academic Secretary of the Science Faculty (1996-2000) and Chief of Staff for the University of Salamanca (2000-2003). Since 1990, she has cooperated with the Home Ministry, and from 2008 with the Home and Justice Counsel of the local government (Junta de Castilla y León). She is a member of the research group BISITE (http://bisite.usal.es) and has led several research projects sponsored by both public and private institutions in Spain. She co-authors several works published in journals, workshops, meetings, and symposia.

Vicente Julian

Vicente Julian holds the position of Full Professor of Computer Science at the Universitat Politècnica de València (UPV) where he has taught since 1996. He is a member of the Valencian Research Institute for Artificial Intelligence (VRAIN), and the Coordinator of the Computer Science Ph.D. Program at the UPV. Four international projects, two international excellence networks, twenty-five Spanish projects, and four technology transfer projects have covered the research on Artificial Intelligence. He has more than 300 works published in journals with outstanding JCR positions or in relevant conference proceedings. Vicente Julian has supervised 10 Ph.D. Thesis.

An Investigation Into Different Text Representations to Train an Artificial Immune Network for Clustering Texts

Matheus A. Ferraria[1], Vinicius A. Ferraria[1], Leandro N. de Castro[2,3] *

[1] Graduate Program in Electrical Engineering and Computing, Mackenzie Presbyterian University (Brazil)
[2] U.A. Whitaker College of Engineering, Florida Gulf Coast University (USA)
[3] Graduate Program in Technology, University of Campinas (Brazil)

Received 12 March 2023 | Accepted 18 August 2023 | Published 28 August 2023

Abstract

Extracting knowledge from text data is a complex task that is usually performed by first structuring the texts and then applying machine learning algorithms, or by using specific deep architectures capable of dealing directly with the raw text data. The traditional approach to structure texts is called Bag of Words (BoW) and consists of transforming each word in a document into a dimension (variable) in the structured data. Another approach uses grammatical classes to categorize the words and, thus, limit the dimension of the structured data to the number of grammatical categories. Another form of structuring text data for analysis is by using a distributed representation of words, sentences, or documents with methods like Word2Vec, Doc2Vec, and SBERT. This paper investigates four classes of text structuring methods to prepare documents for being clustered by an artificial immune system called aiNet. The goal is to assess the influence of each structuring method in the quality of the clustering obtained by the system and how methods that belong to the same type of representation differ from each other, for example both LIWC and MRC are considered grammar-based models but each one of them uses completely different dictionaries to generate its representation. By using internal clustering measures, our results showed that vector space models, on average, presented the best results for the datasets chosen, followed closely by the state of the art SBERT model, and MRC had the overall worst performance. We could also observe a consistency in the number of clusters generated by each representation and for each dataset, having SBERT as the model that presented a number of clusters closer to the original number of classes in the data.

Keywords

Artificial Immune Network, Clonal Selection, Natural Computing, Text Clustering, Text Structuring.

DOI: 10.9781/ijimai.2023.08.006

I. Introduction

Text mining corresponds to a set of techniques used to extract patterns or identify trends in documents (textual datasets), bringing together Information Retrieval, Natural Language Processing (NLP), and Data Mining techniques [1]–[5]. Thus, text mining involves knowledge in linguistics, informatics, statistics and cognitive sciences, among other areas. Whilst data mining seeks patterns in numerical and categorical data, text mining is about looking for patterns in texts. This superficial similarity between the two areas hides their main difference [6]: data mining deals with *structured data* in standard databases, whilst texts are *semi-* or *unstructured data* covered with uncertainties, context and ambiguity, which make their analysis and interpretation even more difficult. Thus, text mining deals with semi- or unstructured data that is usually pre-processed before a learning algorithm can be applied.

The pre-processing step performs all the cleaning and structuring in the text to generate its structured representation suitable for the application of standard machine learning algorithms [7]. This text structuring step is usually the most sensitive and expensive one in computational terms, as it requires the processing of unstructured data [4], [8]. It can be divided into four main steps: 1) tokenization; 2) stop words removal; 3) lemmatization; and 4) representation of documents. After Steps (1) to (3), it is necessary to find a suitable representation for the documents (Step (4)), and there are different methods to do so.

The most common text representation approach is the so called *Bag of Words (BoW)* [9], [10], which models the documents only based on a specific weight calculated for each token (word) in the document, disregarding grammar, word order and context. These are called *vector space models*, in the sense that they transform texts into a set of vectors in a usually high-dimensional space, where each dimension corresponds to a word in the documents.

Alternative methods to represent documents include those that, instead of having each word as a dimension, use a pre-defined set of word categories to represent the documents and classify the words available into one of these categories. Examples include the *Linguistic Inquiry and Word Count* (LIWC), which references a dictionary of

* Corresponding author.

E-mail addresses: 72208635@mackenzista.com.br (M. A. Ferraria), 72208643@mackenzista.com.br (V. A. Ferraria), leandron@unicamp.br, ldecastrosilva@fgcu.edu (Leandro N. de Castro).

grammatical, psychological and content word categories, the *Part-of-Speech Tagging* (sTagging), which accounts for the definition and context of words [11]–[13], and the MRC, which uses a predefined dictionary to map words into their respective psycholinguistic information [14].

A third class of text structuring method investigated in this paper is based on the concept of *word embedding* [15]–[17], that is, each word is represented as a real-valued vector that encodes its context and meaning, such that words with similar meanings appear closer to one another in the vector space. This type of distributed representation of words is generated by specific neural network architectures. Examples of these approaches include the Doc2Vec and Word2Vec algorithms, which model each paragraph and word, respectively, as a numerical vector representing its meaning and main characteristics.

The fourth and last representation is based on sentence embeddings, an extension of word embeddings generated by deep neural architectures. Sentence embedding models [18]–[21], such as SBERT, generate a vector taking into consideration both the semantics and linguistic aspects of a sentence or phrase by making use of the position, context and how every word is being used in the sentence [22]. While based on word embeddings, these models differ from them since word embeddings only hold isolated information for each word, while sentence embeddings are capable of extracting relationship between words and capture contextual information of a group of words like sentences, phrases and paragraphs [23].

After the documents are structured, any type of machine learning algorithm can be used to extract knowledge from the data. Tasks like clustering, classification and association rule mining can be performed. This paper investigates the influence of different text representation schemes, more specifically BoW, LIWC, sTagging, MRC, Doc2Vec, Word2Vec, and SBERT to prepare texts for being clustered by an *Artificial Immune Network* algorithm named aiNet [24]–[27]. To assess the performance of the algorithm, four datasets from the literature and two internal clustering measures were chosen.

The paper is organized as follows. Section II provides some background knowledge on the Immune Network Theory, the aiNet Algorithm, and the different text representation schemes used in the paper. Section III describes the implementation performed, results obtained and a discussion. The paper is concluded in Section IV with some general discussions and future research.

II. Background Knowledge

This research investigates the use of different text representation schemes combined with the aiNet algorithm to detect and extract clusters from text data. This section briefly reviews the biological phenomenon from which aiNet was inspired, the aiNet learning algorithm, and the different text representation schemes that will be used in the research.

A. Immune Network Theory

Among the most diverse components present in the immune system, *antibodies* (Ab) play a key role in its learning and evolution [28]. They work as a line of defense, recognizing and binding with *antigens* (Ag), thus generating Ag-Ab complexes that are then identified and destroyed by other immune cells [29], [30]. These cellular interactions are responsible for regulating and allowing the evolution of the Immune System (IS) [29] and the immune networks are responsible for key activities of immune cells, such as the emergence of memory cells and the *self - non-self discrimination* [30]–[32].

The *immune network theory* is a proposal that aims at explaining how the adaptive immune system works. It is based on the notion that antibodies contain receptors capable of recognizing one another and the foreign disease-causing agents, called antigens. This self-recognition capability implies that immune cells and molecules are naturally connected to one another forming an internal network in a dynamical equilibrium state. The invasion of antigens would then disturb the network, promoting a change in its internal state. As the network already contains the receptors for such antigens, these would be called *internal images* of the antigens [30].

Ag-Ab interactions are extremely important for the learning and evolutionary processes of the IS [32], since the affinity of these interactions help guiding the creation of *memory cells*, that is, a set of specialized cells that are rescued by the IS to promote a faster and more effective response to future invasions of the previously seen antigens [28].

Immune system adaptation to foreign antigens is based on a learning and evolutionary process that allows the maturation of antibody receptors so that they become increasingly better at recognizing antigens and, also, the increase in the sets of memory cells to known antigens. This means that after the immune system eliminates a certain disease-causing agent, its immune cells and molecules are more adapted (i.e., with greater affinity) to that specific antigen, and the concentration of these cells also increased significantly, ensuring an effective response to future invasions of the same or similar antigen [28], [30], [31].

In summary, an adaptive immune response involves the recognition of antigenic patterns, followed by the expansion (cellular reproduction) of high-affinity cells, antibody maturation (i.e., mutations that lead to better Ag-Ab affinity match), and clonal expansion (i.e., the increase in number of high-affinity cells) [28], [32]. Altogether, these processes are called *clonal selection and affinity maturation*. It is worth mentioning that antibody mutation during clonal expansion is inversely proportional to the Ag-Ab affinity, that is, the higher the affinity, the smaller the mutation rate, and vice-versa [29]. Also, the immune network theory brings an explanation for the structure (architecture) and dynamics of immune cells and molecules.

By observing the essence of the immune system processes and their computational counterpart Artificial Immune System (AIS), it is possible to find several features that make them applicable to different types of tasks. For instance, Ag-Ab interactions are intrinsically a pattern recognition process, and clonal selection and affinity maturation are akin to an evolutionary search mechanism. The immune network theory adds another sophistication level to AIS by embedding a network structure to a system that was originally composed of separated individual components. When connections are added to the components of the system, pre-defined communication pathways are created and can be subjected to varying weights. By using these immune inspirations, it is possible to design algorithms for solving a vast array of problems, such as autonomous navigation, vehicle routing, clustering, classification, pattern recognition, and anomaly detection [30], [33]–[35].

B. aiNet: An Artificial Immune Network Model

The Artificial Immune Network model called aiNet is an algorithm inspired by the immune network theory aimed at clustering spatial data [29], [36]. aiNet takes as inspiration the pattern recognition of antigens by antibodies, the clonal expansion of high-affinity cells, the affinity proportional mutation of antibodies, the maintenance of high-affinity cells as immune memories, and the immune network theory that explains structural properties of immune cell repertoires.

In the aiNet metaphor, antigens are the input data (objects) while antibodies are the prototypes representing the immune network internal representations of the antigens, learnt from the input data. For the algorithm, antigens and antibodies are represented by *N*-dimensional vectors, therefore, Ag-Ab recognition is calculated using a similarity or dissimilarity measure [28]. The evaluation of the affinity between Ag and Ab is of paramount importance for the algorithm training process. Biologically, Ag-Ab recognition is based on the complementarity of their shapes, but for engineering purposes affinity can be measured either with similarity or dissimilarity measures [29].

Algorithm 1 presents a pseudo-code of the aiNet learning algorithm and its main steps. The algorithm works as follows. An initial set of antibodies *Ab* and a matrix of memory cells *M* are created, serving the purpose of maintaining sets of prototypes that will represent clusters of data in the available datasets. After initialization, an iterative search process starts until a certain number of iterations has been run. Within this loop, a number of steps occur. First, a partial random population is created and added to *Ab*. Then, for each input object (Ag), its affinity with all prototypes (Abs) is calculated and the N_b highest affinity ones are selected and cloned proportional to affinity (the higher the affinity, the larger the clone size) and mutated inversely proportional to affinity (the lower the affinity, the higher the mutation rate). A number ζ% of the highest affinity mutated clones are selected, the redundant ones, based on a similarity threshold ϵ, eliminated, and those whose affinity with the antigen are smaller than σs are eliminated. After all these steps are performed for each input object, the remaining prototypes are added to the set of *M* memory cells, and another suppression step removes redundancy within *M* .

Algorithm 1 The aiNet learning algorithm.

```
initialize the antibody set $Ab$
initialize the memory matrix $M$
while stopping criterion is not met do
    initialize a random population with size $m$ and concatenate
    it with $Ab$
    for each input object do
        calculate its affinity with every member of $Ab$
        select the $N_b$ highest affinity antibodies
        clone the $N_b$ antibodies proportional to their affinity
        mutate the clone inversely proportionally to their affinity
        select the $\zeta$% highest affinity clones
        for each clone in the selected clone set do
            if aff > $\sigma_p$ then remove (prune) it from the set of
            selected clones
            end if
            if aff < $\sigma_s$ then remove (suppression) it from the set of
            selected clones
            end if
        end for
    end for
    calculate the affinity between all objects in $M$ and suppress
    those with affinity smaller than $\sigma_s$
    replace $Ab$ with $M$
end while
```

The main parameters necessary to run the algorithm are:

- max_{it}: maximal number of iterations;
- N_b: number of antibodies to be selected for cloning;
- N_c: multiplier of the number of clones to be generated;
- ζ%: percentage of antibodies to be selected;
- σ_p: pruning threshold;
- σ_s: suppression threshold used to control redundancy.

At the end of aiNet training, the memory matrix *M* generated in the last iteration will contain the prototypes found based on the learning from the input data. From this matrix, it is possible to calculate the affinity among its antibodies and find groups of prototypes representing groups of data. The division into groups, also known as clustering, will be performed by applying the Minimal Spanning Tree (MST) [37] followed by a pruning method for inconsistent edges. The MST together with the pruning method will allow the separation of the data into cohesive groups, that is, groups with elements closer to one another [38], [39]. The idea is very simple. After building the MST among all memory antibodies, remove those edges whose weight are significantly larger than the average of nearby edge weights on both sides of the edge.

The aiNet dynamics followed by the application of the MST pruning method described above, makes it a clustering algorithm suitable for solving problems in which the clusters present differences in density. This is because aiNet will tend to uniformly place antibodies in regions of the space where the objects in the dataset are located. After that, the MST pruning method will look for links in the MST that are inconsistent and prune these links. Inconsistency here means being significantly longer than those in neighboring regions, what naturally implements a cluster separation method that searches for variations in the density of data in their original space.

C. Text Representation

Text Analysis refers to the process of extracting knowledge from texts based on their content [40]. As a computer is not intrinsically capable of understanding texts, it is necessary to establish an interface between the language of computers and the human language, which is obtained through computable numeric representations. Text Analysis is part of Natural Language Processing (NLP), which aims to study ways to model human language for computational purposes, thus allowing computers to be able to understand the texts to be analyzed [5], [41].

Over the past years, many works have been proposed involving the application of deep neural networks to text analysis and NLP [42], [43]. Despite that, not so many papers have addressed the problem of comparing the more traditional methods for text structuring (or representation) among themselves and with those based on deep network architectures. Useful review works in this direction include the papers [44]–[47].

The present paper investigates different text representation schemes commonly found in the literature: N-Gram, through Bag of Words (BoW); Linguistic Inquiry and Word Count (LIWC); Part of Speech Tagging (PoS Tagging); MRC; Doc2Vec; Word2Vec; and SBERT. These will be employed to structure text documents to be used by the aiNet clustering algorithm. This section provides a brief review of these text structuring methods.

1. BoW

N-Gram is a simple model used in natural language processing to represent textual data where every sequence of *N* tokens is considered a new feature [48]. Sequences with a size of one can either be referred to as a Unigram or as a Bag of Words [49]. To simplify, it will be called Bag Of Words (BoW) in this paper. While the Bag of Words is the most popular model, models using greater values of *N* are extremely useful for text prediction, translation techniques and search engines [48], making them more versatile.

This technique consists of creating a dictionary from the sentences used as input disregarding grammar and context. BoW is often used in conjunction with other pre-processing techniques to remove meaningless words, such as *stopwords*, and standardize the input data by removing special characters and keeping all words in uppercase or lowercase. After these steps, it is necessary to determine the weight of each feature, formed by *N* sequential tokens, in the document, and this was performed here using the *Term Frequency Inverse Document Frequency* (TF-IDF) method.

TF-IDF is a statistical measure that determines the importance (weight) of each sequence of *N* words in the analyzed documents. This measure is calculated using the relative frequency of each sequence in the analyzed document in relation to the inverse of the number of documents that have the word being evaluated [50]:

$$TF(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{1}$$

$$IDF(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|} \tag{2}$$

$$TFIDF(t,d,D) = TF(t,d) \cdot IDF(t,D) \tag{3}$$

where *TF* (*t, d*) is the relative frequency of term *t* within document *d*, $f_{t,d}$ is the number of times the sequence of *N* terms *t* occurs in document *d*, *IDF (t, D)* is the inverse document frequency, and *N* is the total number of documents in the corpus *D*. The higher the TF-IDF value, the greater the relevance of a sequence in the document [51].

2. LIWC

The *Linguistic Inquiry and Word Count* (LIWC) [12] is a textual analysis tool composed of: four categories of general descriptors (total word count, number of words per sentence, percentage of words captured by the dictionary, and percentage of words with more than six letters); seven categories of personal concern (e.g. *work*, *home*, *leisure activities*); three speech categories (consent, e.g. *agree*, *OK*, *yes*; onomatopoeia, e.g. *Er*, *hm*, *umm*; fillers, e.g. *so*, *such as*, *is*, *hum*, *well*); and 12 punctuation categories (e.g. *dots*, *commas*, *etc*). In addition, it has 22 standardized linguistic dimensions (e.g. *the percentage of words in the text that are pronouns*, *articles*, *auxiliary verbs*, *etc.*) and 32 psychological constructor word categories (e.g. *affect*, *cognition*, *biological processes*) [12]. It should be noted that LIWC extracts meta-attributes from a document rather than representing the document by its words, like BoW does.

3. Part of Speech Tagging

Part of Speech Tagging, called here sTagger, originally written by Kristina Toutanova [52], is a Part-of-Speech (POS) tool whose function is to assign each word of the text a tag, such as noun, verb, adjective, etc. In the case of sTagger, its main differential comes from the use of a bidirectional dependency network to predict tagged sequence of words. This bidirectional approach allows it to better extract words' interactions and conditioning features [52].

When structuring a document via sTagger, a count is made of the number of words in each tag. Thus, at the end of the structuring process, there is a matrix in which each attribute refers to a tag [13], [52], [53].

4. MRC

The MRC is a machine usable dictionary of psycholinguistic information containing 150,837 words, where each word is composed of up to 26 linguistic and psycholinguistic attributes. The attributes are obtained from publicly available sources and structured in a single dictionary [14]. The MRC representation consists of mapping words contained in the dictionary into a vector representing the 26 attributes mentioned before.

5. Word2Vec

Word2Vec is a family of algorithms and models that are used to learn *word embedding* from texts and the relationships between words. A word embedding is a representation of a word that encodes its meaning in a real-valued lower-dimensional vector, where the representation of words with similar meaning are closer together in the vector space.

The application of word embedding grants the *Word2Vec* model the ability to generate real-valued dense vectors for each word that is capable of capturing each linguistic regularity and linear relationships, allowing those vectors to be applied to mathematics operations like + and -.

An example of the linguistic regularities and their math properties is subtracting the vector representation of the word *man* from the vector representation of the word *king*, resulting in a vector that is close to the vector representation of the word *queen* [54].

6. Doc2Vec

Doc2Vec is a set of paragraph embedding models, inspired by the Word2Vec model, but with emphasis on documents, and that produces better results than averaging all the word vectors in a document.

A paragraph embedding is a representation of a variable-length text, such as documents, sentences and paragraphs, by real-valued vectors with fixed-length features [55]. The main difference between Word2Vec and Doc2Vec is that besides the word vectors generated by both models, Doc2Vec also has a single shared paragraph embedding which allows to better represent the document and its meaning [55].

The Doc2Vec model learns the paragraph embedding of a text by training to predict the vector representation for each word in a document in conjunction with a vector representing the paragraph, the paragraph vector. The predict task of the model concatenates the paragraph vector with word vectors to predict the next word in the context. The outcome of the learning task is a model whose vectors are capable of representing documents in a vector space.

7. SBERT

SBERT is a sentence embedding representation model built on top of BERT [56] and RoBERTa models [57]. These models present state-of-the-art performances for many text mining tasks, but have poorer performances when used for semantic-similarity, making them unsuitable for clustering tasks [58].

Since this embedding was created with the goal of extending the state-of-the-art results provided by those models for sentence embedding generation, this representation makes use of an elegant modification on the BERT/RoBERTa models by adding a pooling operation to its output. In order to provide a more contextualized and semantically meaningful embedding, the BERT/RoBERTa are first fine-tuned with siamese and triplet networks [58].

The SBERT representation can be generated by using many of the pre-trained models available in public repositories, like Hugging Face Hub. Since this representation uses pre-trained models, the quality of the embeddings generated may vary depending on the model used.

III. Performance Assessment

The goal of this paper is to investigate how different text representation methods influence the clustering performance of aiNet. To do so, three types of text representations were chosen: one standard vector space model (BoW); three grammar-based models (LIWC, sTagger, MRC); and two word embedding models (Word2Vec and Doc2Vec). Two clustering internal measures were selected for comparison: the Dunn (DU) index and the Davies Bouldin (DB) index. The methodology used, results obtained, and discussions are presented in this section.

For this research the *stsb-roberta-large* model [58] was chosen since some preliminary tests with four different models indicated that the *stsb-roberta-large* consistently generated the best results for all datasets.

A. Methodology

This subsection describes the experimental methodology employed. It starts by providing some distinctions among the representations and then follows with the hyperparametrization of aiNet, Word2Vec and Doc2Vec. Then, the datasets chosen are summarized and the evaluation measures described.

1. Some Comments on the Selected Representations

The three classes of text representations are considerably different from one another. BoW works by finding and weighing tokens that are expected to have a high discriminating capability among the text categories, but usually results in very high dimensional feature vectors. Grammar-based representations, such as LIWC, sTagger and MRC, are characterized by a limited number of word categories, but privilege a low dimensional representation of word categories in detriment of the context. Word embeddings, like Word2Vec and Doc2Vec, by contrast, try to extract the semantic meaning of the texts by representing the words by means of word vectors that are expected to capture the context of each word or document. SBERT generates fixed-size vector representations of sentences or short texts, extending the concept of word embeddings to the sentence level. These representation schemes will be used and compared here.

2. Some Comments on the Pre-Processing Step

In recent years, questions have been raised about the need of a pre-processing step for generating text representations. This is because most of the recently developed representations, like SBERT, are based on deep neural networks, for which the removal of any word can have an impact on the contextual and semantic understanding of the model, potentially leading to worse representations [59], [60].

3. aiNet's Hyperparameters

The aiNet parameters were chosen based on an iterative process aimed to maximize the selected evaluation metrics while making it possible to investigate how different representations behave when paired with the model, assessing their strengths and weaknesses. To maintain consistency when comparing different representation methods, most of the aiNet hyperparameters were fixed for all representations and datasets used, as follows:

- max_{it} = 50;
- N_b = 500;
- N_c = 40;
- ζ% = 10%;
- σ_s = 0.05.

This consistency is important because if aiNet had to be fine tuned for each representation it would be very difficult to compare the results and understand how each representation method influences the clustering results.

However, it is well known that some representation models, like BoW, generate high dimensional datasets, and calculating the similarity among the immune cells and between them and the input objects may require some tuning. Some preliminary experiments showed that higher values for the pruning threshold σ_p should be adopted for higher dimensional spaces, so the defined value for BoW, Word2Vec, and Doc2Vec was 0.9, while the remaining representations had σ_p = 0.5.

4. Word2Vec and Doc2Vec Hyper-Parameters

Word2Vec and Doc2Vec are the only representations that require hyper-parameter tuning. Since both models are based on the same algorithm, they share most of their parameters and can be trained using the same package, a very popular library called *Gensim* [61]:

- *vector_size* = 50;
- *window* = 5;
- *min_count* = 2;
- *epochs* = 40;
- *negative* = 5.

The Word2Vec was trained using the preprocessed sentences of the datasets and *sg* = 0. For the Doc2Vec *dm* = 0 (PV-DBOW).

5. Datasets

To test aiNet's clustering capability for different text representation schemes, four datasets from the literature were selected:

- *Sentiment Labelled Collection*: a collection of 3 datasets from 3 different websites (Amazon, Yelp and IMDB) containing users' reviews. Each dataset has 1,000 objects with 500 positive reviews and 500 negative ones. The datasets are available at https://archive.ics.uci. edu/ml/datasets/sentiment+labelled+sentences.
- *SMS Spam Data*: a dataset collected by [62] containing 5,572 messages (4,825 ham and 747 spam). The dataset is available at https://archive.ics.uci.edu/ml/ datasets/SMS+Spam+Collection.

It is important to highlight that although the selected datasets are textual, each one of them represent a different type of text with its own characteristics. The SMS Spam Data is considered a short text dataset due to its character limitation and informal nature, making it likely the use of *slangs* and shorter texts that sometimes need to be combined to present the whole context [63]. The Sentiment Labelled Collection, by contrast, is composed of reviews extracted from three different websites with substantially different products, but grouped based on language and semantics. Reviews are different from text messages since their nature revolves around more descriptive texts and a semantics that expresses how someone feels about the reviewed product.

The difference among the selected datasets can have an impact on the representations generated. For instance, models like the Bag of Words are likely to be more sensitive to the SMS Spam Collection because its shorter length could generate a more sparse representation of each text.

6. Evaluation Measures

The Dunn and Davies-Bouldin indices [3] were used to assess the quality of the clusters obtained [3]. To calculate them, it is necessary to evaluate the cohesion (compactness) and separation of each cluster [64], what can be performed using intra (Eq. (4)) and inter-cluster distances (Eq. (5)):

$$Intra(g_i) = max_{x,y \in g_i}\{d(x,y)\} \tag{4}$$

$$Inter(g_i, g_j) = \frac{1}{|g_i| \cdot |g_j|} \sum d(x,y) | x \in g_i, y \in g_j \tag{5}$$

where g_i refers to group *i*, $|g_i|$ is the number of objects in group *i*, and *d*(*x*, *y*) is a distance measure between objects *x* and *y*.

7. Dunn Index (DU)

The Dunn Index combines both the inter- and intra-cluster distance to provide a cluster quality measure. It ranges over the interval [0, ∞], where the higher the values, the more cohesive and separated the clusters [65]:

$$DU(g) = \min_{i=1,\ldots,k} \left\{ \min_{j=1,\ldots,k; j \neq i} \left\{ \frac{\text{Inter}\left(g_i, g_j\right)}{\max\limits_{l=1,\ldots,k} \{\text{Intra}\left(g_l\right)\}} \right\} \right\} \tag{6}$$

where g is the resultant clustering, *k* is the number of clusters, g_i is a cluster from the dataset, and *Inter*(.) and *Intra*(.) are the inter- and intracluster measures defined previously.

8. Davies-Bouldin Index (DB)

Similarly to the DU, the Davies-Bouldin Index [66] also uses the inter- and intra-cluster distances to determine its value, but combining these measures in a different way:

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j} \left(\frac{\text{Intra}\left(g_i\right) + \text{Intra}\left(g_j\right)}{\text{Inter}\left(g_i, g_j\right)} \right) \tag{7}$$

where g is the resultant clustering, k is the number of clusters, g_i is a cluster from the dataset, and *Inter*(.) and *Intra*(.) are the inter- and intracluster measures defined previously.

The DB index measures the similarity between each cluster [67], [68] and it varies in the range [0, ∞], where the lower the values, the better the quality.

9. Experimental Results

The experiments were performed in Python with the use of third party libraries such as *Numpy*, *Spacy*, *Scikit-learn*, *Gensim*, *Spacy-stanza* and *Pandas*. With all the representations, hyperparameters, datasets, and evaluation measures defined, the experiments were organized as follows:

- Each text representation was generated from a dataset after going through a processing pipeline adapted to the specificity of each representation. The structured texts were then used to train the aiNet model.
- As the aiNet relies heavily on the Ag-Ab affinity and considering that some of the text representations generate high-dimensional input vectors, the *cosine similarity* was chosen as the affinity measure in all experiments.
- Based on the clusters determined by aiNet, its performance was evaluated using the Dunn Index (DU) and the Davies-Bouldin Index (DB) for each representation used and for all selected datasets.
- Since aiNet is a non-deterministic algorithm, 10 experiments were performed for each text representation and the results presented are the average and standard deviation of the 10 results.

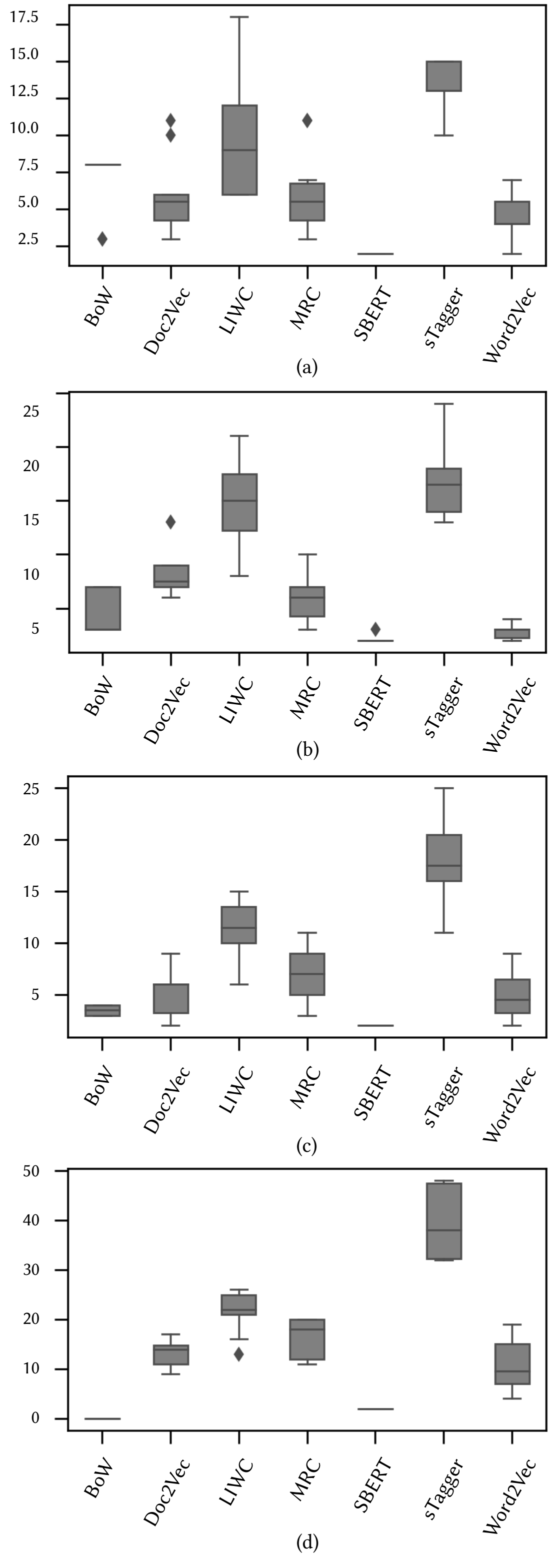
B. Results and Discussion

Table I summarizes the clustering results of the seven text structuring methods when used in conjunction with aiNet. By analyzing all results it is possible to observe similar trends between each representation across all four datasets, with the Bag of Words and MRC consistently presenting the best and the worst results for DU and DB, respectively. Another similarity is related with the total number of clusters identified, as presented in Fig. 1. It can be observed that the number of clusters resultant from each representation follows the same pattern.

While the state of the art representation, SBERT, presented only the second best results for both indices for the first three datasets, it presented, by a vast margin, the best results for the SMS Spam Collection. Another aspect to observe is that SBERT constantly generated the smallest number of clusters, closer to the number of classes of each dataset.

Considering the different sets of text representations studied in this paper, high dimensional feature vectors, grammar-based, word embedding, and sentence embedding, it is possible to note similar results between representations that belong to the same set, for example Doc2Vec and Word2Vec present similar results for both measures (DU and DB).

Interestingly, Bag of Words performed competitively even when compared with the state of the art, SBERT, and consistently outperformed word embedding and grammar-based representations, achieving the second best overall result with the Sentiment Labelled Review Datasets. Although the BoW model in general presented very good results for the Sentiment Labelled Review Datasets, it also presented a very undesired behaviour when paired with the SMS Spam Collection Dataset. In this scenario, despite its higher value of σ_p, the aiNet paired with BoW was unable to find more than one cluster, reason why its performance was zeroed.



(a)

(b)

(c)

(d)

Fig. 1. Boxplots presenting the number of clusters per representation over 10 executions. (a) Amazon Labelled Reviews. (b) IMDB Labelled Reviews. (c) Yelp Labelled Reviews. (d) SMS Spam Collection.

TABLE I. Performance Evaluation of the Seven Text Structuring Methods (TSM) in the Four Datasets Chosen, Detaching the Best Performance for Each Dataset. DU: Dunn Index; DB: Davies Bouldin Index. The SBERT Representation, As Previously Mentioned, Was Generated Using the Sentence-transformers/stsb-large-model Available at Https://huggingface.co/sentence-transformers/ Stsb-roberta-large

	TSM	DU	DB
Amazon Labelled Reviews	**BoW**	**0.98 ± 0.00**	**1.95 ± 0.13**
	Doc2Vec	0.29 ± 0.02	3.73 ± 0.55
	LIWC	0.32 ± 0.03	3.65 ± 0.20
	MRC	0.19 ± 0.07	4.71 ± 0.57
	SBERT	0.77 ± 0.02	2.49 ± 0.05
	sTagger	0.26 ± 0.03	3.70 ± 0.20
	Word2Vec	0.26 ± 0.07	4.53 ± 0.89
IMDB Labelled Reviews	**BoW**	**0.99 ± 0.00**	**2.01 ± 0.00**
	Doc2Vec	0.25 ± 0.06	4.91 ± 0.61
	LIWC	0.31 ± 0.03	3.79 ± 0.31
	MRC	0.18 ± 0.05	5.17 ± 0.53
	SBERT	0.76 ± 0.01	2.48 ± 0.14
	sTagger	0.23 ± 0.02	3.23 ± 0.24
	Word2Vec	0.23 ± 0.05	4.26 ± 0.47
Yelp Labelled Reviews	**BoW**	**0.98 ± 0.00**	**2.02 ± 0.00**
	Doc2Vec	0.30 ± 0.05	3.76 ± 0.28
	LIWC	0.32 ± 0.04	3.46 ± 0.12
	MRC	0.21 ± 0.05	4.92 ± 0.55
	SBERT	0.77 ± 0.01	2.52 ± 0.03
	sTagger	0.27 ± 0.02	3.15 ± 0.15
	Word2Vec	0.26 ± 0.07	4.85 ± 1.21
SMS Spam Collection	BoW	0.00 ± 0.00	0.00 ± 0.00
	Doc2Vec	0.33 ± 0.02	3.50 ± 0.28
	LIWC	0.25 ± 0.00	4.43 ± 0.13
	MRC	0.09 ± 0.02	7.30 ± 0.43
	SBERT	**0.77 ± 0.01**	**2.57 ± 0.03**
	sTagger	0.14 ± 0.01	4.37 ± 0.15
	Word2Vec	0.09 ± 0.02	7.30 ± 0.43

Further investigations showed that the higher number of objects in the SMS Spam Collection resulted in a very sparse representation for BoW, with over 7,200 dimensions, almost a three times increase when compared with the BoW dimension for the Amazon Labelled Reviews. The sparsity found with this representation posed a challenge for the aiNet model since the Ag-Ab interactions ended up presenting very high values and thus all antibodies were consistently eliminated every iteration, resulting in an empty memory matrix after the training process.

In addition to the higher dimension, the SMS Spam Collection also presented a very unbalanced proportion between its original clusters, with the spam class having six times more objects than the other class, making it much more difficult to extract patterns from each cluster. This proportion made it difficult for the SBERT representation to detect more than one cluster in a couple of executions, which indicated that a fine tuning of the algorithm can lead to an even better performance for this representation.

Another point that can be observed when analyzing the results of grammar-based representations is that their dictionaries have a significant impact on the final representation dimensionality with each representation using different dictionaries, each with its own categories. Due to the grammar-based representations dependency on a predefined dictionary, the final representation is subjected to the words available in the dictionary and mismatches of words between the texts and dictionaries can occur. Such scenario becomes evident when older dictionaries are paired with modern texts, such as internet discussions and reviews, given that it does not account for today's dialects and *slangs*. Usually, some of the grammar-based representations have a specific *category* that is used to account the mismatches, such as LIWC, but not all of them have it, as is the case for MRC.

The issues mentioned above can be observed when assessing the results of MRC, which presented the worst results for all datasets. The dictionary used by MRC was released in 1988 and has several relevant psychological attributes that are difficult to be synthesized and some of them do not have value for all the words contained in its dictionary and some of the words are not contained in it. Due to the complexity of the attributes and the date the dictionary was created, the probability of a word from the texts of the chosen datasets being present and having values for all the features is low, causing words not to have a significant value or to have a sparse representation, thus impacting its performance. It is also possible to infer from the results that the LIWC representation, which has a more complete and more recent dictionary, created in 2015, that it can also account for words that are not present in it, has better metric values and greater number of clusters of the grammar-based category.

The results also emphasize that the SMS database is the most complex to represent, resulting in lower metric values for most representations, with the exception of Doc2Vec and Word2Vec, which presented their best results among all datasets.

Fig. 1 shows the boxplot of the number of clusters found by aiNet for each of the four datasets over the 10 runs performed. Note that all datasets are originally divided into two classes, but the class labels are not used to train aiNet. It is a general tendency that sTagger generates more clusters than the other approaches, followed by LIWC. Also, it was noted that Word2Vec and Doc2Vec present similar behaviors with small numbers of clusters.

IV. Conclusions and Future Trends

This paper aimed at investigating the influence of seven different text structuring methods to be used in conjunction with the aiNet clustering algorithm. These methods fall into four categories: vector space models, grammar-based models, word embeddings, and sentence embeddings. Each category has a specific form of structuring the text, capturing or not information like syntax and context. Performance evaluation was made using four datasets from the literature, and internal clustering measures (Dunn and Davies Bouldin indices).

After running a number of experiments and analyzing the results, it was possible to observe that the aiNet's pruning threshold is sensitive to the dimensionality of the representation, especially those with more sparse representations, like the Bag of Words (BoW) model.

Considering all the results obtained in this paper, the state-of-the-art model, SBERT, consistently presented good results on all selected datasets, while other distributed representations, Doc2Vec and Word2Vec, did not perform as well, especially when paired with the Sentiment Labelled Review Dataset. The results suggest that this type of representation performs better with datasets containing a larger number of objects, that is, a larger variety of words. This observation is in contrast with the remaining representations, which performed worse when used with the SMS Spam Collection.

Although the BoW representation is the simplest in terms of generation when compared with the others studied, its results were fairly competitive, especially with the state of the art representation. While it is true that this representation was unable to present any result for the SMS Spam Collection, this reinforces the BoW's main weakness: its dimensionalty (the larger number of objects provides a greater variety of words, which greatly increases the dimensionality of this representation making it extremely sparse and decreasing the effectiveness of similarity techniques that are intensively used by the aiNet algorithm.)

The results provided interesting insights about the peculiarity of each type of text representation. It is clear the need of running new experiments with larger datasets to further evaluate and improve the performance of the aiNet algorithm. Another point of improvement is the use of other high dimensional representations to further evaluate the impact of very sparse data matrices when paired with the aiNet algorithm. Finally, the results presented are relevant since they can be used as a baseline to fine tune the aiNet algorithm for each representation studied.

Acknowledgment

This work was financially supported by FAPESP, CNPq and MackPesquisa.

References

[1] C. S. Kumar, R. Santhosh, "Effective information retrieval and feature minimization technique for semantic web data," *Computers & Electrical Engineering*, vol. 81, p. 106518, 2020.

[2] S. S. Tandel, A. Jamadar, S. Dudugu, "A survey on text mining techniques," in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, 2019, pp. 1022–1026, IEEE.

[3] L. N. de Castro, D. G. Ferrari, *Introdução à mineração de dados*. Saraiva Educação SA, 2017.

[4] T. Jo, "Text mining: Studies in big data," 2019.

[5] K. Chowdhary, K. Chowdhary, "Natural language processing," *Fundamentals of artificial intelligence*, pp. 603–649, 2020.

[6] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, M. DATA, "Practical machine learning tools and techniques," in *Data Mining*, vol. 2, 2005.

[7] Y. HaCohen-Kerner, D. Miller, Y. Yigal, "The influence of preprocessing on text classification using a bag- of-words representation," *PloS one*, vol. 15, no. 5, p. e0232525, 2020.

[8] G. Miner, J. Elder IV, A. Fast, T. Hill, R. Nisbet, D. Delen, *Practical text mining and statistical analysis for non-structured text data applications*. Academic Press, 2012.

[9] W. A. Qader, M. M. Ameen, B. I. Ahmed, "An overview of bag of words;importance, implementation, applications, and challenges," in *2019 International Engineering Conference (IEC)*, 2019, pp. 200–204.

[10] D. Yan, K. Li, S. Gu, L. Yang, "Network-based bag-of- words model for text classification," *IEEE Access*, vol. 8, pp. 82641–82652, 2020.

[11] W. A. Qader, M. M. Ameen, B. I. Ahmed, "An overview of bag of words; importance, implementation, applications, and challenges," in *2019 international engineering conference (IEC)*, 2019, pp. 200–204, IEEE.

[12] J. W. Pennebaker, M. E. Francis, R. J. Booth, "Linguistic inquiry and word count: Liwc 2001," *Mahway: Lawrence Erlbaum Associates*, vol. 71, no. 2001, p. 2001, 2001.

[13] A. Chiche, B. Yitagesu, "Part of speech tagging: a systematic review of deep learning and machine learning approaches," *Journal of Big Data*, vol. 9, no. 1, pp. 1–25, 2022.

[14] M. D. Wilson, "*MRC Psycholinguistic Database: Machine Usable Dictionary: Version 2.00*," *Behavior Research Methods, Instruments, & Computers*, vol. 20, pp. 6–10, 1988.

[15] J. Lastra-Díaz, J. Goikoetxea, M. A. Hadj Taieb, A. Garcia-Serrano, M. Ben Aouicha, E. Agirre, "A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 645–665, 2019, doi: 10.1016/j.engappai.2019.07.010.

[16] U. Naseem, I. Razzak, S. K. Khan, M. Prasad, "A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 5, pp. 1–35, 2021.

[17] F. Almeida, G. Xexéo, "Word embeddings: A survey," *arXiv preprint arXiv:1901.09069*, 2019.

[18] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, L. Li, "On the sentence embeddings from pre-trained language models," *arXiv preprint arXiv:2011.05864*, 2020.

[19] M. N. Moghadasi, Y. Zhuang, "Sent2vec: A new sentence embedding representation with sentimental semantic," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 4672–4680.

[20] T. Gao, X. Yao, D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," *arXiv preprint arXiv:2104.08821*, 2021.

[21] T. Jiang, J. Jiao, S. Huang, Z. Zhang, D. Wang, F. Zhuang, F. Wei, H. Huang, D. Deng, Q. Zhang, "PromptBERT: Improving BERT sentence embeddings with prompts," *arXiv preprint arXiv:2201.04337*, 2022.

[22] X. Zhu, T. Li, G. De Melo, "Exploring semantic properties of sentence embeddings," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 632–637.

[23] M. K. Mishra, J. Viradiya, "Survey of sentence embedding methods," *International Journal of Applied Science and Computations*, vol. 6, no. 3, pp. 592–592, 2019.

[24] S. A. Hofmeyr, S. Forrest, "Immunity by design: An artificial immune system," in *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2*, 1999, pp. 1289–1296, Citeseer.

[25] F. A. González, D. Dasgupta, "Anomaly detection using real-valued negative selection," *Genetic Programming and Evolvable Machines*, vol. 4, pp. 383– 403, 2003.

[26] E. Bendiab, M. K. Kholladi, "The negative selection algorithm: a supervised learning approach for skin detection and classification," *International Journal of Computer Science and Network Security*, vol. 10, pp. 86– 92, 2010.

[27] M. Ayara, J. Timmis, R. de Lemos, L. N. de Castro, R. Duncan, "Negative selection: How to generate detectors," in *Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS)*, vol. 1, 2002, pp. 89–98, University of Kent at Canterbury Printing Unit Canterbury, UK.

[28] J. Timmis, M. Neal, J. Hunt, "An artificial immune system for data analysis," *Biosystems*, vol. 55, no. 1-3, pp. 143–150, 2000.

[29] L. N. de Castro, F. J. Von Zuben, "aiNet: an artificial immune network for data analysis," in *Data mining: a heuristic approach*, IGI Global, 2002, pp. 231–260.

[30] D. Dasgupta, S. Yu, F. Nino, "Recent advances in artificial immune systems: models and applications," *Applied Soft Computing*, vol. 11, no. 2, pp. 1574–1587, 2011.

[31] J. Greensmith, A. Whitbrook, U. Aickelin, "Artificial immune systems," *Handbook of Metaheuristics*, pp. 421– 448, 2010.

[32] J. Timmis, "Artificial immune systems–today and tomorrow," *Natural computing*, vol. 6, no. 1, p. 1, 2007.

[33] N. Bayar, S. Darmoul, S. Hajri-Gabouj, H. Pierreval, "Fault detection, diagnosis and recovery using artificial immune systems: A review," *Engineering Applications of Artificial Intelligence*, vol. 46, pp. 43–57, 2015, doi: https://doi.org/10.1016/j.engappai.2015.08.006.

[34] S. Alhasan, G. Abdul-Salaam, L. Bayor, K. Oliver, "Intrusion detection system based on artificial immune system: A review," in *2021 International Conference on Cyber Security and Internet of Things (ICSIoT)*, 2021, pp. 7–14.

[35] L. N. de Castro, J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer- Verlag UK, 2002.

[36] I. Čisar, S. M. Čisar, B. Popović, K. Kuk, I. Vuković, "Application of artificial immune networks in continuous function optimizations," *Acta Polytechnica Hungarica*, vol. 19, no. 7, pp. 53–164, 2022.

[37] P. C. Pop, "The generalized minimum spanning tree problem: An overview of formulations, solution procedures and latest advances," *European Journal of Operational Research*, vol. 283, no. 1, pp. 1–15, 2020.

[38] D. Cheng, Q. Zhu, J. Huang, Q. Wu, L. Yang, "Clustering with local density peaks-based minimum spanning tree," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 2, pp. 374–387, 2019.

[39] G. Mishra, S. K. Mohanty, "A fast hybrid clustering technique based on local nearest neighbor using minimum spanning tree," *Expert Systems with Applications*, vol. 132, pp. 28–43, 2019.

[40] M. L. Jockers, R. Thalken, *Text analysis with R*. Springer, 2020.

[41] J. Hirschberg, C. D. Manning, "Advances in natural language processing,"

Science, vol. 349, no. 6245, pp. 261–266, 2015.

[42] J. Chai, A. Li, "Deep learning in natural language processing: A state-of-the-art survey," in *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, 2019, pp. 1–6.

[43] D. W. Otter, J. R. Medina, J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2021, doi: 10.1109/TNNLS.2020.2979670.

[44] K. Singh, H. Devi, A. Mahanta, "Document representation techniques and their effect on the document clustering and classification: A review.," *International Journal of Advanced Research in Computer Science*, vol. 8, 2017.

[45] M. H. Ahmed, S. Tiun, N. Omar, N. S. Sani, "Short text clustering algorithms, application and challenges: A survey," *Applied Sciences*, vol. 13, no. 1, p. 342, 2023.

[46] K. Babić, S. Martinčić-Ipšić, A. Meštrović, "Survey of neural text representation models," *Information*, vol. 11, no. 11, p. 511, 2020.

[47] S. A. Farimani, M. V. Jahan, A. Milani Fard, "From text representation to financial market prediction: A literature review," *Information*, vol. 13, no. 10, p. 466, 2022.

[48] G. E. Pibiri, R. Venturini, "Handling massive n-gram datasets efficiently," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 2, pp. 1–41, 2019.

[49] M. Schonlau, N. Guenther, "Text mining using n- grams," *Schonlau, M., Guenther, N. Sucholutsky, I. Text mining using n-gram variables. The Stata Journal*, vol. 17, no. 4, pp. 866–881, 2017.

[50] D. E. Cahyani, I. Patasik, "Performance comparison of TF-IDF and word2vec models for emotion text classification," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2780–2788, 2021.

[51] M. Das, S. Kamalanathan, P. Alphonse, "A comparative study on tf-idf feature weighting method and its analysis using unstructured dataset.," in *COLINS*, 2021, pp. 98–107.

[52] K. Toutanova, D. Klein, C. D. Manning, Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 2003, pp. 252–259.

[53] J. Awwalu, S. E.-Y. Abdullahi, A. E. Evwiekpaefe, "Parts of speech tagging: a review of techniques," *Fudma Journal of Sciences*, vol. 4, no. 2, pp. 712–721, 2020.

[54] T. Mikolov, K. Chen, G. Corrado, J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[55] Q. Le, T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, 2014, pp. 1188–1196, PMLR.

[56] A. Rogers, O. Kovaleva, A. Rumshisky, "A primer in BERTology: What we know about how BERT works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, 2021.

[57] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019.

[58] N. Reimers, I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[59] Z. Rahimi, M. M. Homayounpour, "The impact of preprocessing on word embedding quality: A comparative study," *Language Resources and Evaluation*, vol. 57, no. 1, pp. 257–291, 2023.

[60] K. V. Ghag, K. Shah, "Comparative analysis of effect of stopwords removal on sentiment classification," in *2015 international conference on computer, communication and control (IC4)*, 2015, pp. 1–6, IEEE.

[61] "Gensim." https://github.com/ RaRe-Technologies/gensim.

[62] T. A. Almeida, J. M. G. Hidalgo, A. Yamakami, "Contributions to the study of sms spam filtering: new collection and results," in *Proceedings of the 11th ACM symposium on Document engineering*, 2011, pp. 259–262.

[63] H. Yin, X. Song, S. Yang, G. Huang, J. Li, "Representation learning for short text clustering," in *Web Information Systems Engineering–WISE 2021: 22nd International Conference on Web Information Systems Engineering, WISE 2021, Melbourne, VIC, Australia, October 26–29, 2021, Proceedings, Part II 22*, 2021, pp. 321–335, Springer.

[64] W. Wu, H. Xiong, S. Shekhar, J. He, A.-H. Tan, C.-L. Tan, S.-Y. Sung, "On quantitative evaluation of clustering systems," *Clustering and information retrieval*, pp. 105– 133, 2004.

[65] C.-E. B. Ncir, A. Hamza, W. Bouaguel, "Parallel and scalable dunn index for the validation of big data clusters," *Parallel Computing*, vol. 102, p. 102751, 2021.

[66] D. Davies, D. Bouldin, "A cluster separation measure," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-1, pp. 224 – 227, 05 1979, doi: 10.1109/TPAMI.1979.4766909.

[67] I. F. Ashari, R. Banjarnahor, D. R. Farida, S. P. Aisyah, A. P. Dewi, N. Humaya, *et al.*, "Application of data mining with the k-means clustering method and davies bouldin index for grouping imdb movies," *Journal of Applied Informatics and Computing*, vol. 6, no. 1, pp. 07– 15, 2022.

[68] M. Mughnyanti, S. Efendi, M. Zarlis, "Analysis of determining centroid clustering x-means algorithm with davies-bouldin index evaluation," in *IOP Conference Series: Materials Science and Engineering*, vol. 725, 2020, p. 012128, IOP Publishing.

Matheus A. Ferraria

Matheus is an aspiring student who started his master's degree at Mackenzie Presbyterian University (UPM) back in the first trimester of 2022. His study area is focused on Electrical Engineering and Computing, more specifically Natural Computing and Data Mining. He received his bachelor's degree in computer science from UPM at the end of 2021. His passion for research started during its finals days as a bachelor, after working with Leandro N. de Castro, writing his final paper, where he became fascinated by the nature's complexity and how it can inspire solutions to solve daily recurring problems. Passionate about learning and exploring his most recent academic experience is publishing and presenting an article at DCAI22 and he expects to contribute much more to the computer science community.

Vinicius A. Ferraria

A student inspired by studying the nature and its repercussion who graduated with a Bachelor's degree in Computer Science at Mackenzie Presbyterian University (UPM) in the second semester of 2021 and followed his studies by ingressing on a master's degree in Eletctrical Engineering and Computing in the field of Natural Computing and Data Mining also at UPM. His passion for natural computing started when he watched one of lecture about nature inspiring algorithms and since then he has worked in a couple article surrounding the Immune System and its infinity possibilities, ranging from his final thesis to scientific article published in conferences, the most recent one was an article published at DCAI22 related with an Artificial Immune System.

Leandro N. de Castro

Leandro has a B.Sc., M.Sc., and Ph.D. in Electrical Engineering from the Federal University of Goiás and Unicamp. He also holds an MBA in Strategic Business Management from the Catholic University of Santos. He was a Research Associate at the Computer Laboratory of the University of Kent in Canterbury (2001-2002), a Visiting Professor at the Technological University of Malaysia in 2005, a Visiting Professor at Unicamp (2012), and a Visiting Researcher at the University of Salamanca (2014). He was a research professor at the Master's Program in Informatics at Unisantos (2003-2008), and a research professor at the Graduate Program in Electrical Engineering and Computing at Universidade Presbiteriana Mackenzie (2008-2022). His main lines of research are Natural Computing and Machine Learning, with applications in Intelligent Data Analysis and Optimization. Leandro is the author of four academic books and has more than 250 papers published in national and international journals and conferences. He was the proponent and Editor-in-Chief of the International Journal of Natural Computing Research (IJNCR) from 2010 to 2015, published by IGI-Global. Leandro also has extensive entrepreneurial and leadership experience, having already participated in the founding of four Artificial Intelligence startups and invested, as an angel investor, in three others. He is currently a Visiting Professor at the Faculty of Technology at Unicamp and an Artificial Intelligence and Data Science Professor at the Florida Gulf Coast University.

Automatic Cell Counting With YOLOv5: A Fluorescence Microscopy Approach

Sebastián López Flórez[1,2]*, Alfonso González-Briones[1,2,4], Guillermo Hernández[1,2], Carlos Ramos[5,6], Fernando de la Prieta[1]

[1] BISITE Digital Innovation Hub, University of Salamanca, Edificio Multiusos I+D+i,Calle Espejo 2, 37007 Salamanca (Spain)
[2] Air Institute, IoT Digital Innovation Hub, 37188 Salamanca (Spain)
[3] Universidad Tecnológica de Pereira Cra. 27 N 10-02, Pereira, Risaralda (Colombia)
[4] Institute For Artificial Intelligence & Big Data, Universiti Malaysia Kelantan, City Campus, Pengkalan Chepa, Kota Bharu 16100, Kelantan (Malaysia)
[5] GECAD - Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Institute of Engineering - Polytechnic of Porto, Porto (Portugal)
[6] LASI - Intelligent Systems Associate Laboratory, Institute of Engineering - Polytechnic of Porto, Porto (Portugal)

Received 29 March 2023 | Accepted 24 July 2023 | Published 1 August 2023

Abstract

Counting cells in a Neubauer chamber on microbiological culture plates is a laborious task that depends on technical experience. As a result, efforts have been made to advance computer vision-based approaches, increasing efficiency and reliability through quantitative analysis of microorganisms and calculation of their characteristics, biomass concentration, and biological activity. However, variability that still persists in these processes poses a challenge that is yet to be overcome. In this work, we propose a solution adopting a YOLOv5 network model for automatic cell recognition and counting in a case study for laboratory cell detection using images from a CytoSMART Exact FL microscope. In this context, a dataset of 21 expert-labeled cell images was created, along with an extra Sperm DetectionV dataset of 1024 images for transfer learning. The dataset was trained using the pre-trained YOLOv5 algorithm with the Sperm DetectionV database. A laboratory test was also performed to confirm result's viability. Compared to YOLOv4, the current YOLOv5 model had accuracy, precision, recall, and F1 scores of 92%, 84%, 91%, and 87%, respectively. The YOLOv5 algorithm was also used for cell counting and compared to the current segmentation-based U-Net and OpenCV model that has been implemented. In conclusion, the proposed model successfully recognizes and counts the different types of cells present in the laboratory.

Keywords

Cell Counting, Deep Learning, Microscopic, Yolov5.

DOI: 10.9781/ijimai.2023.08.001

I. Introduction

Scientists have collected large amounts of data thanks to measurement-taking in bioengineering, tissue engineering, regenerative medicine, and biomedical research where microscopy and sample preparation techniques have been able to provide images of different phenomena of study and where the quantification of information plays an essential role for the analysis of more accurate and reliable statistics [1]–[3]. Obtaining useful and accurate information from an image quickly and easily remains a challenge in many research areas. Especially in biology and medicine, it is essential to measure cellular characteristics, such as shape and size, for statistical analysis when comparing different samples or experiments [4]. For this purpose, different methods can be used such as the use of vital dyes, the use of counting chambers, or the use of automatic cytometers. It often involves manual counting of thousands of cells with certain markers or measuring their shape and characteristics [5]. This manual process is tedious and time-consuming, which increases the workload of technicians [6]. Therefore, researchers propose automatic models such as plate counting [7], real-time quantitative PCR [8], hemocytometers [9], automatic cell counting instruments [10], and flow cytometry counting in biological systems. A clear example where the use of automatic counting tools can be beneficial is in the study of leukemia, which is a type of cancer that occurs in the human bone marrow and produces abnormal white blood cells in excess. These white blood cells can vary greatly in number and behavior compared to normal ones, which can indicate that the immune system is failing and that the patient is exposed to antigens. Therefore, white blood cell counts are a quantitative measure of disease progression [11].

It is possible to address the task of cell counting in images using state-of-the-art detection techniques such as YoloV5 [12]. These systems can be trained to adapt autonomously to the task, using data provided by researchers in their laboratories. Although there are several automatic analyzers capable of counting cells and providing

* Corresponding author.

E-mail address: sebastianlopezflorez@usal.es

statistics, these often present limitations in terms of accuracy, speed, and resolution [13]. These factors can hinder the accurate identification of cells, especially when there is overlap, which can negatively affect the quality of counting. Detection-based methods first determine the centroid locations of cells and then count them to estimate the total number of cells. Due to the success of these systems in counting and detection tasks in various areas such as agriculture, urban systems, and driving [14]–[16], it has been shown that the accuracy of these methods is strongly influenced by the accuracy of the detection results. However, in practical applications, such as fruit detection in clusters, where objects are densely concentrated and surrounded by structures that can interfere with detection, cell arrest could positively favor the results [17]. The paper [18] presents a promising approach to cell counting using the YOLOv3 detection technique. While this method has shown significant improvements over manual cell counting methods, it has some limitations that have prompted the need for further research. The performance of YOLOv3 is heavily influenced by the accuracy of the detection results, which can be hindered by overlapping cells and other interfering structures. Moreover, YOLOv3 struggles with handling of small objects and dealing with large variance in object scales. The need for a more efficient and accurate cell counting method is evident, especially in the study of diseases like leukemia where precise white blood cell counts are crucial [11].

In light of these challenges, this paper proposes a new cell counting method based on the YOLOv5 model, which offers several improvements over YOLOv3. Our proposed model aims to serve as a more accurate and efficient solution to cell counting in real-time in microscopic images, a task challenged by the low quality of visual features and the criticality of accurately locating cells for correct classification. Our initial results suggest that the proposed YOLOv5 model improves the prediction accuracy on a database of images taken by microbiology experts with a CytoSMART Exact FL microscope.

This work is an extended version of a preliminary paper presented in [19]. In this version, we have incorporated a more advanced object detection model, based on a machine-learning method that detects objects without the need for an exhaustive search. The proposed model applies to cell counting in real-time in microscopic images, which is a difficult task due to the low quality of visual features and the importance of locating the desired object for correct cell classification. We have compared our implementation with a method we had worked with previously [19]. The results indicate that the proposed yolov5 model improves prediction accuracy on the database that contains images taken by experts in microbiology with a CytoSMART Exact FL (Fluorescence) microscope that captures cells for counting. Fig 1 shows the components of the cell counting application.

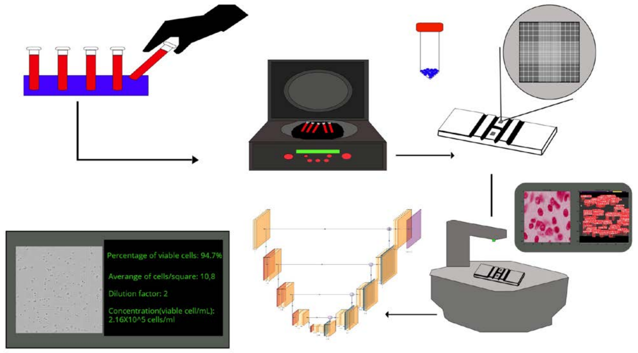

Percentage of viable cells: 94.7%
Averange of cells/square: 10,8
Dilution factor: 2
Concentration(viable cell/mL):
2.16X10^5 cells/ml

Fig. 1. The figure shows the components of cell counting using YOLOv5 and the CytoSMART microscope. The sample is mixed and placed in the hemocytometer. The sample rests, then is covered and observed under the microscope. YOLOv5 automatically identifies and counts the cells in the large squares. The software calculates the total number of cells. This completes the cell count.

After this introductory section, the remainder of this article is structured as follows: Section II provides a review of related work in the areas of image processing and deep learning. Section III describes the methods and network used in our study, including the handling of the Neubauer Cell Counting Chamber, the CytoSMART Exact FL microscope, the data set used, and the implementation of YOLOv5. Section IV covers our experimental setup and results, with a focus on training validation, materials used, model tuning, and the results obtained. The counting results are discussed in E. The article concludes with Section V, where we summarize our findings and offer some concluding thoughts.

II. Related Work

Cell counting is performed using electronic and optical technologies that analyze images [20], [21]. Previously, it was performed on a cell suspension sample by manually manipulating of the hemocytometer, flow cytometry, and chemical compounds, which was time-consuming and error-prone [22]. However, with the introduction of image analysis, cells can be identified and counted more accurately. Initial studies focus on handcrafted features and use statistical models to detect and classify cells [23], [24]. In recent years, cell counting has been achieved in an automated way thanks to the use of image processing and machine learning techniques [25], [26].

A. Image Processing

Cell counting plays a crucial role in various biomedical applications, such as cancer detection, drug discovery, and toxicity testing. However, traditional manual cell counting methods, performed by skilled workers using microscopes and counting chambers, are labor-intensive, time-consuming, and prone to human error, making standardization and result replication challenging across different samples [27], [28]. Moreover, distinguishing between cells of similar size and shape or cells that cluster together can lead to inaccuracies in cell counts [29]. To address these issues, automated cell counting techniques have been developed. One early approach involved electronic particle counting, which detected cells passing through a small aperture using impedance or light scattering. Although quick and precise, this method failed to differentiate between live and dead cells and required high cell density [30]. These techniques can be categorized into direct and indirect methods. Direct methods involve marking cells with stains or dyes and counting them based on fluorescence or absorbance. Indirect methods rely on analyzing morphological characteristics like size, shape, and texture to identify and count cells in digital images. Automated cell counting techniques can also be classified based on deep learning, machine learning, or image processing approaches. In light of the limitations and advancements in cell counting, the present work aims to propose an improved methodology by building upon the studies conducted by Payasi and Patidar [31], Acharya and Kumar [21], Clarke et al. [32], and Kaur et al. [33]. These studies have contributed valuable insights into counting tuberculosis bacilli, red blood cells, colonies, and platelets, respectively. However, each study has specific limitations related to image preprocessing, segmentation, feature extraction, and counting algorithms, which the present work seeks to address and overcome. By incorporating advancements in image processing, machine learning, and other relevant techniques, the goal is to develop a more accurate and robust automated cell counting method for enhanced biomedical applications.

B. Deep Learning

In the field of automated cell counting, two primary methodologies are employed: detection-based and regression-based methods. Detection-based methods, which aim to identify and count cell centers, are instrumental in locating individual cells and their precise positions, fitting specific lab conditions [18], [34], [35]. Conversely, regression-based techniques are more suitable for assessing cell sample density and conditions associated with cell dispersion, rather than individual or clustered cells [31], [32], [36]. In related work, Kumaar et al. [37] proposed a novel approach for brain tumor classification using a pre-trained Auxiliary Classifying Style-Based Generative Adversarial Network, demonstrating the broader applicability of machine learning techniques in the medical field.

In the context of automated cell counting and medical imaging, another study worth mentioning uses deep learning for detecting Ventricular Septal Defects in ultrasound images. Chen et al. (2021) proposed a modified YOLOv4-DenseNet algorithm for this purpose. They found the algorithm to be effective, and it outperformed other methods such as YOLOv4, YOLOv3, YOLOv3–SPP, and YOLOv3–DenseNet in terms of the mAP-50 metric. This study demonstrates the applicability of deep learning methods like YOLOv4-DenseNet in medical imaging and could provide insights for enhancing automated cell counting techniques [38].

The limitations of the aforementioned methods illustrate the challenges inherent in cell counting and analysis. These techniques, while effective in their specific applications, illustrate the need for a more versatile approach that can account for the diversity of cell types and variations in cell morphology.

Kaur et al.'s method [39], for instance, uses the circular Hough transform to count platelets in blood images. This method is effective due to the size and shape characteristics of platelets. However, when applied to cells of different sizes and shapes, its effectiveness may decrease.

The machine learning approach proposed by other researchers [22] employs the YOLO object detection and classification algorithm to identify and count three types of blood cells. This method is innovative in its use of machine learning for cell counting, but its generalizability to other cell types may be limited.

In a subsequent study, an algorithm using YOLOv3 for counting red and white blood cells was introduced [18]. This method relies on image density estimation for counting grouped red blood cells, which may lead to inaccuracies due to variations in cell grouping and distribution.

Single-stage detector methods, such as YOLO [40], are pivotal tools in cell counting due to their speed, efficiency, and accuracy. They are typically faster than two-stage detection methods, like R-CNN (Region with Convolutional Neural Networks), which is a critical advantage in healthcare applications where time can be essential, such as in disease diagnosis and treatment. These methods are capable of detecting and classifying objects (in this case, cells) in a single pass through the network, which can be more efficient in terms of computational resources than methods requiring multiple passes [41]. Although single-stage detection methods may not be as precise as some two-stage detection methods, their performance is often sufficient for many applications, including cell counting. In summary, single-stage detection methods offer a balance between speed, efficiency, and accuracy that makes them valuable for cell counting and analysis.

These methods exhibit the complex challenges associated with cell counting and analysis. They underscore the need for a method that is not only effective with a specific type of cell or under specific conditions but can also adapt to different cell types and conditions. This study aims to address these challenges by developing a more versatile and accurate approach to cell counting and analysis.

III. Methods Network

The proposed method consists of three elements: image capture using a CytoSMART Exact FL microscope with open API for cells in Neubauer plates, labeling and cell detection and counting. Due to the existence of several types of Neubauer plates, an additional database was searched to strengthen the model and then tests were performed with images under laboratory conditions.

A. Handling of the Neubauer Cell Counting Chamber

The counting chamber system involves placing a small amount of the cell suspension to be counted in the center of a special slide called a counting chamber. This slide has a known surface pattern and a fixed height. Next, the chamber is covered with a coverslip that rests on pillars that determine the volume of the suspension between the slide and the coverslip. Then, the chamber is observed under a microscope, and the cells or particles that are found within the areas marked by the pattern, are counted. Finally, the concentration of cells or particles in the suspension is calculated using the number of cells counted, the area, and the volume of the chamber. This system is mainly used in blood analysis, counting bacterial, sperm, and fungal cells [42].

The microscope was used to capture the information presented on CytoSMART Exact FL neubauer cameras. Using the 6.4 MP CMOS camera combined with 10x magnification, the CytoSMART Exact FL can view and count cells down to 4 µm in diameter [43].

B. CytoSMART Exact FL Microscope

The CytoSMART Exact FL microscope is a key tool in biological research due to its advanced, integrative features. Unlike other microscopes, it combines high-resolution imaging with cloud-based analysis and automated cell counting, offering a comprehensive solution for cellular studies. Its fluorescence capabilities allow visualization and quantification of fluorescently labeled cells, crucial for various forms of research. The cloud-based platform facilitates collaboration and remote analysis, fitting well with the modern trend of remote work. Despite its advanced features, the CytoSMART Exact FL is user-friendly, making it accessible to a wide user base. Its compact design further enhances its practicality in various lab settings. In essence, the CytoSMART Exact FL microscope, with its unique combination of features, provides convenience, efficiency, and accuracy, making it indispensable in cellular research [44].

C. Data Set

This research was based on the Sperm DetectionV4 Image Dataset [45], which consists of a total of 1024 images. Of these images, 820 were used for training with pixel-level annotations, 104 for testing, and 100 for validation. For our case study, we needed data on cells obtained through CytoSMART's Neubauer Exact FL cameras. Since object detection methods require object position data, we needed to create our own labels for the data. We used an annotation tool that exports boxes as coordinates that will be used later for training. This tool allowed us to locate the cells within a rectangle, generating a specific label for each patch. All of this was done through the LabelImg program, as shown in Fig 2. In total, this image set consists of 16 training images and 5 validation images.

D. YOLOV5

YOLOv5, the base of our proposed method, employs advanced modules such as Mosaic, Focus, BottleneckCSP, SPP, and PANet to enhance object detection performance [46]. Its architecture is composed of three key parts: a backbone network, a detection neck, and three detection heads.

The training images, denoted as I with dimensions $H \times W \times C$ (height, width, and number of channels, respectively), first undergo mosaic processing before being fed into the backbone network. This backbone network, consisting of convolutional layers, extracts features at multiple scales, transforming the input image into a set of feature maps, $F = F_1, F_2, ..., F_n$, where each F_i has size $H_i = W_i \times C_i$.

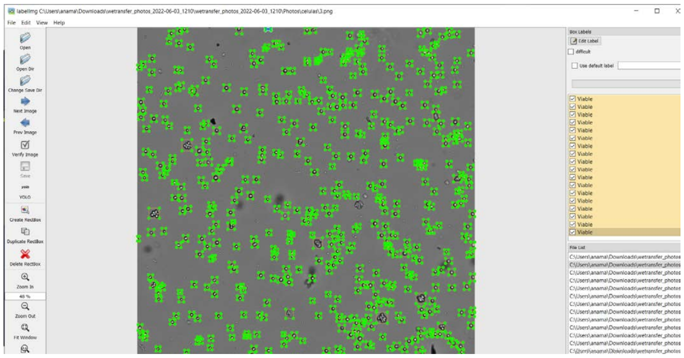


Fig. 2. The figure shows the annotation of cell images using a labeling program. The user loads an image, selects cells by clicking or drawing rectangles, the program assigns unique labels, and saves coordinate data and cell sizes. This process is repeated on multiple images to create a training data set that trains a neural network to automatically detect cells.

Following feature extraction, these maps are then processed by the detection neck, which applies a series of $F_i' = T(F_i)$, where T is the transformation operation, and F_i' is the transformed feature map.

The detection heads make final predictions for objects of varying sizes. Each detection head outputs a tensor, $D = D_1, D_2, ..., D_m$, where D_i represents a detected object and consists of the object's category c_i, confidence score s_i, and bounding box position $b_i = (x_i, y_i, w_i, h_i)$ (with xi, yi being the coordinates of the box's center, and wi, hi being the width and height of the box, respectively).

YOLOv5 utilizes the FPN-PAN structure, CSP2 structure from CSPNet, and PANet as the neck for feature aggregation, improving the detection of objects of varying scales. The model employs a new FPN structure in the feature extractor, and the PAN structure helps transfer robust localization features from lower to higher feature maps, improving low-level feature propagation and enhancing the feature fusion capability of the Neck network [47].

The loss function for YOLOv5 was changed from binary cross-entropy to focal loss. This can be explained as:

Binary cross-entropy loss:

$$Loss_{clc,obj} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p(y_i)) + (1 - y_i)\log(1 - p(y_i)) \tag{1}$$

(1) Binary cross-entropy is used to judge the difference between the predicted result of a classification model and the true value. If the predicted value $p(y_i)$ is closer to 1, then the value of the loss function should be closer to 0, that is, the smaller the difference between the predicted result and the true value, the smaller the value of the loss function.

Focal loss:

$$\begin{aligned} Loss_{fl} = -\frac{1}{N}\sum_{i=1}^{N} & \alpha_{y_i}\,(1 - p(y_i)^{\gamma})\log(p(y_i)) \\ & +(1 - \alpha)(1 - y_i)p(y_i)^{\gamma}\log(1 - p(y_i)) \end{aligned} \tag{2}$$

(2) where $Loss_{fl}$ is the Focal loss function, the α weight factor is used to regulate the balance between positive and negative samples, the γ weight factor is to regulate the weight balance between difficult samples. y_i is the true value of the tag, 1 is a positive sample and the rest is a negative sample, $p(y_i)$ is the predicted value output by the network model.

Compared to the binary cross-entropy loss, the focal loss dynamically scales the loss contribution from easy samples and focuses on hard samples. By down-weighting easy examples and emphasizing hard examples, the focal loss accelerates model convergence and improves accuracy -which is important for object detection tasks like in YOLOv5.

For transfer learning, YOLO-v5 used pre-trained weights from 70 epochs trained on the COCO dataset. The model was trained for a maximum of 20 trials with a patience level of 100, meaning that training would stop after 20 consecutive trials without improvement. The image resolution was set to 640 px by 640 px, and the batch size was 32. Model performance during training was evaluated using visual analysis of the training loss and validation curves. To test the model's performance, a set of images with correct and faulty states were processed from the camera.

IV. Experimental Setup and Results

This section presents the experiments that were carried out to evaluate the effectiveness of the proposed approach. First, the data set used in the research is described, then the performed experiments are analyzed, and the results obtained with the proposed approach are compared with other competitive approaches.

A. Data Augmentation

In the context of cell counting using the YOLOv5 object detection model, data augmentation is applied to the microscopy image data of the cells. Techniques include image translation, where images are shifted horizontally or vertically. This can help the model generalize to scenarios where the cells may not be perfectly centered in the field of view. Rotation or scaling of images can help the model learn to recognize cells in various orientations and sizes. Flipping images horizontally or vertically can assist the model in recognizing cells that can appear in different orientations within a biological sample. Adjusting the brightness and contrast of images can help the model generalize to different lighting conditions that can occur during microscopic imaging. Lastly, image cropping can create 'new' images by focusing on different parts of the original image, which can help the model learn to recognize cells even when only a part of them is visible. This is particularly useful in scenarios where cells may be partially obscured by other biological material.

B. Training Validation

The YOLOv5-based network was pre-trained using the sperm detection database, and the obtained weights were saved for future use. The appropriate number of epochs to train a new dataset of molds was determined by selecting the 205-epoch model, which took approximately 40 minutes to complete. During the model training process, the training and validation dataset was used, while an additional test dataset of 5 images was presented to independently evaluate the model performance.

C. Materials

After 50 epochs, the YOLOv5 model demonstrated good performance. However, as the epochs were increased, all losses including classification loss, box loss, and objectness loss increased, resulting in a decrease in the model's performance. The YOLOv5 model was used to detect cells under various microscope imaging conditions after creating a labeled dataset to achieve optimal cell detection. During the model training, several image resolutions were used, but an appropriate image resolution of 500×500 pixels was chosen.

After training the model, the precision, recall, and average precision (AP) of the detected objects were calculated and compared with other models [46].

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

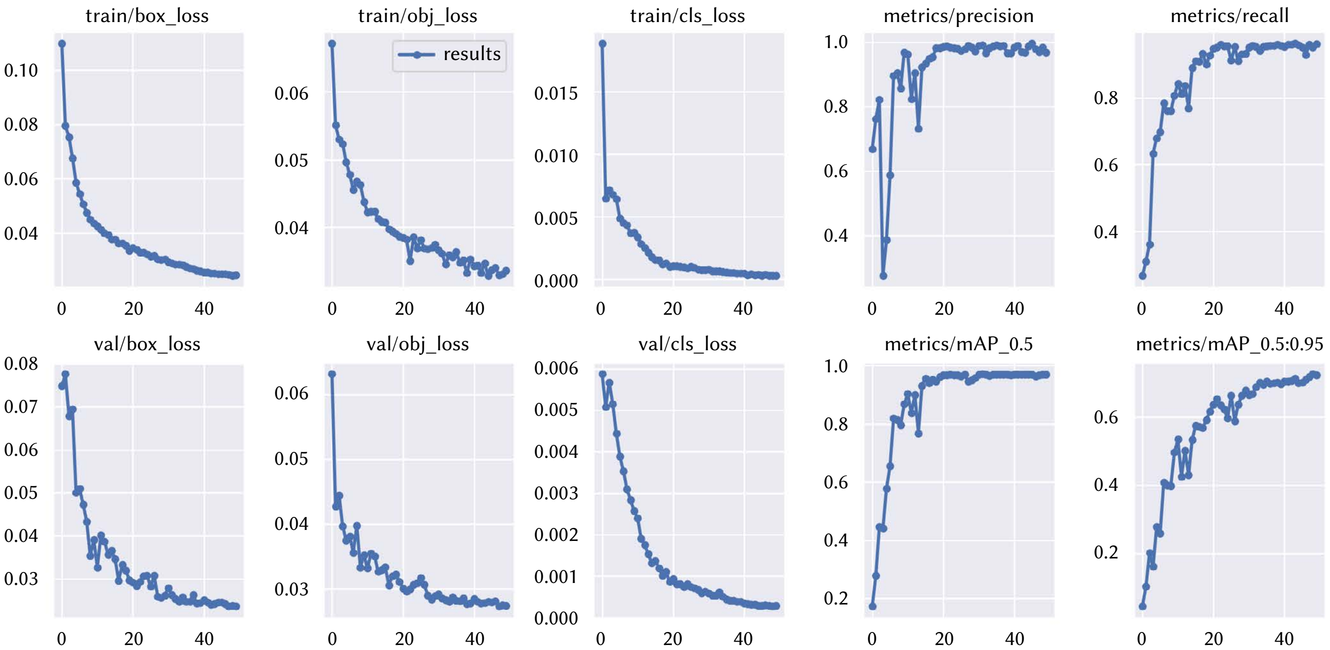


Fig. 3. YOLOv5: (top row) training, and (bottom row) validation graphs.

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{6}$$

D. Model Tuning

The study was conducted on a local machine that included a 16 GB NVidia RTX2080 GPU, 32 GB main memory, 1.9 GHz CPU and SSD hard disk. cuDNN 10 was used to run YOLOv5 on this GPU. The YOLOv5 architecture was tuned and configured to fit the sperm detectionV4 image dataset by using transfer learning. Previously pre-trained weights were used, which were trained on the sperm detectionV4 dataset. The last three YOLOv5 and convolutional layers were adjusted to match the number of classes present in the dataset.

The original pre-trained YOLOv5 model was trained on 2 classes, so we reconfigured it to a single "valid" class to address the sparseness of our dataset. To further address data sparsity and cover semantic variations, we applied several data augmentation techniques before and during YOLOv5 training. Data augmentation parameters were tuned to generate multiple images from a single image and enrich the training data. Additionally, we set the number of batches to 6 to increase model robustness and better fit GPU memory, and the number of training epochs to 50, at which point the model stabilized. Other hyperparameters were kept at default values [48].

Finally, we trained and tested YOLOv5 on our local machine using the laboratory's dataset. We trained YOLOv5 for 50 iterations, saving weights every 10 iterations. We then plotted mean average precision (mAP) vs. a number of iterations at four different saved weight points to analyze performance over training.

E. Results

Fig 3 shows how the YOLOv5 model performs as it is trained. The top row shows the results of the model using the training set, while the bottom row shows the results of the model using the validation set. It can be seen that the accuracy of the model in detecting drone objects improved significantly after 50 epochs, reaching a loss of less than 0.03. To avoid overtraining, the early stopping technique was used, which means that the training process is stopped when no noticeable improvements in performance are observed. In Fig. 3, some fluctuations in the signals can be seen, which are common during the training process and are due to divergent weights.

Table I compares the approach proposed in this research for the object detection task with other competing methods in the literature. As shown in the table, the best results are achieved with the yolov5 model. As the dataset used in this research contains small objects, such as cells, the accurate detection of these objects is a critical challenge for object detection models. In this context, the yolov5 model has proven to be an effective choice, as it achieves the best results for the cell detection task.

TABLE I. The Testing Results of Different Object Detection Algorithms. Accuracy. F1: F1-score. FPS: Frames Per Second; FPS Represents the Detection Speed of the Algorithm Under CPU Computing Conditions, Respectively

Model	Accuracy	Precis	Recall	F1Score	FPS
Yolov4	0.90	0.80	0.89	0.84	30.85
Yolov5	0.92	0.84	0.91	0.87	35.86

Despite the cell model's commendable detection rate and satisfactory loss value outcomes, there remain instances where the test set images display errors. These errors primarily arise from excessive occlusion and light interference that confound the localization and classification modules, as depicted in Fig 4.

Simultaneously, the functionality of our cell detection system is vividly illustrated. The system's efficiency and precision come to the fore through a microscopic view of a cell sample, where bounding boxes produced by our system are prominently displayed. Each of these boxes encapsulates a single cell, thereby underlining the system's adeptness in accurately identifying and isolating individual cells within the sample. This integration of the two paragraphs provides a balanced view of the system's capabilities and areas for improvement.

The application of deep learning models in cell counting has shown promising results, improving accuracy and efficiency in biological research. This article focuses on the evaluation of YOLOv5 in comparison with its predecessor, YOLOv4, for automatic cell counting using fluorescence microscopy.

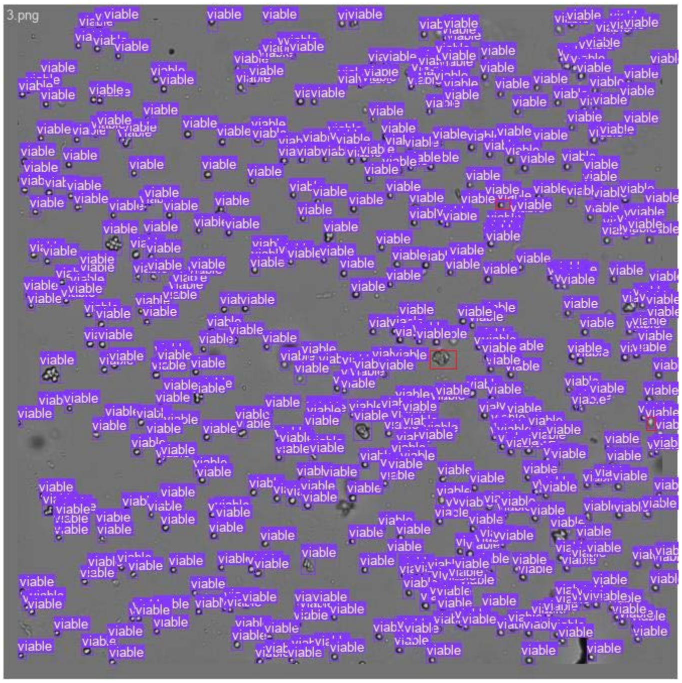


Fig. 4. Visual Representation of the YOLOv5 Model Detecting Minute Cells in a Functional System.

The choice to compare YOLOv5 with YOLOv4 is deliberate. YOLOv4 is a previous version of the You Only Look Once (YOLO) series of object detection models. Comparing YOLOv5 with YOLOv4 allows us to highlight the improvements and advancements in the latest version, demonstrating why YOLOv5 is a more suitable choice for our study.

F. Counting Result

The cell counting results from the computer vision models are summarized in Table II. The YOLOv5 model achieved a mean relative error of 1.84% on the cell counting task, significantly outperforming the U-Net model which attained an error of 39.9%. This substantial discrepancy in performance can be attributed to the superior ability of the YOLOv5 framework to handle the complexity and nuances of the cell counting environment. In particular, the YOLOv5 model can efficiently detect objects amid semantic clutter and occlusions, enabling it to generate more accurate cell counts than the U-Net approach which struggles with such challenging conditions.

TABLE II.: Microscope Cell Count Results With the Yolov5 and Semantics Technique, Where N Tests Is the Number of Tests, R Error Is the Relative Error, and A RE Error Is the Average Relative Error

Techniques	Exp	N Test	Re Error	A R Error	FPS
Yolo V5	1	201/201	0%	1.84%	28
	2	150/148	1.3%		36
	3	522/500	4.21%		30
	4	323/317	1.86%		35
U-Net [49]	1	201/124	38.30%	39.9%	28
	2	150/86	42.66%		36
	3	522/284	45.59%		30
	4	323/216	33.12%		35

In the domain of automated cell counting, various models exhibit different levels of precision and operational efficiency. This work examines and contrasts two methodologies: YOLOv5, U-Net, and OpenCV. Table II presents experimental results using these techniques, comparing their relative errors and processing speed (FPS). The evaluation includes U-Net, a deep-learning model previously used in our laboratory, and YOLOv5, the most recent model adopted for cell detection and counting.

OpenCV is a programming function library used alongside U-Net for cell counting after segmentation. Performance metrics, namely relative error and FPS, provide a comprehensive evaluation of each model's capabilities. Relative error quantifies the deviation between expected and actual cell counts, whereas FPS measures the speed of processing frames, thereby demonstrating each model's efficiency.

V. Conclusion

In this work, we demonstrate the application of a deep learning system for cell counting. The proposed YOLOv5 model, applied to CytoSMART Exact FL microscope images, enables a customized tool for the specific use case of counting different cells studied in the laboratory. Our model adopted object detection and multi-object tracking technology to achieve feasible cell detection and counting. The proposed architecture was compared with a segmentation-based method, which yielded promising results by outperforming the current method implemented in the laboratory. The introduction of semantic expert context labels improves the detection of clustered or overlapping cells. Automating cell counting could save time spent on this tedious and time-consuming task, freeing workers to focus on other important tasks and reducing costs and workload. The network used transfer learning to adapt network weights from a Sperm DetectionV4 database. To verify the effectiveness of the algorithm, a dataset of cell count use cases obtained in the laboratory was used to train and test the algorithm. Experimental results indicate that, compared to the original U-Net segmentation-based network implemented in the laboratory, the improved network achieves faster image processing, averaging 32.25 fps versus 0.95 fps per image for the previous system. The accuracy, precision, recall and F1 score for detection between YOLOv4 90%, 80%, 89% and 84%, and YOLOv5 reached 92%, 84%, 91% and 87%, respectively. Counting performance had a mean relative error of 1.84% for YOLOv5 versus 39.9% for U-Net, demonstrating considerable improvement.

As future work it is necessary to expand the database to include a larger number of events for which the current algorithm may not be prepared, this could give a better perspective if there are changes in scale or new cells are brought into the laboratory.

Acknowledgements

This work has been supported by the project "XAI -Sistemas Inteligentes Auto Explicativos creados con Módulos de Mezcla de Expertos", ID SA082P20, financed by Junta Castilla y León, Consejería de Educación, and FEDER funds.

References

[1] M. Anderson, P. Hinds, S. Hurditt, P. Miller, D. McGrowder, R. Alexander-Lindo, "The microbial content of unexpired pasteurized milk from selected supermarkets in a developing country," *Asian Pacific journal of tropical biomedicine*, vol. 1, no. 3, pp. 205–211, 2011.

[2] T. E. Gray, D. G. Thomassen, M. J. Mass, J. C. Barrett, "Quantitation of cell proliferation, colony formation, and carcinogen induced cytotoxicity of rat tracheal epithelial cells grown in culture on 3t3 feeder layers," *In Vitro*, pp. 559–570, 1983.

[3] Y. Li, G. Hetet, A.-M. Maurer, Y. Chait, D. Dhermy, J. Briere, "Spontaneous megakaryocyte colony formation in myeloproliferative disorders is not neutralizable by antibodies against il3, il6 and gm-csf," *British journal of haematology*, vol. 87, no. 3, pp. 471–476, 1994.

[4] W. Xie, J. A. Noble, A. Zisserman, "Microscopy cell counting and detection with fully convolutional regression networks," *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization*, vol. 6, no. 3, pp. 283–292, 2018.

[5] T. Falk, D. Mai, R. Bensch, Ö. Çiçek, A. Abdulkadir, Y. Marrakchi, A. Böhm, J. Deubner, Z. Jäckel, K. Seiwald, *et al.*, "U-net: deep learning for cell counting, detection, and morphometry," *Nature methods*, vol. 16, no. 1, pp. 67–70, 2019.

[6] V. Gallego Albiach, L. M. Pérez Igualada, "Estimación de la densidad celular mediante el uso de cámaras de recuento," 2021.

[7] C. Wilson, R. Lukowicz, S. Merchant, H. Valquier- Flynn, J. Caballero, J. Sandoval, M. Okuom, C. Huber, T. D. Brooks, E. Wilson, *et al.*, "Quantitative and qualitative assessment methods for biofilm growth: a mini-review," *Research & reviews. Journal of engineering and technology*, vol. 6, no. 4, 2017.

[8] G. M. Knight, E. Dyakova, S. Mookerjee, F. Davies, E. T. Brannigan, J. A. Otter, A. H. Holmes, "Fast and expensive (pcr) or cheap and slow (culture)? a mathematical modelling study to explore screening for carbapenem resistance in uk hospitals," *BMC medicine*, vol. 16, no. 1, pp. 1–11, 2018.

[9] B. Song, B. Zhuge, H. Fang, J. Zhuge, "Analysis of the chromosome ploidy of candida glycerinogenes," *Wei Sheng wu xue bao= Acta Microbiologica Sinica*, vol. 51, no. 3, pp. 326–331, 2011.

[10] S. I. Kim, H. J. Kim, H.-J. Lee, K. Lee, D. Hong, H. Lim, K. Cho, N. Jung, Y. W. Yi, "Application of a non- hazardous vital dye for cell counting with automated cell counters," *Analytical biochemistry*, vol. 492, pp. 8–12, 2016.

[11] D. Wang, M. Hwang, W.-C. Jiang, K. Ding, H. C. Chang, K.-S. Hwang, "A deep learning method for counting white blood cells in bone marrow images," *BMC bioinformatics*, vol. 22, no. 5, pp. 1–14, 2021.

[12] X. Zhu, S. Lyu, X. Wang, Q. Zhao, "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 2778–2788.

[13] S.-J. Lee, P.-Y. Chen, J.-W. Lin, "Complete blood cell detection and counting based on deep neural networks," *Applied Sciences*, vol. 12, no. 16, p. 8140, 2022.

[14] Y. Egi, M. Hajyzadeh, E. Eyceyurt, "Drone-computer communication based tomato generative organ counting model using yolo v5 and deep-sort," *Agriculture*, vol. 12, no. 9, p. 1290, 2022.

[15] S. Xiang, S. Wang, M. Xu, W. Wang, W. Liu, "Yolo pod: a fast and accurate multi-task model for dense soybean pod counting," *Plant Methods*, vol. 19, no. 1, p. 8, 2023.

[16] R. K. Purwar, S. Verma, "Analytical study of yolo and its various versions in crowd counting," in *Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2021*, Springer, 2022, pp. 975–989.

[17] S. He, K. T. Minn, L. Solnica-Krezel, M. A. Anastasio, H. Li, "Deeply-supervised density regression for automatic cell counting in microscopy images," *Medical Image Analysis*, vol. 68, p. 101892, 2021.

[18] D. Zhang, P. Zhang, L. Wang, "Cell counting algorithm based on yolov3 and image density estimation," in *2019 IEEE 4th international conference on signal and image processing (ICSIP)*, 2019, pp. 920–924, IEEE.

[19] S. L. Flórez, A. González-Briones, G. Hernández, F. de la Prieta, "Automated counting via multicolumn network and cytosmart exact fl microscope," in *Ambient Intelligence—Software and Applications—13th International Symposium on Ambient Intelligence*, 2023, pp. 207–218, Springer.

[20] S. Chakraborty, C. Das, K. Ghoshal, M. Bhattacharyya, A. Karmakar, S. Chattopadhyay, "Low frequency impedimetric cell counting: analytical modeling and measurements," *IRBM*, vol. 41, no. 1, pp. 23–30, 2020.

[21] A. Aijaz, D. Trawinski, S. McKirgan, B. Parekkadan, "Non-invasive cell counting of adherent, suspended and encapsulated mammalian cells using optical density," *BioTechniques*, vol. 68, no. 1, pp. 35–40, 2020.

[22] M. M. Alam, M. T. Islam, "Machine learning approach of automatic identification and counting of blood cells," *Healthcare technology letters*, vol. 6, no. 4, pp. 103– 108, 2019.

[23] J. Matas, O. Chum, M. Urban, T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[24] C. Arteta, V. Lempitsky, J. A. Noble, A. Zisserman, "Learning to detect cells using non-overlapping extremal regions," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012: 15th International Conference, Nice, France, October 1-5, 2012, Proceedings, Part I 15*, 2012, pp. 348–356, Springer.

[25] V. Acharya, P. Kumar, "Identification and red blood cell automated counting from blood smear images using computer-aided system," *Medical & biological engineering & computing*, vol. 56, pp. 483–489, 2018.

[26] M. L. Clarke, R. L. Burton, A. N. Hill, M. Litorja, M. H. Nahm, J. Hwang, "Low-cost, high-throughput, automated counting of bacterial colonies," *Cytometry Part A*, vol. 77, no. 8, pp. 790–797, 2010.

[27] A. Vembadi, A. Menachery, M. A. Qasaimeh, "Cell cytometry: Review and perspective on biotechnological advances," *Frontiers in bioengineering and biotechnology*, vol. 7, p. 147, 2019.

[28] M. M. Alam, M. T. Islam, "Machine learning approach of automatic identification and counting of blood cells," *Healthcare technology letters*, vol. 6, no. 4, pp. 103– 108, 2019.

[29] P. J. Schüffler, T. J. Fuchs, C. S. Ong, P. J. Wild, N. J. Rupp, J. M. Buhmann, "Tmarker: A free software toolkit for histopathological cell counting and staining estimation," *Journal of pathology informatics*, vol. 4, no. 2, p. 2, 2013.

[30] R. J. Santen, "Automated estimation of diploid and tetraploid nuclei with an electronic particle counter," *Experimental Cell Research*, vol. 40, no. 2, pp. 413–420, 1965, doi: 10.1016/0014-4827(65)90274-0.

[31] Y. Payasi, S. Patidar, "Diagnosis and counting of tuberculosis bacilli using digital image processing," in *2017 international conference on information, communication, instrumentation and control (ICICIC)*, 2017, pp. 1–5, IEEE.

[32] M. L. Clarke, R. L. Burton, A. N. Hill, M. Litorja, M. H. Nahm, J. Hwang, "Low-cost, high-throughput, automated counting of bacterial colonies," *Cytometry Part A*, vol. 77, no. 8, pp. 790–797, 2010.

[33] P. Kaur, V. Sharma, N. Garg, "Platelet count using image processing," in *2016 3rd International conference on computing for sustainable global development (INDIACom)*, 2016, pp. 2574–2577, IEEE.

[34] V. Acharya, P. Kumar, "Identification and red blood cell automated counting from blood smear images using computer-aided system," *Medical & biological engineering & computing*, vol. 56, pp. 483–489, 2018.

[35] C. Arteta, V. Lempitsky, J. A. Noble, A. Zisserman, "Detecting overlapping instances in microscopy images using extremal region trees," *Medical image analysis*, vol. 27, pp. 3–16, 2016.

[36] W. Xie, J. A. Noble, A. Zisserman, "Microscopy cell counting and detection with fully convolutional regression networks," *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization*, vol. 6, no. 3, pp. 283–292, 2018.

[37] M. A. Kumaar, D. Samiayya, V. Rajinikanth, D. Raj Vincent PM, S. Kadry, "Brain tumor classification using a pre-trained auxiliary classifying style-based generative adversarial network," 2023.

[38] S.-H. Chen, C.-W. Wang, I. Tai, K.-P. Weng, Y.-H. Chen, K.-S. Hsieh, *et al.*, "Modified yolov4-densenet algorithm for detection of ventricular septal defects in ultrasound images," 2021.

[39] P. Kaur, V. Sharma, N. Garg, "Platelet count using image processing," in *2016 3rd International conference on computing for sustainable global development (INDIACom)*, 2016, pp. 2574–2577, IEEE.

[40] M. Yuzkat, H. O. Ilhan, N. Aydin, "Detection of sperm cells by single-stage and two-stage deep object detectors," *Biomedical Signal Processing and Control*, vol. 83, p. 104630, 2023.

[41] W. Han, L. Cao, S. Xu, "A method of the coverage ratio of street trees based on deep learning.," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 7, no. 5, 2022.

[42] J. G. A. Barbedo, "Automatic object counting in neubauer chambers," *Scientific. net*, 2013, doi: 10.14209/sbrt.2013.224.

[43] M. J. Sanderson, I. Smith, I. Parker, M. D. Bootman, "Fluorescence microscopy," *Cold Spring Harb. Protoc.*, vol. 2014, p. db.top071795, Oct. 2014.

[44] X. Xu, Y. Feng, C. Han, Z. Yao, Y. Liu, C. Luo, J. Sheng, "Autophagic response of intestinal epithelial cells exposed to polystyrene nanoplastics," *Environmental Toxicology*, vol. 38, no. 1, pp. 205–215, 2023.

[45] king mongkuts university of technology thonburi, "sperm detectionv4 dataset." https://universe.roboflow.com/king-mongkuts-university-of-technology-thonburi-ybmh7/sperm-detectionv4, feb 2023. [Online]. Available: https://universe.roboflow.com/king-mongkuts-university-of-technology-thonburi-ybmh7/sperm-detectionv4, visited on 2023-03-07.

[46] Z. Wang, L. Jin, S. Wang, H. Xu, “Apple stem/calyx real-time recognition using yolo-v5 algorithm for fruit automatic loading system,” *Postharvest Biology and Technology*, vol. 185, p. 111808, 2022.

[47] X. Dong, S. Yan, C. Duan, “A lightweight vehicles detection network model based on yolov5,” *Engineering Applications of Artificial Intelligence*, vol. 113, p. 104914, 2022.

[48] N. Al-Qubaydhi, A. Alenezi, T. Alanazi, A. Senyor, N. Alanezi, B. Alotaibi, M. Alotaibi, A. Razaque, A. A. Abdelhamid, A. Alotaibi, “Detection of unauthorized unmanned aerial vehicles using yolov5 and transfer learning,” *Electronics*, vol. 11, no. 17, p. 2669, 2022.

[49] S. L. Flórez, A. González-Briones, G. Hernández, F. de la Prieta, “Automated counting via multicolumn network and cytosmart exact fl microscope,” in *International Symposium on Ambient Intelligence*, 2022, pp. 207–218, Springer.

Sebastian Lopez Florez

Mr.Sebastian Lopez Florez received his Mechatronics Engineering and the MSc degree in Electrical Engineering. Both degrees were received from Universidad Tecnológica de Pereira (UTP). He is currently working as candidato a doctor en la Universidad de Salamanca in a program EU Horizon 2020 Marie Skłodowska-Curie European Training Network on Internet of Things. His current research interests include developing flexible, interpretable, and scalable machine learning models, often involving deep learning, Gaussian processes, and kernel learning.

Alfonso González-Briones

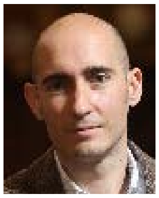
Alfonso González-Briones earned a Ph.D. in Computer Engineering in 2018 at the University of Salamanca; his thesis obtained the second place in the 1st SENSORS+CIRTI Award for the best national thesis in Smart Cities (CAEPIA 2018). At the same University, he had also obtained a Bachelor of Technical Engineering in Computer Engineering (2012), a Bachelor’s Degree in Computer Engineering (2013), and a Master’s Degree in Intelligent Systems (2014). Since 2014, Alfonso González Briones has worked in different public research centres, such as the BISITE Research Group, and at Complutense University of Madrid as a “Juan De La Cierva” Postdoc. Currently, Alfonso González Briones is Associate Professor at the University of Salamanca in the Department of Computer Science and Automation.

Guillermo Hernández González

Ph.D. in “Fundamental Physics and Mathematics” and a “Master in Intelligent Systems” from the University of Salamanca. He also holds other degrees in the fields of Physics, Computer Science and Mathematics. His current research focuses on the lines of machine learning, deep learning, natural language processing and visual analytics. He currently enjoys a postdoctoral contract at the University of Salamanca on self-explanatory intelligent systems.Additionally, he has worked on the applications of laser acceleration systems to biological tissue irradiation, where he has made several contributions to fields such as radiation-matter interaction and characterization of ultrashort pulse-generated emissions. He has also worked in the field of radioprotection and shielding design.

Carlos Ramos

The director of Gecad (the Knowledge Engineering and Decision Support Research Centre) and coordinator professor at the Polytechnic of Porto’s Institute of Engineering. His main areas of interest are ambient intelligence, knowledge-based systems, decision support systems, multiagent systems, and planning. He received his PhD in electrical and computer engineering from the University of Porto. He’s a member of the IEEE. Contact him at ISEP, Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto, Portugal; csr@isep.ipp.pt.



Fernando De la Prieta Pintado

Associate Professor at the University of Salamanca Department of Computer Science and Automation,University of Salamanca. Over the past years, he has followed a clearly defined line of research, focusing on the integration of multi-agent organisations, machine learning and advanced architectures in different fields. He has worked on more than 90 research projects. In addition, he has participated in more than 30 research contracts, in some of them as the principal investigator. As a result of his work, around 40 intellectual properties have been registered. He has done several stays abroad (pre- and post-doctoral) in Portugal, Japan and South Korea. He has also taken an active part in the organisation of international conferences, some of them included in the CORE ranking: IEEE-GLOBECOM (core B), ICCBR (Core B), CEDI, PAAMS (core C), ACM-SAC (core B), IEEE-FUSION (core C), and others.

Violence Detection in Audio: Evaluating the Effectiveness of Deep Learning Models and Data Augmentation

Dalila Durães*, Bruno Veloso, Paulo Novais*

ALGORITMI Centre/LASI, University of Minho, Guimarães (Portugal)

Received 4 March 2023 | Accepted 22 August 2023 | Published 29 August 2023

Abstract

Human nature is inherently intertwined with violence, impacting the lives of numerous individuals. Various forms of violence pervade our society, with physical violence being the most prevalent in our daily lives. The study of human actions has gained significant attention in recent years, with audio (captured by microphones) and video (captured by cameras) being the primary means to record instances of violence. While video requires substantial processing capacity and hardware-software performance, audio presents itself as a viable alternative, offering several advantages beyond these technical considerations. Therefore, it is crucial to represent audio data in a manner conducive to accurate classification. In the context of violence in a car, specific datasets dedicated to this domain are not readily available. As a result, we had to create a custom dataset tailored to this particular scenario. The purpose of curating this dataset was to assess whether it could enhance the detection of violence in car-related situations. Due to the imbalanced nature of the dataset, data augmentation techniques were implemented. Existing literature reveals that Deep Learning (DL) algorithms can effectively classify audio, with a commonly used approach involving the conversion of audio into a mel spectrogram image. Based on the results obtained for that dataset, the EfficientNetB1 neural network demonstrated the highest accuracy (95.06%) in detecting violence in audios, closely followed by EfficientNetB0 (94.19%). Conversely, MobileNetV2 proved to be less capable in classifying instances of violence.

Keywords

Audio, Deep Learning, Human Action Recognition, Machine Learning, Transfer Learning, Violence Detection in a car.

DOI: 10.9781/ijimai.2023.08.007

I. Introduction

As stated by Koritsas in 2009 [1], violence manifests in both verbal and physical aspects. Verbal aggression entails employing disrespectful speech, shouting, or shrieking with the purpose of causing offense or inducing fear. Physical aggression involves physically assaulting or trying to assault others, encompassing actions like striking, slapping, kicking, or utilizing a weapon or any object with the intention of inflicting bodily harm. As outlined in a study [2], almost half (48%) of the individuals who fell victim to interpersonal violence in South Korea in 2015 were fatally injured by sharp instruments like knives, whereas such fatalities attributed to sharp objects were approximately 25%. In the year 2020, the Portuguese Association for Victim Support (APAV) reported a total of 66,408 cases, with 31% attributed to "crimes and other forms of violence." Among these cases, 94% involved acts of violence against individuals [3]. The identification and acknowledgment of violence have been focal points of research interest, particularly within surveillance. The primary aim of detecting and recognizing violence revolves around achieving automated and real-time capabilities, enabling timely assistance to victims [4]. It is crucial to identify and prevent such actions before they escalate into catastrophic situations.

Modern society is placing increasing emphasis on automated surveillance as it helps manage an overwhelming amount of data, including attention bias, and ensures the privacy of those being surveyed [5], [6]. Well-designed surveillance software can process multiple sets of sensor data over an extended period of time without risking disengagement. On the other hand, an extra safeguard for data privacy is a properly auditable system that anonymises or deletes data in cases where violence is not detected.

Most violence recognition methods primarily rely on video detection, which necessitates high-performance hardware and software for recording [7]. An alternative technique for violence detection involves using audio, which can be effectively recognized and classified using deep learning algorithms [8]. Audio signals can be effortlessly captured by microphones, which possess strong capabilities to record human behavior and emotions. Therefore, it is crucial to have a robust audio representation that complements and validates the video's audio quality [9], [10].

Another consideration is that violence detection is often associated with crowd violence detection [11]–[14]. However, in recent times, there has been a notable surge in interest surrounding audio-based violence detection, owing to its capacity to identify and prevent violent incidents and also by the increase of car sharing. Particularly,

* Corresponding author.

E-mail addresses: dad@di.uminho.pt (D. Durães), pjon@di.uminho.pt (P. Novais).

researchers have directed their attention to the detection of violence within vehicles using audio-based methods [5], [7], [15].

Despite the promising prospects of audio-based violence detection inside vehicles, its effectiveness relies on various factors, such as the type of microphone employed, background noise levels, and microphone placement within the car [7], [16], [17]. Nonetheless, leveraging audio-based violence detection in vehicles holds potential to enhance the safety of both passengers and drivers.

According to the search results provided, various models have been employed for audio-based violence detection inside vehicles. These models include the ResNet model utilizing the Mel-spectrogram methodology for audio signals [10], [18], CNN-based Audio Event Recognition [15], ensemble deep learning, and multimodal approaches [19], as well as the application of machine learning (ML) models for detecting violence in video streams [20]. The studies indicate that deep learning techniques, such as artificial neural networks and convolutional neural networks, have demonstrated notable enhancements in the accuracy of audio event classification when compared to traditional feature-based classification methods.

Deep learning [21], a novel approach to data modeling that has gained significant traction in recent years, has led to the development of innovative structures and learning algorithms. These advancements have enabled breakthroughs in areas such as recognition [22], object recognition, and machine translation [23], [24]. In the realm of audio-related tasks, deep learning models have played a pivotal role in enhancing accuracy and robustness across diverse categories. As a result, deep learning has become a fundamental area of research in various fields of knowledge [25].

Notwithstanding the extensive research conducted thus far, the realm of identifying violence within the confines of a vehicle remains severely limited in terms of available studies. This scarcity of research is attributed to the distinctive attributes of the car's interior, which pose challenges to the effectiveness of existing models in yielding favorable outcomes [5]. As audio requires minimal storage, our intention is to carry out a study focused on detecting violence within a car using audio.

A. Main Contributions

We utilized a custom dataset designed specifically for detecting violence within a car environment using audio data. It is worth noting that this paper is an extension of the previously published work [26], with the primary focus being on violence detection within a car.

The main objective of this paper is to present the outcomes of our experiments conducted using in-car audio data and deep learning frameworks for the purpose of violence identification. The dataset used for training and validation serves as the foundation for the results presented in this study. Due to the relatively small size of the dataset, data augmentation techniques were applied to augment its volume.

The research questions to be addressed are as follows: RQ1) Can violence inside a car be effectively detected using audio data and deep learning models? RQ2) Can the use of data augmentation enhance the accuracy of violence detection results? To limit the scope of the study, incidents will be classified solely as either violent or non-violent, without considering the specific type of human action or the nature of the violence involved.

B. Organization

The organization of this document is as follows: Section II, Background, discusses the current state of the field, while Section III, Methods, outlines the Mel Spectrogram concepts, public dataset, In car dataset, pre-processing techniques, algorithms, and training procedures employed. Section IV, Results and Discussion, presents the obtained outcomes and corresponding discussions. Lastly, Section V, Conclusion, offers the final conclusions drawn from the study.

II. Background

Different methodologies adopted in some previously conducted studies on the use of audio in violence detection were explored.

A. Models

The detection of violence inside a car using audio-based methods has garnered significant interest as it holds the promise of enhancing road safety by preventing violent incidents and aiding in criminal investigations. Over time, research in this domain has resulted in the advancement of sophisticated algorithms and techniques that significantly improve the accuracy of identifying violent activities within vehicles.

Audio violence detection offers several advantages over video approaches, particularly in terms of bandwidth, storage, and computing requirements, which are significantly lower [9]. While audio sensors have their limitations, they are relatively minor compared to video cameras. For instance, microphones can have an omnidirectional capability, providing a spherical field of view, unlike video cameras with limited angular views. Additionally, audio event acquisitions can outperform video acquisitions due to the longer wavelength of audio, allowing for acoustic wave reflections when encountering obstacles in the direct path. Moreover, audio processing is not affected by issues like lighting and temperature, unlike video processing [9]. The audio approach also captures a wealth of information that visual data alone cannot represent, including screams, explosions, abusive language, and emotional cues conveyed through sound passages. Despite these advantages, there are still limited applications for violence detection using audio-based methods.

Souto, Mello, and Furtado [27] conducted research on domestic violence and acoustic scene classification using machine learning. The parameters employed for feature extraction and processing in both short and medium terms included MFCC (Mel Frequency Cepstral Coefficients), Energy, and ZCR (Zero Crossing Rate). For classification, they utilized the SVM (Support Vector Machine) technique. The resulting models, post-training, included the MFCC-SVM classifier, the Energy-SVM classifier, and the ZCR-SVM classifier.

In their previous work, Purwins, Virtanen, Schluter, Chang, and Sainath [28] explored audio signal processing methods like Gaussian mixture models, hidden Markov models, and non-negative matrix factorization. However, they found that these traditional methods were often outperformed by deep learning models when sufficient data was available. They applied various techniques such as categorization, audio features, models, data, and evaluation, and conducted cross-domain comparisons with speech, music, and environmental sounds. Additionally, for audio synthesis and transformation, they employed source separation, speech enhancement, and audio generation methods.

Rouas [29], based on public transport vehicles, studied the detection of audio events. For this purpose he created an automatic audio segmentation, which divides an audio signal into several consecutive, almost stationary zones. The developed algorithm detected activity, i.e., ignored the quiet and low noise zones, focusing exclusively on the high noise zones. In this work the SVM model was used.

Crocco [9] conducted a systematic review of surveillance based on the audio signal. In this review, several approaches are presented, namely: i) background subtraction by monomodal analysis; ii) background subtraction by multimodal analysis; iii) audio event classification; iv) source localisation and tracking, especially audio source localisation; v) audiovisual source localisation; and vi) audio source tracking and audiovisual source tracking.

Gavira [30] has presented a device designed to accurately perform the recognition task in urban areas with high noise. The audio was recorded in real urban environments using a current microphone. The strategy was to train a classifier based on temporal and frequency data analysis, and deep convolutional neural networks were used to develop the work.

Hossain [31] proposes a system for emotion recognition through audiovisuals, using two deep networks to extract features and join the features. In addition, it uses Big Data technology to train the emotion network and separate the information based on gender. The proposed system will also use a CNN network for audio signals and a three-dimensional CNN for video signals.

Another study [32] delves into the intricate task of Motivic pattern classification in music audio recordings, with a particular focus on a cappella flamenco cantes. To tackle this, the paper proposes the application of Convolutional Neural Networks (CNN) architectures for intra-style classification of flamenco cantes, utilizing small motivic patterns. The suggested architecture capitalizes on the advantages of residual CNN for feature extraction and incorporates a bidirectional LSTM layer to handle the sequential nature of musical audio data. Sequential pattern mining and contour simplification techniques are employed to extract relevant motifs from the audio recordings, and Mel-spectrograms of these motifs serve as inputs for the various architectures tested. The research investigates the practicality of motivic patterns for automatically classifying music recordings and explores the influence of audio length and corpus size on the overall classification accuracy.

B. Data Augmentation

Related to data augmentation techniques, a study [33] focuses on enhancing the accuracy of animal audio classification through various data augmentation techniques. These techniques involve manipulating the existing audio data to create additional samples, thereby increasing the diversity and size of the dataset. The study investigates different augmentation methods, their impact on model performance, and their ability to mitigate challenges such as limited labeled data. By implementing these augmentation strategies, the paper aims to enhance the robustness and effectiveness of animal audio classification models, ultimately improving their ability to accurately identify and classify animal sounds.

Another work [34] presents a methodology for effectively classifying environmental sounds using a deep convolutional neural network (CNN) that incorporates regularization techniques and data augmentation. The study emphasizes the challenges of environmental sound classification, including limited labeled data and diverse acoustic variations. To address these challenges, the proposed approach involves augmenting the dataset through various techniques and integrating regularization methods into the CNN architecture. The experimental results demonstrate that the combination of data augmentation and regularization enhances the model's ability to accurately classify environmental sounds, making it more robust to variations in acoustic conditions and contributing to improved classification performance.

Also another study [35] introduces a novel technique for augmenting audio data using an evolutionary-based generative approach. The method involves employing evolutionary algorithms to generate new audio samples that are structurally similar to the existing data while introducing variations. By iteratively refining these generated samples, the approach aims to create diverse and realistic audio data that can expand the training dataset for machine learning models. The paper highlights the benefits of this approach in improving the performance of audio-based tasks such as classification and recognition, demonstrating its effectiveness in enhancing model generalization and accuracy through the incorporation of synthetically generated but plausible audio samples.

Finally, a last study [36] presents a method for automating the selection of effective data augmentation techniques to enhance object detection models. It addresses the challenge of selecting appropriate augmentation strategies from a large set of possibilities by utilizing a reinforcement learning framework. The approach involves training a policy network that learns to select augmentation operations based on their impact on the model's performance. This policy network is optimized through reinforcement learning techniques, resulting in a strategy for augmenting the training data that improves the object detection model's accuracy. The paper demonstrates the effectiveness of the approach through experiments, showing that learned data augmentation strategies can lead to significant performance gains in object detection tasks.

The background discussed in this section highlights the progress achieved in the development of methods for identifying violence and the latest enhancements in data augmentation techniques. However, when we narrow our focus to the particular scenario of detecting violence using audio within a vehicle, the existing models are not well-suited, and there is a lack of datasets recorded in such settings. Therefore, our study aims to enhance the effectiveness of violence detection within cars by utilizing audio inputs and a newly captured in-car dataset. Additionally, we emphasize the significance of employing data augmentation techniques to improve the results in this context.

III. Methods

A. Mel Spectrogram

Audio can be converted into an interpretable format by representing it as visual images. The key concept involves transforming the audio signal into visual images, which can then be utilized to extract features either manually or directly fed into a Deep Learning classifier. There exist classifiers that can learn and extract features from these audio-generated images [37].

There are some methods tha can be used to create this images (spectrograms), that represent the audio, and some are: *Short-Time Fourier Transform*, *Chromagram*, *Mel-Spectrogram* [7]. According to the literature by Choi, Fazekas, Cho, and Sandler [38]; Gaviria et al. [30]; Hossain and Muhammad [39]; Purwins et al. [40], each method for audio representation comes with its own set of advantages and disadvantages. Nonetheless, the Mel-Spectrogram method stands out as the most widely utilized approach. Therefore, we have chosen to employ the Mel-Spectrogram method to represent audio in order to test our model.

A mel-spectrogram is a type of spectrogram, which visualizes the frequency content of an audio signal over time. However, instead of using a linear scale for the frequency axis, the mel-spectrogram uses the mel scale. The mel scale is a perceptual scale that is designed to better align with how humans hear and perceive sound [30]. The mel scale was introduced in the 1930s in order to account for the fact that humans do not perceive changes in frequency linearly - that is, changes in pitch at lower frequencies are more noticeable than at higher frequencies. The mel scale is based on this perceptual phenomenon, and is designed so that equal distances on the scale correspond to equal perceived differences in pitch. In practical terms, the mel scale is used to create a filterbank that is applied to the Fourier transform of an audio signal to map it onto the mel scale [40].

So, a mel spectrogram displays the time-frequency distribution of audio, with the frequency axis based on the mel-frequency scale. The process of converting to a mel spectrogram involves computing the Short-Time Fourier Transform (STFT) of the audio signal. This STFT computation transforms the audio from the time domain to the frequency domain. Once in the frequency domain, the y-axis is scaled using a mel-scale [41].

The mel spectrogram displays the successive frequencies (y-axis) over time (x-axis) as well as the different amplitudes (represented by colors and measured in decibels) for each moment (Fig. 1).

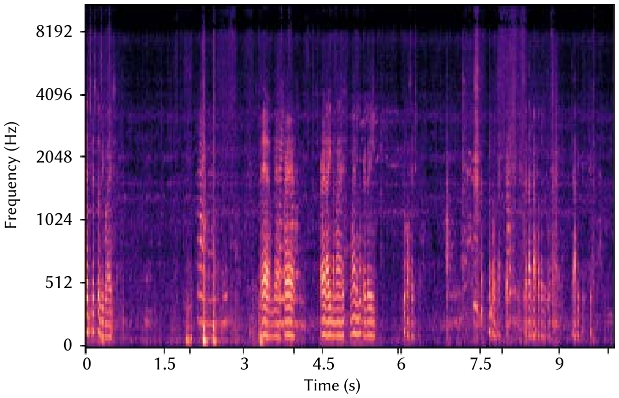


Fig. 1. Representation of a mel spectrogram.

B. Public Datasets

One of the key points in ML is selecting a dataset that has the necessary restrictions of what is intended to be classified. For the problem of this paper, the dataset should have audio entries where each entry would fall into violence or non violence. With this restrictions, we found some datasets worth mentioning.

The XD-Violence dataset is comprised of 4754 videos, with varying degrees of audio availability. It is divided into two categories: violence (2405 videos) and non-violence (2349 videos), totaling 217 hours of footage. The videos that depict violence can be further categorized into six types, including abuse, explosions, car accidents, riots, fights, and shootings. Each video of violence can have 1 to 3 labels, reflecting the significance of each event depicted. The videos come from a variety of sources, including movies, cartoons, video games, news, sports, etc. The XD-Violence dataset is separated into two parts, training and testing. The training section has 3954 videos, while the testing section has 800 videos. In both the training and testing datasets, the six types of violence are present at different points in time within the videos [42] [43].

Nanyang Technological University CCTV-Fights dataset contains 1000 videos obtained from YouTube, some without audio, that display various actions such as pushing, kicking, fighting, etc. It is separated into two categories: CCTV (280 videos captured by surveillance cameras) and NON-CCTV (720 videos captured by dash-cams, cell phones, drones, and helicopters). The CCTV (camera stands for Closed-Circuit Television camera) videos range from five seconds to 12 minutes (average of two minutes), totaling 8.54 hours, while the NON-CCTV videos range from three seconds to seven minutes (average of 45 seconds), adding up to 9.13 hours of footage [44] [45].

Violent Scenes Detection (VSD2014) is widely used when the problem is to detect violence through video or audio. It has two types of videos: clips from Hollywood movies and clips taken from YouTube. The dataset is divided into three groups: "Hollywood: Development", "Hollywood: Test" and "YouTube: Generalization". In terms of the Hollywood group, they selected some movies, and it can go from movies with some violence ("Saving Private Ryan", with 34% frames with violence) to movies without violence ("Legally Blond", with 0% frames with violence). The "Hollywood" group has a total of 63 hours and 55 minutes of movie time (31 movies), while clips from YouTube has a total of two hours and 37 minutes (86 clips) and each clip can last from six seconds to six minutes. The features offered by this dataset are separated into audio and visual features, to make it easier for those without much experience in classification to have a starting point. To complete the dataset, annotations are included for all the content. The annotations identify the start and end frames of each violent segment and are binary in nature. There are seven visual concepts and three audio ones. The visual elements include: fights, blood, fire, knives, car pursuits, and disturbing/bloody images, which may also provide information about the level of intensity. The audio elements include: shots, screams, and explosions. It should be noted that the visual elements provide the start and end of each segment, expressed in terms of frames. Meanwhile, the audio elements are described in terms of seconds for the start and end of each occurrence [46].

The Real Life Violence Situations (RLVS) dataset contains real-world violent scenarios used for research in fields like computer vision. The purpose of the RLVS dataset is to supply a varied and accurate set of violent situations for the purpose of training and evaluating algorithms and systems with the aim of detecting, preventing, and responding to acts of violence. The dataset is comprised of 2000 clips, half of which depict violence and the other half do not. Some of the clips have been manually captured. In an effort to eliminate redundancy of individuals and surroundings, additional videos were taken from the YouTube platform. The lengthy clips have been broken down into shorter ones, ranging from three to seven seconds, with an average duration of five seconds. All of these clips are of high resolution and some of them have no sound. The violent clips depict scenes from places like prisons, schools, streets, etc. The non-violent clips feature individuals participating in activities like playing walking, eating, sports, etc. This dataset includes a wide variety of race, gender and age [47].

A brief summary can be seen on the Table I.

However, despite the existence of several audio-based datasets, none have met the specified constraints for this work. So a group of researchers made their own dataset.

C. In Car Dataset

In order to evaluate the implemented models, a dataset was necessary, but no existing dataset met the specific requirements. Consequently, a team of researchers decided to create their own dataset, capturing video recordings of both violent and non-violent scenarios inside a car, involving real people, and all recorded during the pandemic. The dataset consists of videos, each with accompanying audio, representing 20 distinct scenarios. Among these scenarios, 12 involve violence, including push and punch incidents, different fight scenarios, discussions with physical altercations, sexual harassment situations, and robberies using weapons like knives or guns. One scene depicts one person forcibly looking at another's phone. On the other hand, the remaining 8 scenarios are non-violent, featuring instances such as people hugging, taking photos, fixing hair, sleeping,

TABLE I. Overall Analysis of the Datasets

Dataset	Number of videos	Duration (hours)	Sources	Audio
XD- Violence	4754	217	Movies, cartoons, videogames, news, sport, etc.	Yes
NTU CCTV-Fights	1000	18	Surveillance cameras and mobiles	Yes*
VSD2014	31 Movies + 86 Clips	64+3	Hollywood movies and clips from YouTube	Yes
RLVS	2000	-	Manually recorded and clips from YouTube	Yes*

* Some videos lack sound or only have background music.

sneezing, reading a book, yawning, listening to music, answering calls, coughing, using a notebook, and writing, along with using alcohol gel. Each scenario was recorded with 16 different pairs of actors, and certain scenes include the use of various objects. For each pair, each scenario was recorded twice, P1 is one person and P2 is the other one.

The dataset is comprised of video files, with 494 entries depicting non-violent scenes and 795 depicting violent scenes. Every video file has audio, and that audio can go from the scene in itself or just noise.

The violence scenarios can be described as:

1. A person (P1) requests a kiss; A second person (P2) refuses the kiss; P1 insists; P2 slaps P1; A conflict ensues between the two.
2. P2 is on the phone; P1 approaches; P2 shoves P1; P1 insists on seeing the phone.
3. P2 is sleeping; P1 drinks water from a bottle; P1 throws the bottle at P2; P2 wakes up and shoves P1.
4. P1 and P2 are on the phone. They engage in a dispute, leading to a physical conflict.
5. P1 threatens P2 with a knife; P1 harasses P2 by touching their body.
6. P1 pulls out a knife and points it at P2; P1 stabs P2.
7. P1 draws a gun and points it at P2; P1 shoots P2 with the weapon.
8. P1 greets P2; P1 shows something on the phone and threatens P2 with scissors; P1 robs P2.
9. P1 approaches P2, touches a non-sexual part of P2; P2 slaps P1.
10. P2 threatens to strike P1; P1 behaves in a provocative manner, and P2 slaps P1.
11. P2 performs an obscene gesture; P1 attacks P2 with a closed fist and attempts to strangle him.
12. A discussion with hand gestures, shoves, and punches.

As for the non-violent scenarios, they can be described as follows:

1. P1 is writing in a notebook, while P2 is applying hand sanitizer.
2. P1 answers a phone call; P2 uses a notebook and coughs.
3. P1 drinks and eats; P2 takes pictures.
4. P1 yawns and stretches; P2 puts on the headphones to listen to music.
5. P1 sneezes; P2 reads a book/newspaper/magazine.
6. P1 applies lipstick and arranges her hair; P2 sleeps.
7. P1 asks P2 to take a picture; P2 takes several pictures of him; P2 shows P1 the pictures taken.
8. P1 and P2 talk; P2 cries; P1 and P2 embrace.

D. Pre-Processing

Data pre-processing is a crucial step to reduce the difficulty of learning features of the algorithm [48]. In the section III.C, we talked about the dataset created. This dataset only had 494 videos without violence on it and 795 videos with violence. The data pre-processing follows the flow represented in Fig. 2.

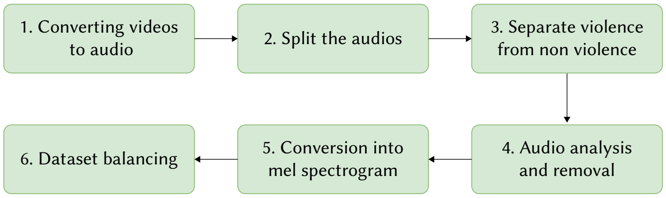
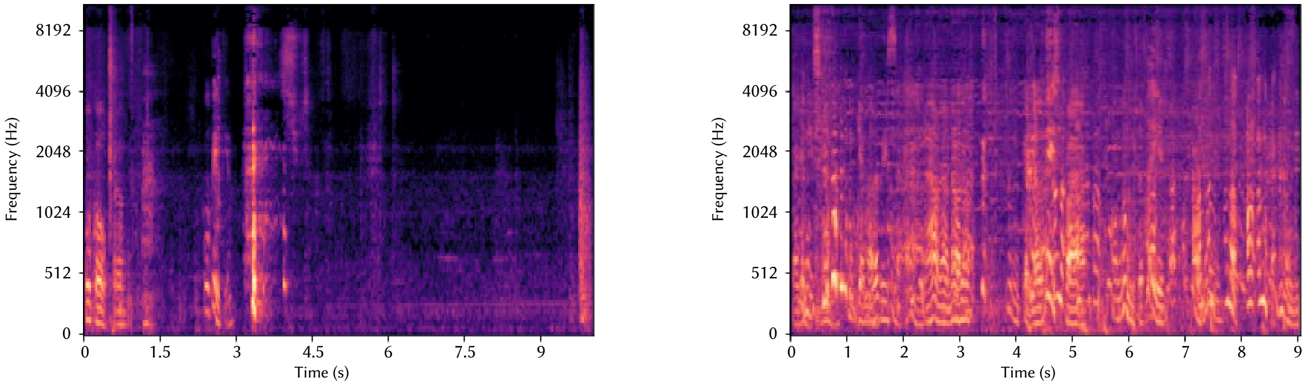

1. Converting videos to audio
2. Split the audios
3. Separate violence from non violence
4. Audio analysis and removal
5. Conversion into mel spectrogram
6. Dataset balancing

Fig. 2. Pre-processing steps.

As for the first step all the videos had to be converted to audios so we could create the mel spectrograms of each one. In the second step, the audios without violence that had more than 40 seconds were split in half so each entry became two entries in the dataset with the same label. The audio recordings of violent incidents were typically longer than those of non-violent incidents, but the issue was that violence was often not present in the beginning of the audio. The solution involved inspecting each audio individually to determine the start of violence, and using the "pydub" library[1], the audio could be divided into two parts - one representing non-violence, and the other representing violence. During the process of analysing each audio (step 4), it was discovered that some files lacked content and that some audio recordings did not have meaningful information for the data (e.g. audio recordings that only had background noise). These were removed from the dataset. By the end of the fourth step in the workflow, the dataset had 860 audio files of non-violence and 755 audio files of violence, for a total of 1615 audio files. The step five was mixed with step six. We converted every audio into a mel spectrogram that could represent the audio in itself, so all the 1615 were converted and then it was decided that a good approach would be to balance the dataset so we used some entries from the RLVS dataset refered in section III.B. We tried to find the best violence videos in RLVS dataset that could go into our dataset. We found 105 violent videos, and those were converted to audio and then converted into a mel spectrogram to be added to the dataset. Ending this workflow, the dataset had 860 non-violence mel spectrograms and 860 violent mel spectrograms, with a total of 1720 mel spetrograms.

In Fig. 3 it is shown what a mel spectrogram created from an audio with violence and and audio without violence looks like.

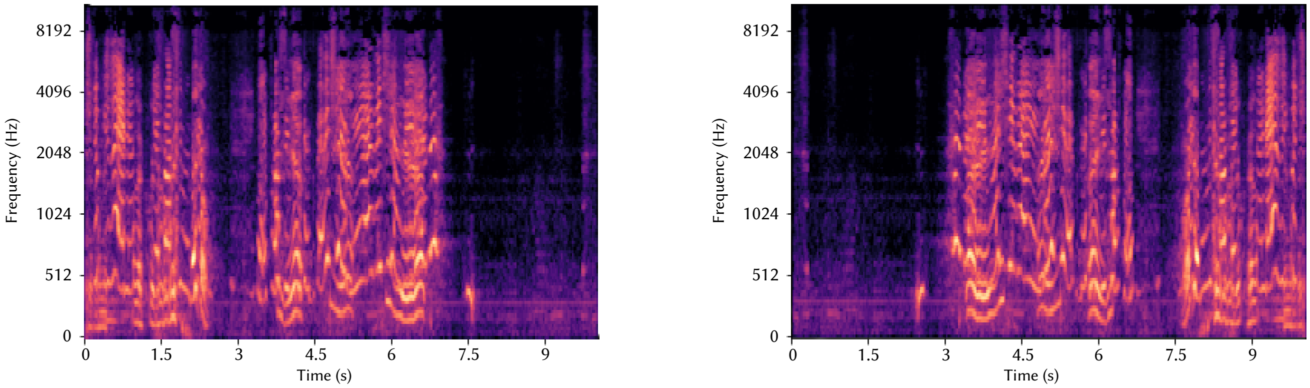
[1] https://thepythoncode.com/assistant/transformation-details/cutting-audio-files-in-python-with-pydub/



(a)

(b)

Fig. 3. Mel spectrogram created from (a) an audio without violence and (b) an audio with violence.



(a) Original spectrogram.

(b) Spectogram after flipping.

Fig. 4. Horizontal flipping of a spectrogram.

E. Data Augmentation

In deep learning, there is the notion that large datasets lead to better training which can lead to best accuracies. But collecting enough data to create a suitable dataset can be a challenging task at times. The data augmentation mechanism is frequently employed to generate a large volume of training data by adding synthetic data to the dataset. These synthetic data may consist of copies of existing data but with minor changes or completely new data created from the data already on the dataset [49].

Some examples of data augmentation were:

- **Cropping**: The process involves trimming the image, thus decreasing its input size;
- **Rotation**: Consists in rotating the image, between 1 and 359 degrees;
- **Translation**: This process involves in moving the image along the x-axis or y-axis (left, right, up or down);
- **Flipping**: The image is flipped vertically or horizontally;
- **Scalling or resizing**: The image is resized to a given size;
- **Noise injection**: The process entails adding a matrix of randomly generated values;
- And other methods that are more complex to achieve.

For the problem meant to be solved which was the classification of mel spectrograms, the data augmentation that we used was flipping. With this flipping method we were able to duplicate the number of entries of the dataset, where we performed a horizontal flip in each entry. This resulted in a dataset with 3440 mel spectrograms, 1720 for each class.

This flip method was most useful in violence entries because some of the entries had violence since the start and it would calm down later in the audio. But after this data augmentation, we were able to show that it also can start with a calm environment and then escalate the situation to pure violence. The Fig. 4 ilustrates this last case, where in a) we see a mel spectrogram taken from an audio with violence and in b) this same mel spetrogram after being flipped. Every mel spectrogram had also his axis removed for the final dataset.

F. Algorithms

This section provides an overview of all the algorithms evaluated in this project. The tested algorithms include Convolutional Neural Network (CNN), EfficientNetB0, EfficientNetB1, EfficientNetB2, MobileNet, MobileNetV2, ResNet50, VGG16, VGG19, and Xception. The selection of these models is supported by the findings from the literature review.

Convolutional Neural Network (CNN) is a deep learning algorithm used for image classification. Its architecture was inspired by the human brain. This network can extract features directly from the image without requiring human assistance [50].

EfficientNets are a type of artificial neural networks that take into account the scaling process and the importance of the base network. They feature a unique mechanism called the compound scaling method, which enables the network to be uniformly scaled in terms of depth, width, and resolution. The base network is the EfficientNetB0 (for example, EfficientNetB1 is a scaled version of EfficientNetB0). These networks can achieve better performance than existing CNN models while using less number of parameters [51] . EfficientNetB0, EfficientNetB1, and EfficientNetB2 belong to the EfficientNet family of image classification models. Here are the key distinctions between these three models: i)Depth: EfficientNetB0 has the fewest layers with 20 convolutional layers, while EfficientNetB1 has 23 convolutional layers, and EfficientNetB2 has 26 convolutional layers; ii) Width: As we progress from B0 to B2, the width of the network increases. This means that the number of channels in each convolutional layer is larger in EfficientNetB2 than in EfficientNetB1, and larger in EfficientNetB1 than in EfficientNetB0; and iii) Resolution: The input resolution of EfficientNetB2 is higher than that of EfficientNetB1, and EfficientNetB1 has a higher resolution than EfficientNetB0. Consequently, EfficientNetB2 is better equipped to handle high-resolution images. In general, moving from EfficientNetB0 to B2 results in a model that is deeper, wider, and more capable of processing high-resolution images. However, with each step up the scale, the model also becomes more computationally demanding. The choice of which model to use depends on specific task requirements, including available compute resources and the resolution of the input images [51].

MobileNet was designed for efficient deployment on mobile and embedded devices with limited computational resources. This network is based on a CNN and uses depthwise separable convolutions, which leads to a decrease in the number of parameters when comparing to networks with regular convolutions and with the same depth. This process allows the network to be a lighter neural network [52].

The Residual Network (ResNet) was created to address the issue of the vanishing gradient problem, making it possible to train a network with more than 1000 layers [7].

VGG, an acronym for Visual Geometry Group, is a deep convolutional neural network (CNN) architecture that is composed of multiple layers. This model is used for image classification and has been trained using the ImageNet dataset, making it a popular choice for transfer learning. VGG16 means that the neural network has 16 layers, while the VGG19 has 19 layers [53].

Xception is a network developed by Google, for image classification tasks. It uses the idea of depthwise separable convolution layers that decreases the computacional cost, and it was designed to be a more efficient alternative for the overall Inception architectures [54].

These models have been developed specifically for image classification tasks. EfficientNet is a family of optimized and efficient models that also achieve top-notch accuracy in image classification. MobileNet, on the other hand, is a family of lightweight and fast models, making them ideal for deployment on mobile and embedded devices. ResNet50 is a CNN architecture that cleverly employs skip connections to address the vanishing gradient problem during training, enabling the creation of very deep neural networks without compromising performance. VGG16 and VGG19 are CNN architectures known for their utilization of small 3x3 convolutional filters. While they demonstrate strong performance in image classification tasks, they can be computationally expensive during both training and deployment. Lastly, Xception is a CNN architecture that incorporates depthwise separable convolutions, performing a depthwise convolution followed by a pointwise convolution. This design results in better performance with fewer parameters compared to other architectures.

G. Training Details

As to prepare for the training of the algorithm, the dataset was divided initially into train and test. We decided that 80% of the dataset would be for the training, and 20%for the testing, giving a total of 2752 entries for training and 688 entries for testing (equal distribution between classes). With the necessity of a validation set, we used the 80% for training where 80% of those would be for training and the other 20% would be for validation. Ending the split phase, the train set consisted of 2202 entries, validation set had 550 entries, and the test set had 688.

Table II shows the class distribution between the three sets (train, validation and test set).

TABLE II. Train, Validation and Test Set

Dataset	Violence	Non violence	Total
Train	1101	1101	2202
Validation	275	275	550
Test	344	344	688

All the algorithms used the same callbacks: EarlyStopping[2], ReduceLROnPlateau[3], ModelCheckpoint[4], and TensorBoard[5]. The EarlyStopping was meant to stop the training of the algorithm in case the validation loss was not getting better. It had a patience of 25 for the CNN and 10 for the other algorithms. The ReduceLROnPlateau would reduce the learning rate if the validation loss did not improve; we used a factor of 0.1 for every algorithm, a patience of 10 for CNN and 5 for the rest of the algorithms. The ModelCheckpoint would save the model weights in a file. To visualize all the training done (accuracy and loss during the different epochs) it was used the callback TensorBoard.

The Table III shows all the training details for each algorithm. All of them used Adamax as optimizer, with a learning rate of 0.001. CNN was meant to run for 200 epochs, while the others ran for 40 epochs. Batch size used was 64, with a resize to (150,150) on each entry (mel spectrograms). The last column of the table (EarlyStopping) shows the epoch that the algorithm stopped the training because of the callback EarlyStopping.

As pre-trained networks on *ImageNet* have demonstrated remarkable results across multiple fields such as image classification datasets, object detection, action recognition, and more [55], we decided that all the algorithms would use the weights from training the network on ImageNet dataset. Those weights are available on the python library *Keras*[6].

The training was done on a computer with a GeForce GTX 1070 Ti, 16GB RAM, and a AMD Ryzen 5 2600 as CPU.

IV. Results and Discussion

Table IV shows the best results obtained by the different algorithms, with all values corresponding to the epoch that achieved the best validation loss.

The VGG16 network performed better on the test in terms of accuracy (91.86%) than VGG19 (91.28%). However, it has slightly worse test loss compared to VGG19. Of the transfer learning networks, Xception had the lowest test accuracy at 90.70%, while ResNet50 had a slightly better result at 90.84%. Both had a similar test loss that was around 0.25.

In regards to the MobileNet family of networks, MobileNet achieved superior results in training, validation, and test, even reaching an accuracy of 93.31% on test. Nevertheless, MobileNetV2 also performed well on test with an accuracy of 92.44% when compared to the previously evaluated networks.

The family of EfficientNet achieved the best results, with EfficientNetB1 achieving the highest accuracy in the test (95.06%), followed by EfficientNetB0 with 94.19%. Moreover, in terms of test loss, EfficientNetB1 had the best performance with a loss of 0.1685, followed by EfficientNetB0 with 0.1772. Although EfficientNetB2 had the weakest performance within the family, it still achieved a satisfactory accuracy of 92.88%.

[2] https://keras.io/api/callbacks/early_stopping/

[3] https://keras.io/api/callbacks/reduce_r_on_plateau/

[4] https://keras.io/api/callbacks/model_checkpoint/

[5] https://www.tensorflow.org/

[6] https://keras.io/

TABLE III. Details of the Training for Each Algorithm

Algorithm	Optimizer	Learning Rate	Epochs	Batch	Resize	EarlyStopping
CNN	Adamax	0.001	200	64	(150,150)	82
EfficientNetB0	Adamax	0.001	40	64	(150,150)	28
EfficientNetB1	Adamax	0.001	40	64	(150,150)	31
EfficientNetB2	Adamax	0.001	40	64	(150,150)	21
MobileNet	Adamax	0.001	40	64	(150,150)	23
MobileNetV2	Adamax	0.001	40	64	(150,150)	20
ResNet50	Adamax	0.001	40	64	(150,150)	14
VGG16	Adamax	0.001	40	64	(150,150)	16
VGG19	Adamax	0.001	40	64	(150,150)	15
Xception	Adamax	0.001	40	64	(150,150)	20

TABLE IV. Accuracy and Loss on the Train, Validation and Test Set, for Each Algorithm

Algorithm	Train	Train Loss	Validation	Validation Loss	Test	Test Loss
CNN	89.28	0.2827	90.00	0.2579	89.53	0.2877
EfficientNetB0	95.19	0.1427	91.09	0.2030	94.19	0.1772
EfficientNetB1	95.19	0.1328	91.92	0.1912	95.06	0.1685
EfficientNetB2	91.05	0.2178	91.82	0.2117	92.88	0.2139
MobileNet	92.51	0.2087	89.82	0.2090	93.31	0.1926
MobileNetV2	88.74	0.2367	88.36	0.2457	92.44	0.2054
ResNet50	83.11	0.3583	89.64	0.2436	90.84	0.2535
VGG16	88.33	0.2679	88.00	0.2873	91.86	0.2259
VGG19	87.33	0.3016	86.55	0.3270	91.28	0.2238
Xception	92.14	0.2056	87.82	0.2967	90.70	0.2527

TABLE V. Results From Precision, Recall and F1-Score of the Algorithms

Algorithm	Precision				Recall	F1-Score
	0	1	0	1	0	1
CNN	0.89	0.90	0.90	0.89	0.90	0.89
EfficientNetB0	0.96	0.92	0.92	0.96	0.94	0.94
EfficientNetB1	0.95	0.95	0.95	0.95	0.95	0.95
EfficientNetB2	0.93	0.93	0.93	0.93	0.93	0.93
MobileNet	0.93	0.94	0.94	0.93	0.93	0.93
MobileNetV2	0.90	0.96	0.96	0.89	0.93	0.92
ResNet50	0.93	0.89	0.89	0.93	0.91	0.91
VGG16	0.92	0.92	0.92	0.92	0.92	0.92
VGG19	0.91	0.92	0.92	0.90	0.91	0.91
Xception	0.92	0.89	0.89	0.92	0.91	0.91

Class 0 represents non-violence inputs; 1 represents violence inputs.

Accuracy and loss are the primary metrics used to evaluate the behavior of various algorithms. However, there are other metrics that assist in the evaluation of algorithms, and these metrics are widely used in the world of ML. The metrics that are often used to evaluate the algorithm are: precision, recall, and f1-score [56].

Table V presents the recall, precision, and f1-score values for each class, with the value 0 representing the non-violence class and the value 1 representing the violence class.

The outcome of an algorithm can fall into four distinct categories, namely, TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative). When considering violence entries as "Positive" and non-violence entries as "Negative," TP represents the correctly predicted violence entries, TN denotes the correctly predicted non-violence entries, FP includes the misclassified non-violence entries, and FN comprises the misclassified violence entries. These four categories collectively form a matrix known as the confusion matrix, which effectively reflects the algorithm's performance.

Taking into account the concept of accuracy and the four aforementioned values, the formula is as follows:

$$\frac{TP + TN}{TP + TN + FP + FN} = \frac{\#\ \text{CorrectForecast}}{\#\ \text{Forecast}} \tag{1}$$

In contrast, the precision indicates the number of correct positive forecast (Equation (2)).

$$\frac{TP}{TP + FP} = \frac{\#\ \text{CorrectlyPredictedPositives}}{\#\ \text{PositiveForecasts}} \tag{2}$$

The recall, as stated in Equation (3), represents the count of true positive cases that the algorithm correctly predicted.

$$\frac{TP}{TP + FN} = \frac{\#\ \text{CorrectlyPredictedPositives}}{\#\ \text{TotalPositiveDataset}} \tag{3}$$

Finally, the f1-score (Equation (4)) combines the *precision* with the *recall*, in order to produce a value that represents both weights (*precision* and *recall*) in a balanced way.

$$F1 - \text{Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

EfficientNetB0 achieved the highest precision of 96% for the non-violence class, although its precision for the violence class was not as high. Nevertheless, it had the highest recall for the violence class, with 96%. EfficientNetB1 came in second for precision for the non-violence class with 95%, and the same precision value was obtained for the violence class. Moreover, EfficientNetB1 attained a recall of 95% for both classes, and the f1-score demonstrated identical results of 95%. EfficientNetB2 achieved a value of 93% in all the analyzed fields.

MobileNetV2 had the highest recall for the non-violence class at 96%. However, it had the poorest recall for the violence class, which is an important consideration when choosing which algorithm to use, even though it had the highest precision for the violence class. The MobileNet algorithm presented good results in all fields of the confusion matrix, even though it did not perform the best when compared to all the algorithms.

Although MobileNetV2 had good accuracy and precision for the violence class, this network is not suitable for classifying violent inputs as it has the lowest recall for this class, which is the most important class to classify.

Evaluating all the results obtained in the two tables, the algorithms most suitable for this problem are: EfficientNetB1, EfficientNetB0, and MobileNet.

A. (RQ1) Can Violence Inside a Car Be Effectively Detected Using Audio Data and Deep Learning Models?

Violence inside a car can be effectively detected using audio data and deep learning models. Audio-based violence detection has gained significant attention in recent years, and deep learning models have shown promising results in accurately classifying violent and non-violent audio events.

The results presented in Table 4 demonstrate the high accuracy in detecting violence behavior. As previously mentioned, the models from the EfficientNet family showcased the best performance. When compared to the background, particularly the study by Duraes, Santos, Marcondes, Hammerschmidt and Novais [18], our models yielded superior results. It is important to note that the other background studies have not been specifically applied to the unique environment inside a car.

B. (RQ2) Can the Use of Data Augmentation Enhance the Accuracy of Violence Detection Results?

Data augmentation typically leads to several benefits in the context of deep learning models:i) improved model accuracy, by creating variations in the training data, data augmentation can enhance the accuracy of deep learning models, particularly when dealing with small datasets; ii) increased amount of training data, because obtaining large amounts of labeled data can be challenging and costly; iii) reduced overfitting, because overfitting occurs when a model becomes overly complex and starts fitting noise in the training data instead of the underlying pattern and data augmentation introduces variations to the training data, mitigating overfitting and preventing the model from relying too heavily on a limited number of training examples; iv) better generalization by adding variability to the training data through data augmentation aids deep learning models in generalizing better to new and unseen data, leading to improved performance in real-world scenarios; and v) faster model development, where data augmentation can accelerate the model development process by reducing the time required to collect and label large datasets for training deep learning models.

In comparison with the previous study [26], where data augmentation was not applied, the results presented in this paper show a better increase in performance.

V. Conclusion

The fact that violence is very present in today's society makes the study of violence detection an asset.

Determining how to capture violence is the primary factor that determines the selection of an architecture. Studies have shown that violence can be captured using either video (cameras) or audio (microphones). Since the use of audio to detect violence has more advantages when compared to video, it was decided that audio would be the mechanism to use.

To enable ML architectures to accurately classify audio, it was necessary to find a way to represent all the information contained in it in a compact way (such as an image). Mel spectrograms were utilized to represent audio as images for this task, since this approach is commonly employed and yields good accuracies in audio classification.

Datasets that contained the necessary constraints for the problem were also sought. However, there was no dataset that had all the necessary constraints, so a dataset created by researchers was preprocessed accordingly. A data augmentation process was also applied to the dataset, resulting in a dataset with twice the amount of data.

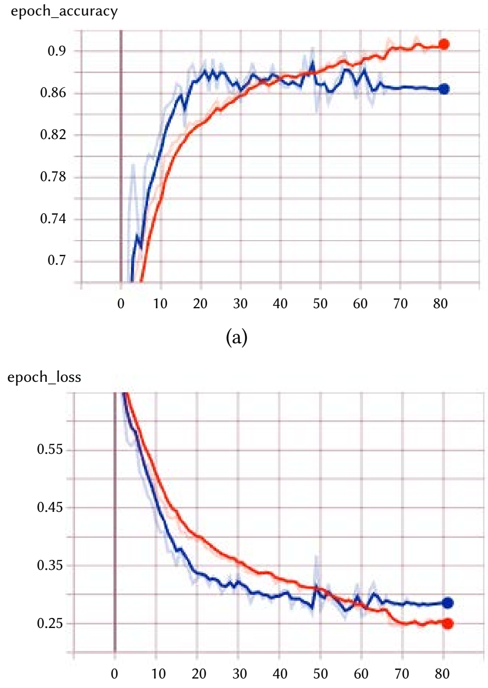
For the final evaluation, the custom CNN algorithm, EfficientNetB0, EfficientNetB1, EfficientNetB2, MobileNet, MobileNetV2, ResNet50, VGG16, VGG19, and Xception were evaluated. The algorithm that achieved the highest accuracy was EfficientNetB1 with an accuracy of 95.06%, followed by EfficientNetB0 with 94.19%, making the EfficientNetB1 the best algorithm to use in order to detect violence in audio. Additionally, it was found that the worst neural network for classifying violence inputs is MobileNetV2, so it should not be the most suitable for solving the problem at hand.

In future work, the intention is to compare the current approach with other methods, specifically those that involve transforming audio data into text and subsequently analyzing the text. This could involve using techniques such as automatic speech recognition (ASR) to convert the audio content into text transcripts, which can then be further processed and analyzed using natural language processing (NLP) or other text-based analysis methods. By exploring these alternative approaches, researchers aim to gain insights into the effectiveness and suitability of different methodologies for violence detection and potentially discover novel insights from the textual representations of audio data.

Appendix

On Appendix we present Fig. 5 to Fig. 14, which contain the detailed training made during the experience.

Fig. 5 depicted the accuracy and loss training curves over the epochs for CNN model. The model achieved it is best results after 80 epochs.

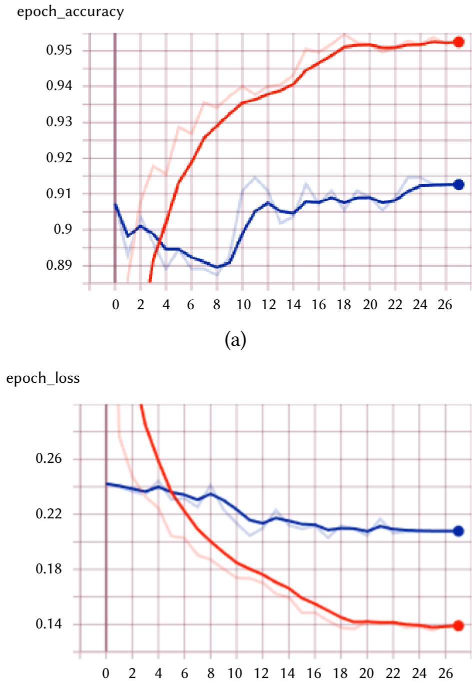


(a)

(b)

Fig. 5. a) Accuracy and b) loss curve for CNN. Training is represented by the orange line, and validation by the blue line.

Fig. 6 illustrated the accuracy and loss training curves across the epochs for the model EfficientNetB0. The model attained its optimal performance after 26 epochs.

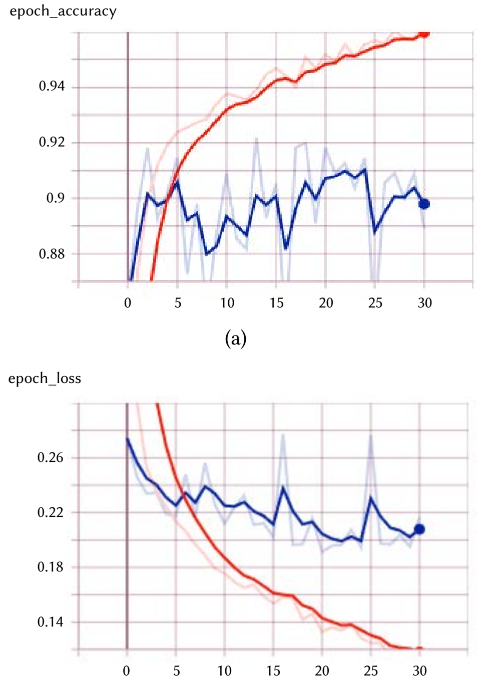


(a)

(b)

Fig. 6. a) Accuracy and b) loss curve for EfficientNetB0. Training is represented by the orange line, and validation by the blue line.

Fig. 7 displayed the accuracy and loss training curves throughout the epochs for the model EfficientNetB1. The model achieved its best performance after 30 epochs. However, it should be noted that the validation line showed some instability during the training process.

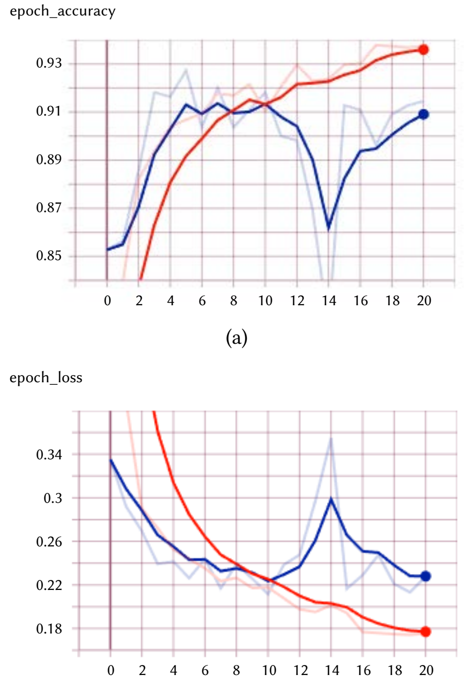


(a)

(b)

Fig. 7. a) Accuracy and b) loss curve for EfficientNetB1. Training is represented by the orange line, and validation by the blue line.

Fig. 8 depicted the accuracy and loss training curves over the epochs for the model EfficientNetB2. The model achieved its peak performance after 20 epochs. However, it should be acknowledged that the validation line displayed some instability around the 14th epoch during the training process.

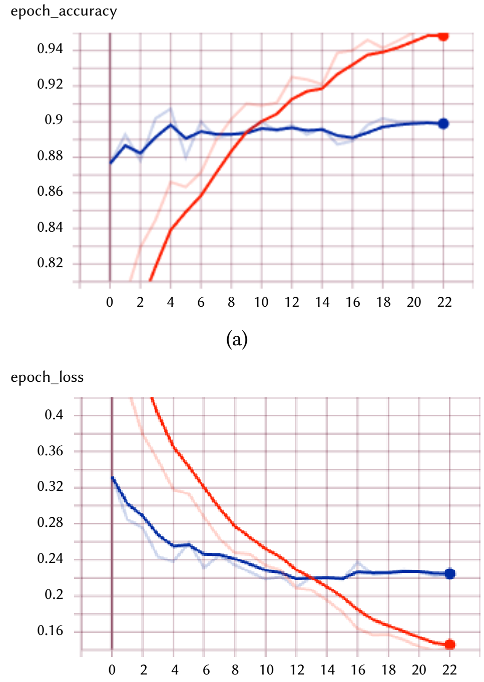


(a)

(b)

Fig. 8. a) Accuracy and b) loss curve for EfficientNetB2. Training is represented by the orange line, and validation by the blue line.

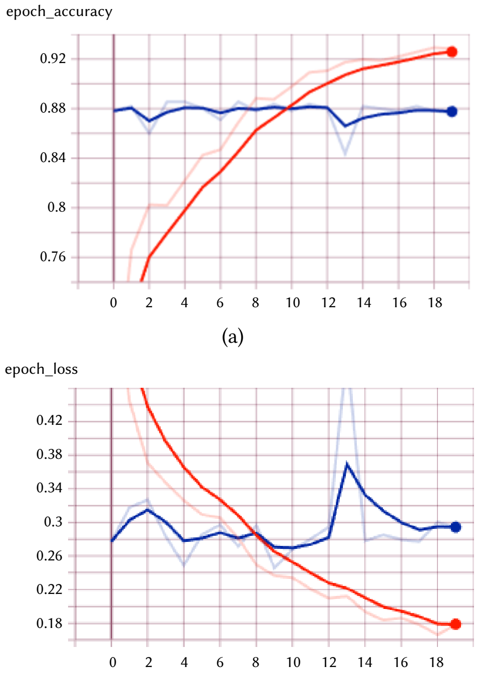
Fig. 9 presented the accuracy and loss training curves across the epochs for the model MobileNet. The model reached its optimal performance after 22 epochs.



(a)

(b)

Fig. 9. a) Accuracy and b) loss curve for MobileNet. Training is represented by the orange line, and validation by the blue line.



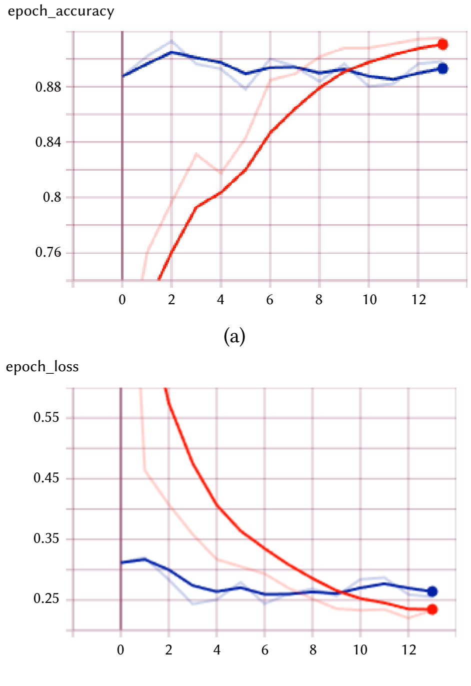
(a)

(b)

Fig. 10. a) Accuracy and b) loss curve for MobileNetV2. Training is represented by the orange line, and validation by the blue line.

Fig. 10 illustrated the accuracy and loss training curves throughout the epochs for the model MobileNetV2. The model achieved its best performance after 19 epochs. However, it is important to note that the validation line showed some instability around the 13th epoch during the training process.

Fig. 11 displayed the accuracy and loss training curves over the epochs for the model ResNet50. The model achieved its best performance after 12 epochs.

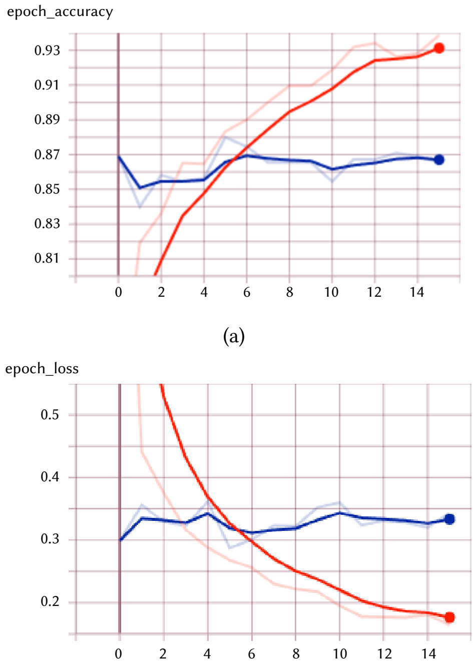


(a)

(b)

Fig. 11. a) Accuracy and b) loss curve for ResNet50. Training is represented by the orange line, and validation by the blue line.

Fig. 12 showed the accuracy and loss training curves throughout the epochs for the model VGG16. The model achieved its peak performance for accuracy after 14 epochs. However, it is important to note that the validation performance stabilized after 6 epochs.

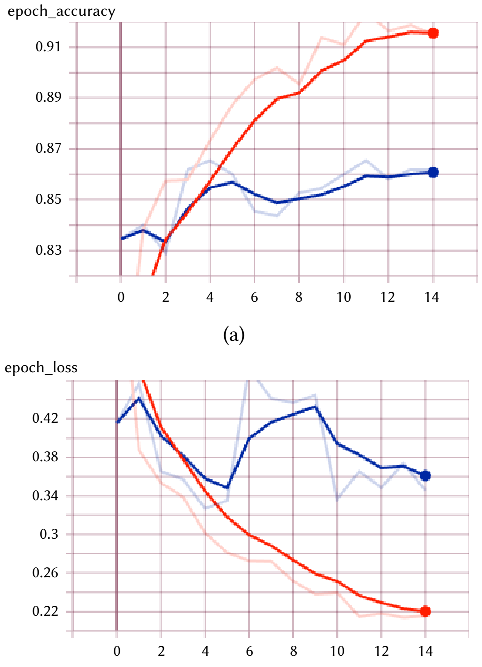


(a)

(b)

Fig. 12. a) Accuracy and b) loss curve for VGG16. Training is represented by the orange line, and validation by the blue line.

Fig. 13 displayed the accuracy and loss training curves over the epochs for the model VGG19. The model reached its optimal accuracy after 14 epochs.

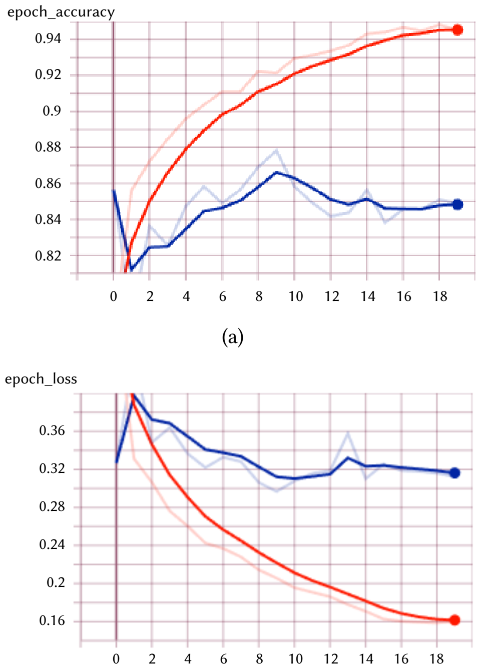


(a)

(b)

Fig. 13. a) Accuracy and b) loss curve for VGG19. Training is represented by the orange line, and validation by the blue line.

Fig. 14 illustrated the accuracy and loss training curves throughout the epochs for the model Xception. The model achieved its best accuracy after 19 epochs.



(a)

(b)

Fig. 14. a) Accuracy and b) loss curve for Xception. Training is represented by the orange line, and validation by the blue line.

Except for the CNN model, which required 80 epochs for training, all the other models needed less than 30 epochs for training. Among them, the VGG16 model had the lowest number of epochs needed for training.

Acknowledgment

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

References

[1] S. Koritsas, M. Boyle, J. Coles, "Factors associated with workplace violence in paramedics," *Prehospital and disaster medicine*, vol. 24, no. 5, pp. 417–421, 2009.

[2] W. So, "Perceived and actual leading causes of death through interpersonal violence in south korea as of 2018," 2019.

[3] APAV, "Estatísticas apav -relatório anual 2020." https://apav.pt/apav_v3/images/pdf/ Estatisticas_APAV_Relatorio_Anual_2020.pdf, 2021. Access at 22/10/2021.

[4] D. Durães, F. Santos, F. S. Marcondes, S. Lange, J. Machado, "Comparison of transfer learning behaviour in violence detection with different public datasets," in *Progress in Artificial Intelligence*, 2021, Springer International Publishing.

[5] D. Durães, F. S. Marcondes, F. Gonçalves, J. Fonseca, J. Machado, P. Novais, "Detection violent behaviors: a survey," in *Ambient Intelligence– Software and Applications: 11th International Symposium on Ambient Intelligence*, 2021, pp. 106–116, Springer.

[6] A. Jan, G. M. Khan, "Real world anomalous scene detection and classification using multilayer deep neural networks," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, no. 2, pp. 158–167, 2023, doi: 10.9781/ijimai.2021.10.010.

[7] F. Santos, D. Durães, F. S. Marcondes, N. Hammerschmidt, S. Lange, J. Machado, P. Novais, "In-car violence detection based on the audio signal," in *Intelligent Data Engineering and Automated Learning– IDEAL 2021: 22nd International Conference, IDEAL 2021, Manchester, UK, November 25–27, 2021, Proceedings 22*, 2021, pp. 437–445, Springer.

[8] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, *et al.*, "Cnn architectures for large- scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, 2017, pp. 131–135, IEEE.

[9] M. Crocco, M. Cristani, A. Trucco, V. Murino, "Audio surveillance: A systematic review," *ACM Computing Surveys (CSUR)*, vol. 48, no. 4, pp. 1–46, 2016.

[10] D. M. Beltrán-Flores, "Ópera nacionalista ecuatoriana," Master's thesis, 2022.

[11] K. Gkountakos, K. Ioannidis, T. Tsikrika, S. Vrochidis, I. Kompatsiaris, "Crowd violence detection from video footage," in *2021 international conference on content-based multimedia indexing (CBMI)*, 2021, pp. 1–4, IEEE.

[12] T. Senst, V. Eiselein, A. Kuhn, T. Sikora, "Crowd violence detection using global motion-compensated lagrangian features and scale-sensitive video-level representation," *IEEE transactions on information forensics and security*, vol. 12, no. 12, pp. 2945–2956, 2017.

[13] K. Gkountakos, K. Ioannidis, T. Tsikrika, S. Vrochidis, I. Kompatsiaris, "A crowd analysis framework for detecting violence scenes," in *Proceedings of the 2020 International Conference on Multimedia Retrieval*, 2020, pp. 276–280.

[14] T. Hassner, Y. Itcher, O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behavior," in *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, 2012, pp. 1–6, IEEE.

[15] M. Sharma, T. Gupta, K. Qiu, X. Hao, R. Hamid, "Cnn- based audio event recognition for automated violence classification and rating for prime video content," *Proc. Interspeech 2022*, pp. 2758-2762, 2022, doi: 10.21437/Interspeech.2022-10053.

[16] A. J. Naik, M. Gopalakrishna, "Violence detection in surveillancevideo-a survey," *International Journal of Latest Research in Engineering and Technology (IJLRET)*, vol. 1, pp. 1–17, 2017.

[17] A. M. Yildiz, P. D. Barua, S. Dogan, M. Baygin, T. Tuncer, C. P. Ooi, H. Fujita, U. R. Acharya, "A novel tree pattern-based violence detection model using audio signals," *Expert Systems with Applications*, vol. 224, p. 120031, 2023.

[18] D. Duraes, F. Santos, F. S. Marcondes, N. Hammerschmidt, P. Novais, "Applying multisensor in-car situations to detect violence," *Expert Systems*, p. e13356, 2023.

[19] V. S. Saravanarajan, R.-C. Chen, C. Dewi, L.-S. Chen, L. Ganesan, "Car crash detection using ensemble deep learning," *Multimedia Tools and Applications*, pp. 1–19, 2023.

[20] F. Reinolds, C. Neto, J. Machado, "Deep learning for activity recognition using audio and video," *Electronics*, vol. 11, no. 5, p. 782, 2022.

[21] I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*. MIT press, 2016.

[22] B. Peixoto, B. Lavi, P. Bestagini, Z. Dias, A. Rocha, "Multimodal violence detection in videos," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2957– 2961, IEEE.

[23] A. S. Arukgoda, *Improving sinhala–tamil translation through deep learning techniques*. PhD dissertation, 2021.

[24] A. Uçar, Y. Demir, C. Güzeliş, "Object recognition and detection with deep learning for autonomous driving applications," *Simulation*, vol. 93, no. 9, pp. 759–769, 2017.

[25] Y. Cho, N. Bianchi-Berthouze, S. J. Julier, "Deepbreath: Deep learning of breathing patterns for automatic stress recognition using low-cost thermal imaging in unconstrained settings," in *2017 Seventh international conference on affective computing and intelligent interaction (acii)*, 2017, pp. 456–463, IEEE.

[26] B. Veloso, D. Durães, P. Novais, "Analysis of machine learning algorithms for violence detection in audio," in *Highlights in Practical Applications of Agents, Multi- Agent Systems, and Complex Systems Simulation. The PAAMS Collection: International Workshops of PAAMS 2022, L'Aquila, Italy, July 13–15, 2022, Proceedings*, 2022, pp. 210–221, Springer.

[27] H. Souto, R. Mello, A. Furtado, "An acoustic scene classification approach involving domestic violence using machine learning," in *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*, 2019, pp.

705–716, SBC.

[28] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[29] J.-L. Rouas, J. Louradour, S. Ambellouis, "Audio events detection in public transport vehicle," in *2006 IEEE Intelligent Transportation Systems Conference*, 2006, pp. 733–738, IEEE.

[30] J. F. Gaviria, A. Escalante-Perez, J. C. Castiblanco, N. Vergara, V. Parra-Garces, J. D. Serrano, A. F. Zambrano, L. F. Giraldo, "Deep learning-based portable device for audio distress signal recognition in urban areas," *Applied Sciences*, vol. 10, no. 21, 2020, doi: 10.3390/app10217448.

[31] M. S. Hossain, G. Muhammad, "Emotion recognition using deep learning approach from audio–visual emotional big data," *Information Fusion*, vol. 49, pp. 69– 78, 2019.

[32] A. Arronte Alvarez, F. Gómez, "Motivic pattern classification of music audio signals combining residual and lstm networks," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 6, pp. 208–214, 2021, doi:10.9781/ijimai.2021.01.003.

[33] L. Nanni, G. Maguolo, M. Paci, "Data augmentation approaches for improving animal audio classification," *Ecological Informatics*, vol. 57, p. 101084, 2020.

[34] Z. Mushtaq, S.-F. Su, "Environmental sound classification using a regularized deep convolutional neural network with data augmentation," *Applied Acoustics*, vol. 167, p. 107389, 2020, doi:10.9781/ijimai.2021.01.003.

[35] S. Mertes, A. Baird, D. Schiller, B. W. Schuller, E. André, "An evolutionary-based generative approach for audio data augmentation," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, 2020, pp. 1–6, IEEE.

[36] B. Zoph, E. D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, Q. V. Le, "Learning data augmentation strategies for object detection," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, 2020, pp. 566–583, Springer.

[37] L. Nanni, Y. M. Costa, R. L. Aguiar, R. B. Mangolin, S. Brahnam, C. N. Silla, "Ensemble of convolutional neural networks to improve animal audio classification," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2020, pp. 1–14, 2020.

[38] K. Choi, G. Fazekas, K. Cho, M. Sandler, "A tutorial on deep learning for music information retrieval," *arXiv preprint arXiv:1709.04396*, 2017.

[39] M. S. Hossain, G. Muhammad, "Emotion recognition using deep learning approach from audio–visual emotional big data," *Information Fusion*, vol. 49, pp. 69– 78, 2019.

[40] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[41] D. de Benito-Gorron, A. Lozano-Diez, D. T. Toledano, J. Gonzalez-Rodriguez, "Exploring convolutional, recurrent, and hybrid deep neural networks for speech and music detection in a large audio dataset," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2019, no. 1, pp. 1–18, 2019.

[42] P. Wu, j. Liu, Y. Shi, Y. Sun, F. Shao, Z. Wu, Z. Yang, "Not only look, but also listen: Learning multimodal violence detection under weak supervision," in *European Conference on Computer Vision (ECCV)*, 2020.

[43] W.-F. Pang, Q.-H. He, Y.-j. Hu, Y.-X. Li, "Violence detection in videos based on fusing visual and audio information," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 2260–2264, IEEE.

[44] R.-R. O. S. Lab, "Ntu cctv-fights dataset." https://rose1.ntu.edu.sg/dataset/cctvFights/, 2019. Access 03/02/2023.

[45] M. Perez, A. C. Kot, A. Rocha, "Detection of real-world fights in surveillance videos," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2662–2666, IEEE.

[46] M. Schedi, M. Sjöberg, I. Mironică, B. Ionescu, V. L. Quang, Y.-G. Jiang, C.-H. Demarty, "Vsd2014: A dataset for violent scenes detection in hollywood movies and web videos," in *2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2015, pp. 1–6, IEEE.

[47] M. M. Soliman, M. H. Kamal, M. A. E.-M. Nashed, Y. M. Mostafa, B. S. Chawky, D. Khattab, "Violence recognition from videos using deep learning techniques," in *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, 2019, pp. 80–85, IEEE.

[48] S. Tang, S. Yuan, Y. Zhu, "Data preprocessing techniques in convolutional neural network based on fault diagnosis towards rotating machinery," *IEEE Access*, vol. 8, pp. 149487–149496, 2020.

[49] C. Shorten, T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[50] K. O'Shea, R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[51] M. Tan, Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 2019, pp. 6105–6114, PMLR.

[52] D. Sinha, M. El-Sharkawy, "Thin mobilenet: An enhanced mobilenet architecture," in *2019 IEEE 10th annual ubiquitous computing, electronics & mobile communication conference (UEMCON)*, 2019, pp. 0280–0285, IEEE.

[53] J. P. Gujjar, H. P. Kumar, N. N. Chiplunkar, "Image classification and prediction using transfer learning in colab notebook," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 382–385, 2021.

[54] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[55] M. Huh, P. Agrawal, A. A. Efros, "What makes imagenet good for transfer learning?," *arXiv preprint arXiv:1608.08614*, 2016.

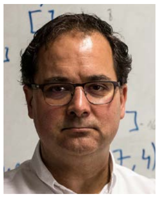
[56] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *arXiv:2010.16061*, 2020, doi: https://doi.org/10.48550/arXiv.2010.16061.

Dalila Durães

Dalila Durães is a Professor of Computer Science at the Department of Informatics, the School of Engineering, the University of Minho (Portugal), and a researcher at the ALGORITMI Centre and LASI Intelligent Systems Associate Laboratory. She is chair of the IEEE in Computational Intelligence Society, Portugal Chapter and a member of the Portuguese Association for Artificial Intelligence (APPIA) social corporate since January 2020 (Secretary of the General Assembly). Dalila is the author of more than 60 scientific publications in international peer-reviewed journals, book chapters, books, and conferences. She is also a member of the editorial board of several international journals. During the past few years, she has served as an expert/reviewer for several conferences and journals. She was also a supervisor of master's and doctoral students and was a member of the jury of several doctoral and master's theses.

Bruno Veloso

Bruno Veloso is 26 years old and he live in Portugal. He studied "Science and Technology" before university, in Portugal, Viana do Castelo, then he enrolled at the University of Minho (Portugal, Braga) in 2015, in the degree of Computer Science. After a year, he decided to change his major to Integrated Master's in Computer Engineering, in which he completed the bachelor's degree in 2020 and he currently finishing the last year of the master's degree (2023), in which he is writing a dissertation related to deep learning. During the master's program, he chose the study profiles of "Data Science" and "Intelligent Systems". With these two profiles, he realized that he would like to pursue a career in machine learning because it is something that belongs to the future and captivates me. Additionally, he is also participating in a research scholarship, in collaboration with two universities and a company, which aims to use machine learning to investigate the success/failure of students. In the future, he would like to find a job in the field of machine learning.



Paulo Novais

Paulo Novais is a Full Professor of Computer Science at the Department of Informatics, the School of Engineering, the University of Minho (Portugal) and a researcher at the ALGORITMI Centre in which he is the leader of the research group ISLab - Synthetic Intelligence lab, and the coordinator of the Portuguese Intelligent Systems Associate Laboratory (LASI). His main research aim is to make systems a little smarter, intelligent and also reliable. He is the co-author of over 400 book chapters, journal papers, conference and workshop papers and books.

Problem Detection in the Edge of IoT Applications

Iván Bernabé-Sánchez, Alberto Fernández, Holger Billhardt, Sascha Ossowski *

CETINIA, Universidad Rey Juan Carlos, Madrid (Spain)

Received 5 May 2022 | Accepted 20 July 2023 | Published 26 July 2023

Abstract

Due to technological advances, Internet of Things (IoT) systems are becoming increasingly complex. They are characterized by being multi-device and geographically distributed, which increases the possibility of errors of different types. In such systems, errors can occur anywhere at any time and fault tolerance becomes an essential characteristic to make them robust and reliable. This paper presents a framework to manage and detect errors and malfunctions of the devices that compose an IoT system. The proposed solution approach takes into account both, simple devices such as sensors or actuators, as well as computationally intensive devices which are distributed geographically. It uses knowledge graphs to model the devices, the system's topology, the software deployed on each device and the relationships between the different elements. The proposed framework retrieves information from log messages and processes this information automatically to detect anomalous situations or malfunctions that may affect the IoT system. This work also presents the ECO ontology to organize the IoT system information.

Keywords

Complex Event Processing, Intelligent Agents, Internet Of Things (IoT), Ontologies.

DOI: 10.9781/ijimai.2023.07.007

I. Introduction

According to [1], 2023 some 29.3 billion devices will be connected to IP networks. This more than triples the world's population. In fact, there will be 3.6 devices per person, a considerable growth of over 50% compared to 2018. Half of these devices will make machine-to-machine (M2M) connections, totalling 14.7 billion M2M connections. This increase in the number of devices and connections will produce an enormous amount of data and create new opportunities for innovative applications in domains such as healthcare [2], environmental sciences and industrialization [3], etc. The inclusion of IoT in these domains requires caution in large-scale implementations because of the risks of saturation of system resources and due to security issues. In many cases, IoT devices are used to improve people's daily activities or optimize important processes in companies, which may expose data [4].

Albeit security of IoT is a major topic addressed in literature [5], there are other important problems that condition the expansion and implementation of IoT solutions. For example, managing a large volume of devices requires dealing with problems like communications interruptions (network connectivity) [6], discontinuity of services, discharge of batteries (energy saving), and problems with the operating environment (overheating, storage management, cybercrime) [7]. All of these issues may apply to any of the devices that integrate an IoT system.

Self-repair or self-healing is defined as a property of systems that are able to identify and diagnose problems that appear during their operation and to determine and propose solution strategies in an autonomous way [8]. More specifically, self-healing provides reliability to a system through responsibility and awareness of the environment. This allows to automatically detect problems and to propose solutions to unwanted situations. In order to do so, a self-healing IoT system must incorporate monitoring, awareness, and knowledge to detect unwanted states. When a problem is detected, the system generates and executes plans with appropriate corrective actions [9], [10].

In this work, we propose a framework for the specification and automatic detection of problems that may occur in an IoT system. This framework consists of independent agents that are distributed on the different devices that make up a system. Setting out from messages stored in log registers, these agents extract information about the operation of devices and the software deployed on them, and process it to identify operating problems. The proposed framework uses knowledge graphs (ontologies) to structure the information, event stream processing to identify problems, and automatic reasoning to infer additional knowledge related to the operation and potential problems of a system. The edge-cloud ontology (ECO) has been designed to structure the system information and possible problems.

The rest of the article is divided into the following sections. Section II contains the state of the art. Section III shows the proposed architectural solution. Section IV details how semantic technologies are used to represent and process the information for identifying existing problems. An example is presented in Section V. We conclude the paper and point to some future lines of research in Section VI. Table I shows the list of acronyms.

II. Related Work

Failures of system elements in IoT systems are usually considered as something inevitable. It is important to consider this possibility and to integrate mechanisms that ensure that the infrastructure will continue to function without interruption, even if some elements fail.

* Corresponding author.

E-mail addresses: ivan.bernabe@urjc.es (I. Bernabé-Sánchez), alberto.fernandez@urjc.es (A. Fernández), holger.billhardt@urjc.es (H. Billhardt), sascha.ossowski@urjc.es (S. Ossowski).

TABLE I. List of Acronyms and Abbreviations Used

Abbreviations	Explanation
ASS	Action Schedule Service
CEP	Complex Event Processing
CLF	Common Log Format
CMA	Complex Management Agent
DIR	Deployed Infrastructure Repository
DPR	Detected Problem Repository
ECO	edge-cloud ontology
ELFF W3C	Extended Log File Format
DPR	Detected Problems Repository
FD	Fog Devices
IoT	Internet of Things
IS	Inference Service
KG	knowledge graphs
LMA	Lightweight Management Agent
LMS	Log Management Service
MMS	Middleware Management Service
OS	operating system
SAREF	Smart Applications REFerence Ontology
SD	Simple Devices
SmD	Smart Devices
SOSA/SSN	Semantic Sensor Network ontology

Some works, like [11]–[13], propose fault-tolerant solutions to recover IoT systems deployed in the cloud and edge computing. These works are mainly focused on managing problems associated with resource exhaustion and performance degradation. However, fault-tolerant distributed systems must be able to go further and handle finer-grained problems such as error management in applications deployed in cloud and edge computing. To this respect, [14], [15] and [16] propose solutions based on micro-services. These works put forward mechanisms for system recovery, but do not describe the previous error detection process. Still, information on the causes of errors and the elements involved can greatly facilitate the generation of potential solutions to a problem.

A starting point for troubleshooting is to know if errors exist and when they appeared. Usually, applications write status information in log files, which are analyzed manually or automatically to find out if there are any problems. Normally, these logs have to be parsed before their contents can be interpreted [17]. A common approach to parse logs is to detect or match with specific error patterns [18], [19]. Other solutions use data mining techniques such as SLCT [20], and its extension LogCluster [21]. These works require a large data set with a large log history to generate efficient log patterns. Still, in recent and very specific or uncommon systems it is complicated to apply these solutions, because there may not be enough information to generate efficient patterns. Some works, such as [14], [15] and [16], also consider mechanisms to recover systems from errors and bring them to the desired operating status. These works use information repositories and catalogues to have a record of the architecture of services that make up an IoT systems. However, such repositories are specifically designed for the proposed solutions and do not have a formal specification of how their information is structured.

We claim that working with structured information can largely facilitate error detection because it allows specifying explicitly the relationships among the different information elements at a conceptual level, and enables the reuse of information across different systems. In particular, knowledge graphs and ontologies can help structuring the failure-related information available in an IoT system. Furthermore, inference based on ontologies may even infer additional information that is not explicitly available. As described in Noy and McGuiness [22], using ontologies provides several benefits: (1) they share a common understanding of information structure among software agents; (2) allow reuse of domain knowledge; (3) domain assumptions are made explicit; (4) domain knowledge can be analyzed. Taking advantage of all these benefits is key to interoperability. As indicated by Bittner et al. [23], as well as by Jasper and Uschold [24], ontologies facilitate the semantic interoperability between humans, computers, and systems. They consider them as a facilitating technologies to achieve communication interoperability between software systems.

The use of ontologies in cloud systems [25] has already been applied to different areas such as resource management [26], service discovery [27], [28], security [29] or even to improve system interoperability. In this line, mOSAIC [30] presents a cloud ontology that provides a detailed description of cloud computing resources. mOSAIC focuses on promoting transparency in accessing multiple clouds. However, mOSAIC has not been updated since its development more than 10 years ago, which implies that new elements that have appeared in the area during this time are not contemplated in this ontology. As a result, other works such as [31] and [32] have appeared so as to try to address these limitations. In [31], a solution for deploying applications on public and private clouds is shown. The solution uses a set of rules to control the deployment of applications. This rule set uses the CAMEL modeling language[1]. ModaClouds [32] is another work that uses ontology-based models to perform semi-automatic code transformations allowing to obtain compatible implementations in public and hybrid cloud provider platforms. The aforementioned works propose mechanisms to improve the interoperability of services hosted by different service providers, improve the description of existing interfaces and even provide decision-making support. However, these projects do not support a broad heterogeneous environment, i.e., they are limited to resource management, hardware accelerators and provide resource abstractions in the cloud. In general, these works are not oriented to work with IoT devices such as sensors, actuators, gateways, etc.

In this line, specific ontologies have been developed to model the capabilities, characteristics and descriptions of systems that integrate IoT devices. The Semantic Sensor Network Ontology (SOSA/SSN) [33] is one of the most prominent efforts in this area. SOSA/SSN describes sensor and actuator networks, their capabilities, features of interest, and observations and serves as a starting point for the creation of new ontologies that integrate these devices. Another effort similar to SOSA is the Smart Applications REFerence Ontology (SAREF) [34] developed by the ETSI's SmartM2M technical committee. SAREF allows the description of devices and their functions and is aligned with the oneM2M ontology [35], which allows syntactic and semantic interoperability between devices and external systems. SAREF and SSN are ontologies that are widely used and there are works that extend their scope of application to other more specific domains. For example, CASO [36] and EEPSA [34] extend SAREF for its application in agriculture and smart buildings domains, respectively. In the case of the SSN ontology, the SSN System module allows modeling systems, capabilities and things.

Still, despite the number of existing original ontologies and their extensions, to the best of our knowledge, there are no ontologies capable of providing mechanisms that integrate information from systems in the cloud, at the edge computing with IoT devices. For this reason, we created and present in this article an ontology for this purpose.

[1] https://camel-dsl.org/

III. Proposed Solution

Fig. 1 shows the type of infrastructures we aim at in this work. The architecture depicts a generic IoT system made up of sensors and other devices that allow the collection of information and interaction with the physical world. In general, the information collected by IoT devices in lower layers is processed, filtered out, and sent to higher levels for further analysis. As we will present in this section, we propose the inclusion of intelligent agents that will monitor the operation and status of the different devices in the system.

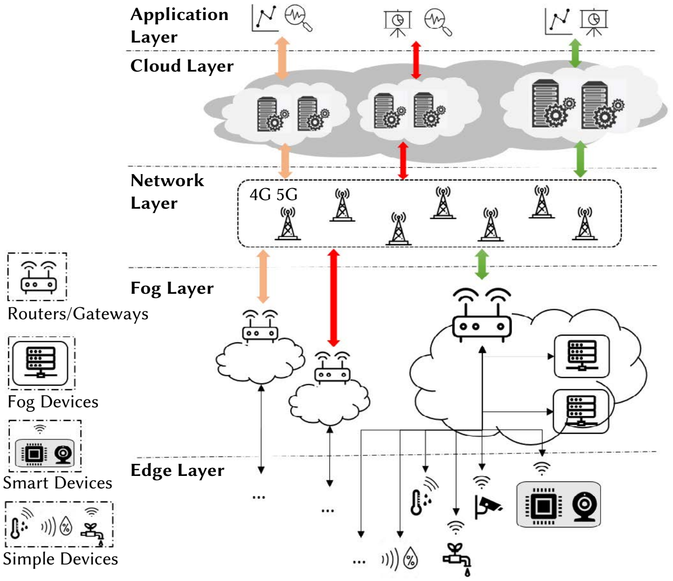


Fig. 1. Typical layered architecture with the distribution of devices that form a generic IoT system. Higher layers include devices with higher computational resources. Green, red and orange arrows represent interactions within different applications.

In the following, we begin with a description of the reference edge-cloud architecture we focus on in this work. Then, we present our architectural proposal for problem identification.

A. Reference Architecture

In the structure of the IoT system described in Fig. 1, there are several types of devices distributed across different layers. A basic IoT system is composed of IoT devices deployed at some physical location, for example, collecting data from the environment and forwarding them to some remote machine for their processing. With this basic infrastructure, systems can become more complex by adding devices, services or applications.

The basic architecture is composed of five layers, each of them grouping different systems, devices and other computational resources, which may involve different service, computation and communication providers. These layers are common to different IoT systems; in fact, Fig. 1 shows different IoT systems (identified by red, green and orange arrows) that are extended across different processing layers. The *Edge Layer* is composed of IoT devices such as sensors, actuators, etc. It is the layer that interacts with the physical world. The *Fog Layer* connects the devices at the Edge Layer with upper layers and can provide basic processing services. The *Network Layer* is in charge of managing communications with data centres located in remote locations. Large-scale computational resources are provided on the Cloud Layer. Finally, applications, typically processing and producing high-level information, are running on the *Application Layer.*

For the two lower layers, in this work, we focus on three types of devices, namely simple, smart and fog devices. Their functionality highly depends on their computational resources. *Simple Devices* (SDs) are low-cost devices (e.g. sensors and actuators). SDs have low computational resources and basic capabilities, for example, to take measurements (depending on the type of physical sensor installed) and send those values to other remote devices where that information is processed. SDs usually use batteries and are typically deployed in remote physical environments.

Smart Devices (*SmDs*) have functions similar to SDs, but with more computing power, which allows them to process the information collected on the same device. For example, in a cultivated field, devices can be deployed to monitor the appearance of imperfections on plant leaves. In that case, the device would have a camera to take pictures of plant leaves, and a running algorithm to detect biological problems (e.g. musty, dry, etc.), which would be forwarded to a processing node in the upper layers.

Finally, *Fog Devices* (*FDs*) are located at the fog layer and have some computing capacity deployed somewhere on a local network. These devices receive information from SmDs and SDs and carry out processing tasks such as aggregation, integration, filtering, statistics, etc.

B. Proposed Architectural Solution

During normal operation, the software deployed on SDs, SmDs and FDs might be subject to errors and malfunctioning. Intelligent management techniques are required to deal with such errors and to make the systems efficient and stable. For this purpose, in our work we propose to use distributed intelligent agents with the aim of monitoring and controlling the software deployed on each of the devices. We focus on distributed systems where software is deployed on lower-level devices as well as on data centres to perform the assigned functions.

Usually, the software is installed and deployed on each device in the traditional way. However, we recommend encapsulating such software in software containers and then deploying such containers on devices. Software containers are a type of lightweight virtualization [37] that allows running multiple isolated software instances on a single operating system (OS) without the need to have an OS for each instance. This type of virtualization is also called containerization and provides encapsulation for each container and resource management. It makes this technology lighter and more efficient than traditional virtualization technologies which require an operating system on each instance. The encapsulation offered by containers does not affect the normal operation of the software running inside them and facilitates their deployment and management. Containers are managed independently of other containers and the OS installed on the device. Generally, a middleware (also called framework) is installed between the OS and the containers. The middleware is responsible for the management of containers and provides mechanisms and interfaces to obtain information and control them.

Using software containers is not only an advantage in terms of ease of management but also offers heterogeneity in terms of being able to run the same container on different devices. As software containers require to work a middleware placed between the device's operating system and the containers, the same container can be executed on different devices if these devices have the middleware installed. The container will be executed on the device regardless of the type of OS and hardware that integrates the device. This is important for the solution proposed in this work because it facilitates moving and running software services between devices.

The container middleware provides mechanisms for starting, stopping and deployment of containers regardless of the device where it is deployed or the software it encapsulates. Fig. 2 shows our proposal of a container-based software architecture that is to be used on the devices deployed in the architecture of Fig. 1. Using makes it easy to deploy software on any device in a system. This is because most container frameworks can connect to remote repositories where the software has been previously uploaded to easily download, install and run the desired software on any particular device. This feature facilitates the resolution of device failures since software from devices with errors can easily be transferred to other devices.

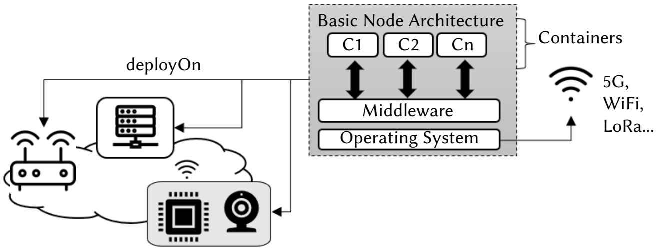


Fig. 2. Device architecture that manages installed software through software containers.

In this paper, we propose two types of intelligent agents that control the functioning of the system. The first one is the *Lightweight Management Agent* (*LMA*), which is responsible for collecting information about the software and device on which it is deployed. LMAs send the collected information and can receive actions to apply on the device. The second type of agent is the *Complex Management Agent* (*CMA*). CMAs are able to carry out more complex reasoning processes, including receiving information from LMAs, detecting existing problems and generating local actions to alleviate existing problems and bring the device operation back to a desired state. CMAs are typically located in the fog or in the cloud. CMAs are deployed at the fog layer to troubleshoot unwanted situations in local device networks. CMAs can also be deployed in the cloud where the CMA is responsible for managing problems that cannot be resolved on the fog level. CMAs are prepared to operate with limited resources but are also capable of dealing with complex problems by scaling the computational resources of the CMA. A CMA deployed in the cloud is capable of addressing problems using a large number of devices and parameters.

Fig. 3 shows the integration of the solution proposed in this work into the general architecture shown in Fig. 1. LMAs and CMAs are described in more detail below and Fig. 4 shows their integration on the different devices.

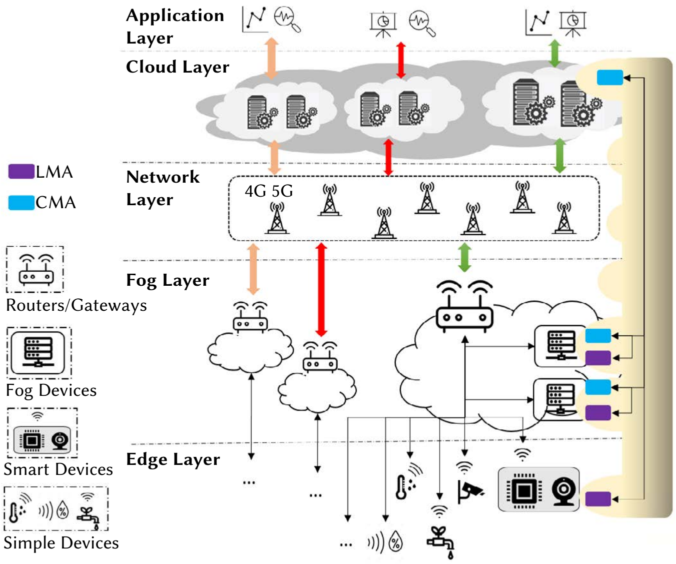


Fig. 3. Integration of the solution proposed in this paper into a generic IoT system.

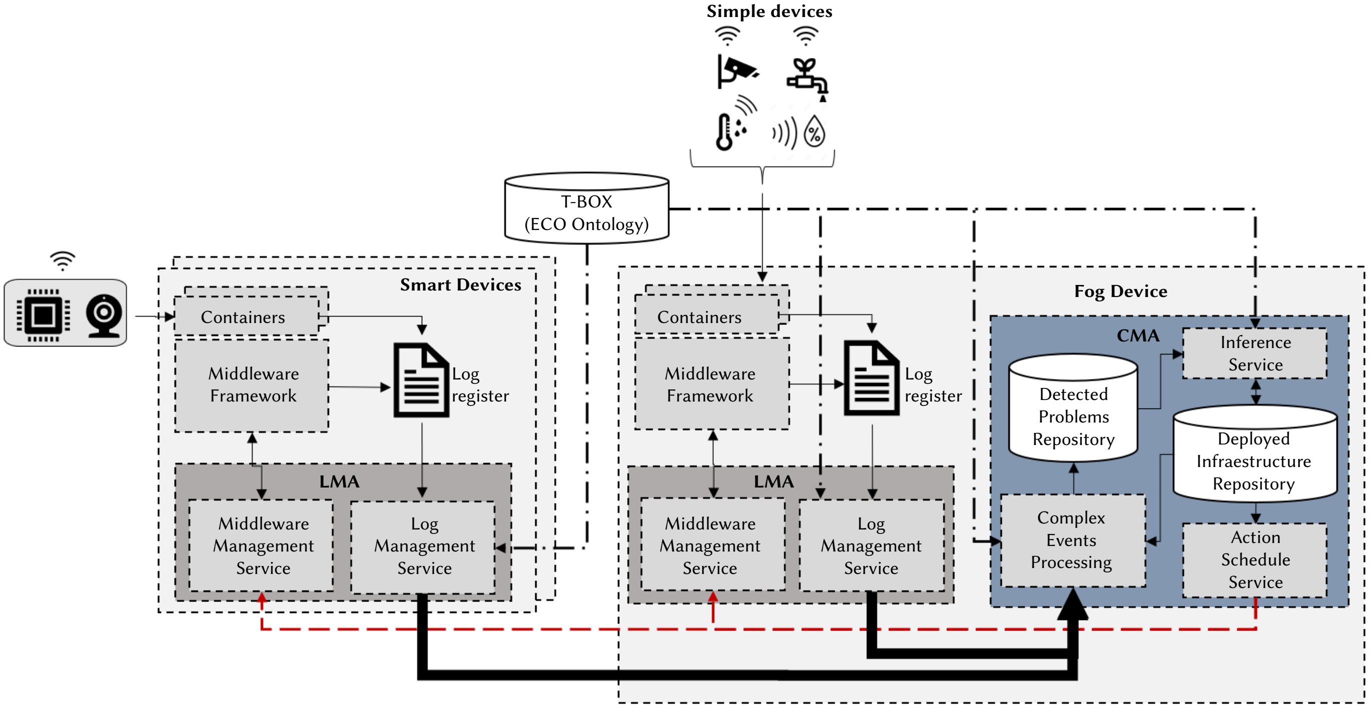


Fig. 4. Disposition and connections of agents, devices, services and software of elements contained in Fig. 1.

1. Lightweight Management Agent

Lightweight Management Agents are deployed on IoT devices with low computational resources, such as SmDs. They read information about the operation and status of the device they control and, if situations are detected that require attention, collect the related information and send it to a CMA, which will process that information and propose corrective actions. LMAs extract information from log files generated by the deployed software, middleware (if installed) or operating system (OS).

The LMA architecture is composed of two services: the Log Management Service (LMS) and the Middleware Management Service (MMS). The Log Management Service is responsible for extracting the data provided from applications, middleware and OS. In addition, it periodically extracts information related to the operating status of the device (e.g. battery charge status, resource usage, etc.). This information may indicate whether the device is operating within pre-established security limits or may stop working when resources collapse. The LMS structures all the collected information following the RDF data model as a list of events using the ECO ontology (described below). Then, all the events are sent to be analyzed by a CMA.

The LMA can also receive actions to be applied to the device. Most actions are expected to be executed via the Middleware Management Service.

2. Complex Management Agent

Complex Management Agents are more complex than LMAs and they are typically deployed on devices located in the Fog layer of an IoT system. These devices are usually advanced routers, gateways or other network devices that, due to their computing power, can provide additional services to the local network. CMAs are in charge of processing the information obtained from LMAs, either related to detected malfunctions or to any other events. A CMA consists of five components: Complex Event Processing (CEP), Detected Problems Repository (DPR), Inference Service (IS), Deployed Infrastructure Repository (DIR) and Action Schedule Service (ASS).

The Complex Events Processing component introduces the events received from LMAs into a stream of events. CEP [38] is a technology that analyses continuous streams of events to identify complex patterns. CEP systems use elements such as timestamps and sliding windows. Several filters are continuously analyzing the stream in order to identify critical operating states, which are registered in the DPR. The DIR contains information about the local network topology (e.g. connections among devices, dependencies among software, etc.). The Inference Service inserts into the DIR additional information which is inferred from the identified problems (available in the DPR) and the current infrastructure status (available in the DIR). Finally, the ASS is in charge of proposing actions with the aim of reducing the impact of the identified problems.

IV. Semantic Technologies Supporting Problem Identification

As mentioned above, we propose a solution to detect problems or undesired operating states in distributed edge-to-cloud infrastructures based on collaborative intelligent agents. Lightweight agents at the edge collect basic pieces of information (raw events) that in correlation may lead to the identification of problems or undesired operating states. In this context, information about dynamic events and the system topology (e.g. physical and/or logical connections among devices and processes running on them) has to be represented and processed. We opt for using knowledge graphs (KG) [39] to represent such information. A knowledge graph is a way of describing information in a graph structure where nodes represent entities (individuals or types of elements) and edges represent relations between them. While KGs have been used in AI for a long time (also known as *semantic networks*), they have been gaining popularity in the last years [40]. A knowledge graph is a flexible and easy-to-extend representation model, which can be endowed with a schema or ontological model (aka T-Box), thus facilitating automatic inference processes.

In the rest of this section, we first (A) present an ontology for representing the information about the topology of an edge-cloud system and the problems that may occur during its operation. Then (B), we describe how to extract and represent basic information (events) about the status of devices while the IoT system is running. Finally, we show (C) how the combination of the knowledge graph and the generated events are used to identify existing problems in the system.

A. The Edge-Cloud Ontology (ECO)

The proposed solution uses a KG and it requires advanced mechanisms to manage that KG in a viable way. The KG organizes data related to the devices connected to the IoT system, the network topology that interconnects different devices, and the software deployed on the devices. Instead of developing an ontology from scratch, we have considered reusing existing ontologies and if necessary adapting them to meet our needs. In particular, the ECO ontology [41] is appropriate for the needs of the proposed system because it provides concepts and properties to represent the state of each of the devices that make up an IoT system.

Fig. 5 shows the main concepts and properties of the ECO ontology. The ECO ontology is based on the SEAS ontology [42] and adds new entities. These new entities allow for specifying the current state of an IoT system thanks to events generated during the operation of the integrated devices. When events are processed, it is possible to identify problems or undesired operating states that are modelled into the system by the *eco:Problem* entity.

The ontology classes and properties can be organised into three main groups describing: (i) the connections among physical devices forming the network topology,(ii) the software deployed on devices and their logical dependencies, and (iii) the events representing relevant states of devices and/or software, and the problems that define critical situations. In the following, we describe the main elements of each group.

The topology of the IoT infrastructure representing computational systems and how they are interconnected can be represented with the following classes:

- *seas:System*. This class describes systems that share connections with other systems;
- *seas:Connection*. A connection describes potential interactions between systems.
- *seas:ConnectionPoint*. This class models the connection between systems.
- *eco:ComputingNode*. This class represents any device with processing capability.

Knowing which software is deployed on each device and how software components logically depend on each other can be important in certain situations in which unexpected problems (e.g. connectivity failures) on one device may affect the behaviour of others.

- *eco:Software*. This class represents any type of software and can be instantiated through three different types of subclasses: *eco:Application*, *eco:Service* and *eco:Middleware*.
- *eco:Application*. This class is a type of software that represents a particular application. An application may be composed of one or more services of type *eco:Service*.

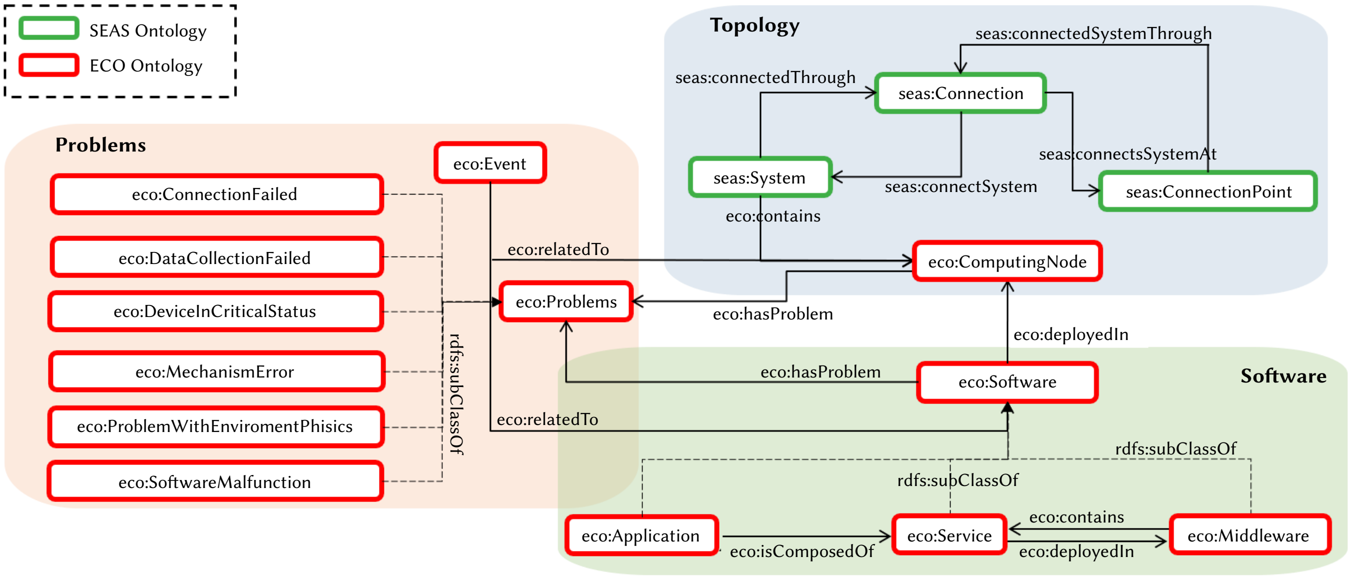


Fig. 5. Main concepts and properties of the ECO ontology. Green concepts are reused from the SEAS ontology, whereas red nodes represent new concepts defined in ECO.

- *eco:Service.* The Service class represents software services that are independent and have been designed to perform a specific function.
- *eco:Middleware.* This class is a type of Software that may contain applications or services. In practice, this class can refer to software frameworks on which applications run. An example of frameworks can be the Java virtual machine, the .NET framework, software container frameworks such as Docker, etc.

Finally, the ECO ontology allows the representation of events and problems that define critical situations.

- *eco:Problem.* This class models problems that may appear in the system. Six types of problems have been identified in this work.
- *eco:ConnectionFailed.* The class represents problems related to the connections between devices and refers to operating states related to the lack of connection between devices or software.
- *eco:DataCollectionFailed.* The DataCollectionFailed class is expected to be instantiated when a sensor has taken a measurement but the result obtained is erroneous (non-consistent value or values out of limits). The *DataCollectionFailed* class is oriented towards simple sensors measuring attributes like temperature, humidity, atmospheric pressure, etc.
- *eco:DeviceInCriticalStatus.* This class models a problem that represents a device that is in a critical state (that could stop working at any time). This situation can occur when the CPU is saturated, the available RAM memory is low, the free disk space is almost exhausted, the device battery is almost discharged or even if the device temperature is relatively high. All these situations are indicators that the device may not be working properly and may affect its performance.
- *eco:MechanismError.* This class models a problem on an actuator device, robot or any other device that has some physical mechanism. The entity models a physical operating problem. i.e. the device receives and processes the indicated actions, but cannot carry them out due to mechanical problems of the device itself.
- *eco:ProblemWithPhysicalEnviroment.* This class models real-world conditions that can adversely affect the functioning of specific devices. It is used to model physical aspects of the environment that may present a problem for specific devices. An example could be the level of luminosity of the environment in which a sensor or camera operates.
- *eco:SoftwareMalfunction.* This entity reflects problems related to the normal operation of software. It may also indicate that a software component is stopped.
- *eco:Event.* This entity represents an event that has been generated during system operation. The event contains relevant information that must be processed to evaluate whether the system is functioning correctly or whether there is an associated problem.

B. Raw Events Generation

The proposed solution requires knowing the current status of devices and software that is deployed in an IoT system. In this sense, log registers can be an important source of information since they will usually include information related to errors and operating status. There have been some efforts to standardize log files such as Common Log Format (CLF) [43], W3C Extended Log File Format[2] (ELFF), RFC5424 [44] or RFC3164 [45]. CLF and ELFF provide guidelines for organizing the information contained in log files generated by web servers and RFC5424 or RFC3164 are oriented to define the transmission of messages generated by log systems. However, there is no general standard nor guidelines for defining log messages or how to structure them. This implies that developers are responsible for designing the structure of the log messages generated by their applications. To improve this situation, we propose some indications to take into account when software has to generate log messages. Basically, we propose to specify: i) what information will be introduced in the log file. ii) how this information will be structured and iii) where the log information will be available.

Specifying the information registered in each log message is not easy. Ideally, the information should be as complete as possible because it will describe in detail why the message was inserted in the log register. However, the nature of applications is very varied and this opens up a wide range of possibilities. We propose the use of a limited set of parameters, grouped by field of application or category, as shown in tables II, III and IV.

Table II contains parameters related to the operation of applications and, whether or not an application is encapsulated on containers. The

[2] https://www.w3.org/TR/WD-logfile.html

table also presents some parameters related to the management of these containers. This information is published in the log register by the applications and by the container framework.

TABLE II. Parameters Registered by the Middleware

Params	Description
softwareIsRunnig	Whether the software is running.
actionsCarriedOut	Last command executed in the Middleware (e.g. stop, start, ...)
numContainersDeployed	Number of containers deployed in the framework.
numContainersRunning	Number of containers running in the framework.

Table III presents specific domain application parameters, usually related to perception and actuation elements (e.g., sensors or actuators). These parameters try to collect common problems that occur when applications use such elements.

TABLE III. Parameters Registered by Domain Applications. They Indicate Problems With Measurements or Actuation Orders

Params	Description
errorDescription	Error message generated by the software.
OutOfRangeReading	This error indicates that the measurement received from a sensor is outside specified limits.
abnormalReading	Measurement taken from a sensor is invalid.
highLightConditions	There is too much light in the physical environment.
lowLightConditions	There is insufficient brightness in the physical environment.
errorInPhysicalMechanism	There was a failure to activate an actuator. This error is usually associated with actuators rather than sensors, since actuators usually have physical mechanisms to interact with the physical environment.

Table IV presents a set of parameters that represent the current state of a device. Essentially, these parameters are intended to indicate the availability of resources. In our proposed solution, these values are obtained by the LMS by querying the operating system.

The proposed set of parameters is oriented to an IoT system (as shown in Fig. 1) and is intended to cover potential unexpected situations that may arise. The ECO ontology includes properties to represent those parameters.

TABLE IV. Parameters Related to Device Operation

Params	Description
connectedToTheNetwork	Whether the device is connected to a network.
droppedPackets	Number of packets discarded per second.
receivedPackets	Number of network packets received per second.
sentPackets	Number of network packets sent per second.
deviceTemperature	Device temperature.
meanPercentageUseOfCPU	Percentage of CPU usage.
batteryChargeLevel	Battery charge percentage.
freeRAM	It is the free RAM memory space. It is measured in MBs.
freeDiskSpace	It is the free disk space. It is measured in MBs.

Normally, applications record log messages to dedicated log files. However, in this paper, we propose using the log register provided by the operating systems since it is possible to access all the information about errors from a single point. Each log entry is organized into the following ordered list of attributes:

- *Time*: when a message is generated in the log register.
- *Machine*: indicates the name of the machine.
- *Application*: refers to the application, service, process, or software that generates the message
- *Message*: information regarding the event registered in the log. The content is represented as key-value pairs in JSON format.

The structure formed by the fields of each log entry facilitates the understanding of the information by humans and machines.

Table V shows an example extracted from a log register. In Linux the log register is called Syslog and contains all the log entries generated by the operating system. It has a semi-structured format where spaces separate multiple segments (timestamp, machine name, application name and message). Similarly, other operating systems have their own log systems (e.g. logcat in Android).

Line 1 has been generated by a Tomcat application server that shows, in the body of the message, information that the application has been started. Line 2 corresponds to the execution of a task that the OS had scheduled. Line 3 shows a message from the Docker (middleware) framework, structured as key-value parameters. This line specifies that a deletion task has been performed ("topic= /tasks/ delete type=event.TaskDelete") on a container ("container=ad9c..."). Line 4 presents information that has been entered by the LMA (called LMA1 in table V).

TABLE V. Parameters Related to Device Operation

	Header entry	Machine	Application	Message entry
	Time			Messages
1	Jan 17 17:11:36	machine 1	tomcat[8944]	17-Jan-2023 14:17:29.850 INFORMATION [main] org.apache.catalina.startup.Catalina.start Server startup in [560] milliseconds
2	Jan 17 17:17:01	machine 1	CRON[9184]	(root) CMD (cd / && run-parts –report /etc/cron.hourly)

3	Jan 17 17:19:39	machine 1	dockerd[751]	time="2023-01-17T17:19:39.911663169+01:00"
				level=info msg="ignoring event"
				container=ad9c3e4f269aff56c60fb3558655de1c3703be4 8b86b848ba62d8510261e8ffe
				module=libcontainerd namespace=moby topic=/tasks
				type="*events.TaskDelete"

4	Jan 17 17:20:00	machine 1	LMA1	meanPercentageUseOfCPU="7.24", freeRAM="63.4", freeDiskSpace="86.5",
				batteryChargeLevel="100.00"

5	Jan 17 17:25:00	machine 1	LMA1	errorInPhysicalMechanism

The LMA reads the log register and generates an event with the necessary information and formats it in RDF triples according to the ECO ontology. Listing 1 shows three events (event1, event2 ad event3) in RDF format[3] generated from lines 3, 4 and 5 of Table V, respectively.

Listing 1: Example of several events in RDF Turtle format

```
: event1 rdf : type eco : Event ;
     eco : date " Jan 17 17:19:39" ;
     eco : relatedToDevice : machine1 ;
     eco : relatedToSoftware : dockerd ;
     eco : actionsCarriedOutExecuted "* events . TaskDelete
     ".
: event2 rdf : type eco : Event ;
     eco : date " Jan 17 17:20:00" ;
     eco : relatedToDevice : machine1 ;
     eco : relatedToSoftware : LMA1 ;
     eco : freeRAM "63.4";
     eco : meanPercentageUseOfCP "9" ;
     eco : batteryChargeLevel "100" ;
     eco : freeDiskSpace "86.5".
: event3 rdf : type eco : Event ;
     eco : date " Jan 17 17:25:00" ;
     eco : relatedToDevice : machine1 ;
     eco : relatedToSoftware : LMA1 ;
     eco : errorInPhysicalMechanism " TRUE ".
```

In principle, ad-hoc parsers are needed for the different applications and operating systems used. However, as we mentioned above, in order to facilitate this task, we propose a *key-value* parameter representation and a set of specific parameters (Tables II, III and IV). LMAs can interpret those parameters as well as some popular application formats such as docker and tomcat. Other log formats are not considered. Thus, the systems that want to integrate our solution approach in the future must follow our proposed log format.

C. Stream Processing

Event processing is mainly carried out by the CEP component, which is responsible for analyzing the events and is able to identify undesirable situations or system problems from the information contained in an event stream. In the case of identifying any undesirable situation, the CEP is able to classify the type of situation and to label that situation with the corresponding problem entity from the ECO ontology. Detecting problems is generally a complex task that may depend on several parameters. Thus, the origin of a problem may be associated with one or several pieces of evidence. In this work, an evidence is an event; therefore, the information contained in the events helps to identify problems. One of the main characteristics of systems based on event processing is that events are not independent of each other, but are related to each other. In the case of systems formed by sensor devices, the data generated by the sensor network are usually related in time and space. For example, in agro-IoT systems, the data measured by a sensor is usually strongly related to the data of a nearby sensor. In a similar way, the measures observed at a particular moment in time are generally correlated to the measures taken in the next unit of time. This is important because applications may not be interested in measurements taken from a sensor at a particular time and place, but in aggregated information in space and time [46].

The main task of event processing is to identify within event streams those event patterns that are of interest in a particular domain. For example, in an agro-IoT system consisting of sensors to measure the conditions of cropland, several sensors may emit events that must be analyzed to discover patterns identified with problems related to plant health. In the context of the work presented in this paper, the events generated from the log messages would also be analyzed in order to discover problems related to software or devices. It will even be possible to predict problems before they actually happen such that corrective actions could be applied before a particular problem appears.

The LMA is responsible for generating the corresponding events and sending them to the CMA. Then, the CEP component located in the CMA filters the events trying to identify problems. This process is done through queries. Each implemented filter returns a problem instance according to the ECO ontology, which is inserted into the Detected Problem Repository (DPR). For example, the

DeviceInCriticalStatus problem instance could be triggered in those cases where a device has a battery below 15% in addition to having high CPU and RAM consumption. This situation could cause a device to consume its low battery power in a short period of time.

For stream processing, we use C-SPARQL [47], a continuous query language that extends SPARQL [48] to work with RDF data streams such as the example shown in listing 1. C-SPARQL queries are continuously monitoring recent events/triples (time windows are specified) to detect particular patterns that correspond to identified problems. When the query is matched, it generates a result as RDF triples.

Listing 2: C-SPARQL query that identifies a problem that a device is in a critical state (CPU and RAM usage higher than 90% and battery charge lower than 15%)

```
CONSTRUCT {
     _: prob rdf : type eco : DeviceInCriticalStatus .
     _: prob eco : relatedTo ? device .
}
FROM STREAM : streamExample1
          [ RANGE 60s STEP 30s]
FROM < instancesTopology .owl >
WHERE {
     ? event rdf : type eco : Event .
     ? event eco : relatedTo ? device .
     ? event eco : percentCPUUsage ? cpuL .
     ? event eco : freeRAM ? ram .
     ? event eco : batteryChargeLevel ? batt .
FILTER (? percentCPUUsage > 90
     && ? freeRAM < 10
     && ? batteryChargeLevel < 15)
}
```

Listing 2 shows an example of a query that returns the instance of *eco:DeviceInStatusCritical* problem, with its associated device. The query checks the value of several parameters such as CPU, RAM consumption and the battery charge level of a specific device. The query only takes into account events occurring in the specified time window (60 seconds).

V. Use Case

This section presents a use case that shows the potential of the error detection framework proposed in this work. We consider a scenario composed of two local Wi-Fi networks, each containing two sensors deployed on SmartDevices (e.g., Smartphones), a computing device (Raspberry Pi4 model B equipped with 4GB of RAM) that acts as a Fog Device and a router that provides the Wi-Fi network to which each of the devices is connected. Fig. 6 shows the example

[3] We use the Turtle RDF serialization. In short, each triple (subject predicate object) is written in a line, ending in ’.’. A ’;’ can be used to avoid repeating the same subject in consecutive lines.

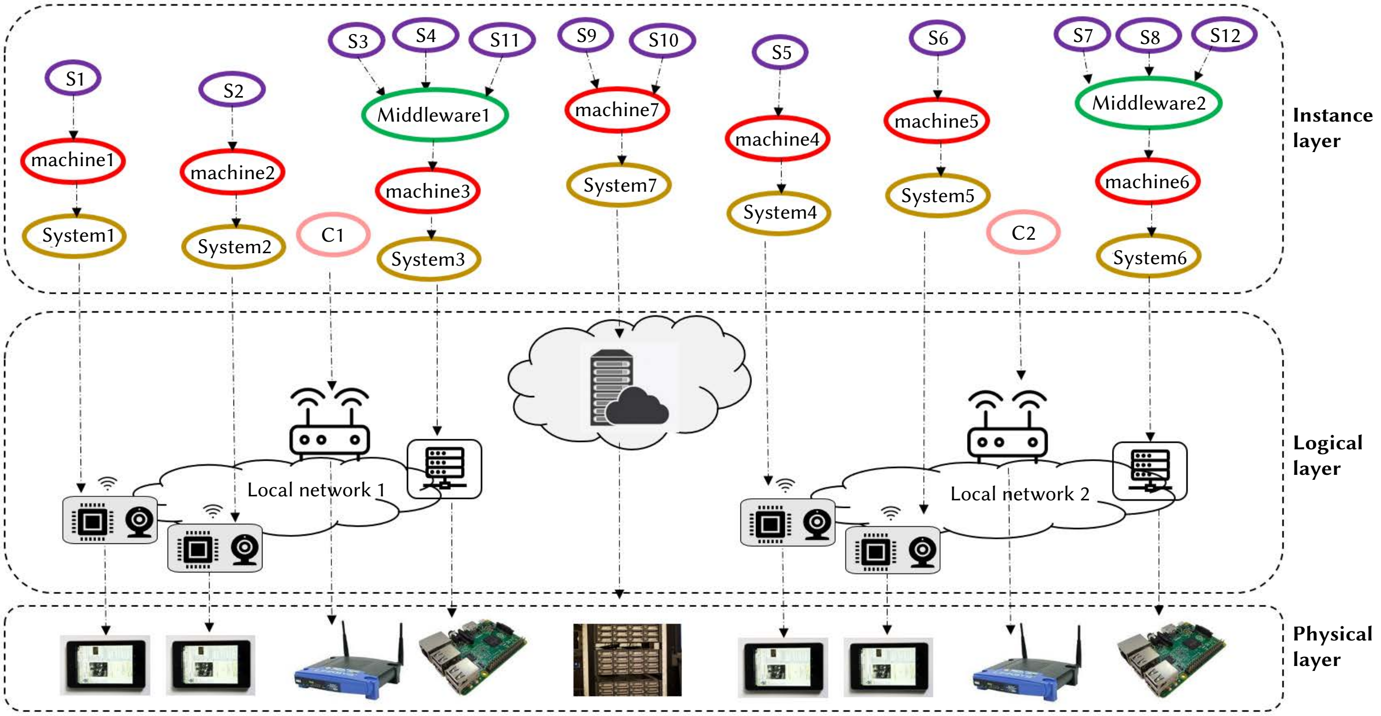


Fig. 6. Infrastructure and layout of the devices used in the use case.

infrastructure. The devices host software services and could provide data to other services. This data is processed by services to generate reports, propose actions or generate new information. According to Fig. 6, *S1*, *S2*, *S5*, *S6* retrieve data from the sensors integrated with them. *S1*, *S2*, *S5*, *S6* are distributed in geographic positions. *S1* and *S2*, located in local network 1, send data to *S4* and *S3* correspondingly (*S3*, *S4* and *S11* are part of *APP1*). *S11* sends to *S9* information processed from data received by *S3*. *S9* is hosted in the cloud (*machine7*). It unifies the information received by remote services and *S10* runs high-level tasks and provides results to end users. In the case of local network 2, the disposition of the elements and their functions are similar to local network 1.

The connections between the services as well as their relationships are shown in Fig. 7. This figure represents the knowledge graph contained in the Deployed Infrastructure Repository hosted in the CMA.

LMAs are installed on *machine1*, *machine2*, *machine4* and *machine5*. LMAs monitor the software deployed on those devices and, if necessary, generate the corresponding events. Listing 3 shows an example of events generated by LMA1, located on *machine1*, and which sends the events to CMA1 (deployed on *machine3*).

The events have parameters related to the operation of the sensor connected to *S1*. The CEP component of CMA1 (at *machine3*) continuously reads and processes the received events in order to detect abnormal situations. If a sensor is damaged, it will produce different types of errors that are reflected in the event stream. For example, if the sensor generates errors of the type *highLightConditions*, *lowLightConditions*, *abnormalReading*, *OutOfRangeReading*, etc. the reason could be that the sensor is damaged. In this case, the CEP will detect this situation and will generate a *eco:DataColletionFailed* entity. In particular, event1 indicates that service *S1* is notifying errors related to measurements taken from a sensor. Event1 has active the abnormalReading and *OutOfRangeReading* parameters, which indicate some problem with the measurement taken from the sensor. Event1 also indicates that service *S1* is generating the error and that *S1* is deployed on *machine1*.

Listing 3: Events generated by the use case shown in Fig. 7

```
: event1 rdf : type eco : Event ;
    eco : date " Jan 20 18:10:29" ;
    eco : relatedToDevice : machine1 ;
    eco : relatedToSoftware :S1 ;
    eco : abnormalReading " TRUE ";
    eco : OutOfRangeReading " TRUE ".
: event2 rdf : type eco : Event ;
    eco : date " Jan 20 18:11:41" ;
    eco : relatedToDevice : machine1 ;
    eco : relatedToSoftware :S1 ;
    eco : lowLightConditions " TRUE ".
: event3 rdf : type eco : Event ;
    eco : date " Jan 20 18:11:57" ;
    eco : relatedToDevice : machine1 ;
    eco : relatedToSoftware :S1 ;
    eco : batteryChargeLevel "10".
: event4 rdf : type eco : Event ;
    eco : date " Jan 20 18:11:41" ;
    eco : relatedToDevice : machine1 ;
    eco : relatedToSoftware :S1 ;
    eco : highLightConditions " TRUE ".
```

In addition to the detected problem of the sensor, the indications that the battery is low (10%) contained in event3 might be relevant. This event indicates that the sensor failure could be associated with the fact that the sensor does not have enough battery.

The CEP component will evaluate the events and their implications on the system depending on the information contained in the event stream. For example, Listing 4 detects the *eco:DataCollectionFailed* problem if the light conditions change between low and high in a short period of time (60s) and the reported sensor values are out of range. In that case, the *eco:DataCollectionFailed* entity is added to the Detected Problems Repository.

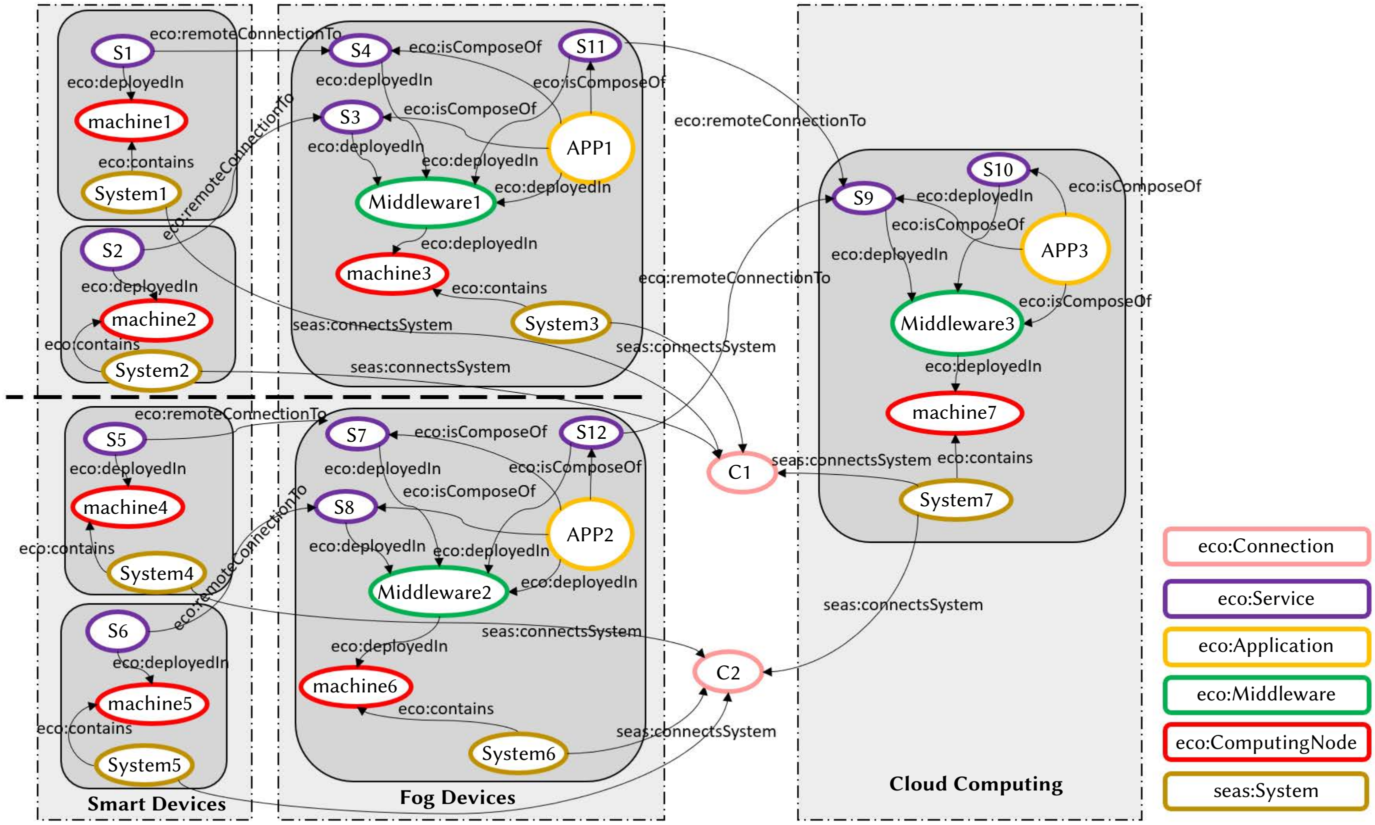


Fig. 7. Knowledge graph representing the deployed architecture. The colour of a node indicates the ontological concept (on the right side of the figure) it instantiates. E.g. S1 is an instance of eco:Service.

Listing 4: C-SPARQL query to identify data collection problems

```
CONSTRUCT {
     _: prob rdf : type eco : DataCollectionFailed .
     _: prob eco : relatedTo ? service
}
FROM STREAM
     : streamExample1
          [ RANGE 60s STEP 30s]
FROM < instancesTopology .owl >
WHERE {
     ? event rdf : type eco : Event .
     ? event eco : relatedTo ? machine .
     ? event eco : lowLightConditions " TRUE ".
     ? event eco : highLightConditions " TRUE ".
     ? event eco : OutOfRangeReading " TRUE ".
}
```

The Inference Service tries to find other elements of the IoT system that might be affected by the problems registered in the DPR. For this, the IS analyses the system architecture described in the Deployed Infrastructure Repository. As a result, the IS infers that *S4* is affected since it is connected to *S1*. Thus, the software application *APP1* is also affected. Furthermore, the problem could be propagated to *S9*, because *S11* (*S11* forms part of *APP1*) is connected to *S9* (by *eco:remoteConnecionTo* property), and this could affect APP3. The affected software components can be detected with standard SPARQL queries. Listing 5 shows an example query to retrieve the applications affected by the malfunction of *S1*.

The query considers four cases: (1) the problematic service is directly part of an application, (2) an application contains a service that is remotely connected to a problematic service (APP1 is identified), (3) an application is remotely connected to an application that contains the problematic service, and (4) an application is remotely connected to another application that includes a service connected to the problematic service (*APP3* is identified).

Listing 5: Query identifying applications affected by remote unconnected services

```
SELECT ? affectedApp
WHERE {
   ? faultServ eco :id "S1" .
   { # Case 1
   ? affectedApp eco : isComposedOf ? faultServ .}
UNION { # Case 2
   ? affectedApp eco : isComposedOf ? servAux .
   ? faultServ eco : remoteConnectionTo + ? servAux .}
UNION { # Case 3
   ? appAux eco : isComposedOf ? faultServ .
   ? appAux eco : isComposedOf ? serv .
   ? serv eco: remoteConnectionTo + ? serviceAux2 .
   ? affectedApp eco : isComposedOf
   ? serviceAux2 .}
UNION { # Case 4
   ? faultServ eco : remoteConnectionTo +
   ? serviceAux4 .
   ? appAux eco : isComposedOf ? serviceAux4 .
   ? appAux eco : isComposedOf ? serv .
   ? serv eco: remoteConnectionTo + ? servAux .
   ? affectedApp eco : isComposedOf ? servAux .}
}
```

Once a problem and its scope have been identified, the Action Schedule Service (ASS) will decide what actions should be taken to resolve or alleviate the problem. The possible solutions to problems will be context dependent. For the above case, for example, the ASS may decide whether or not to stop *APP1* and *APP3* until the problem with the sensor connected to S1 is solved. Alternatively, it may replace some of the services that are affected by the problem. For example, *S4* could be replaced by another service that does not require the information from *S1*. In this way, *APP1* could adapt to the new situation and operate consistently. If *APP1* works correctly then *APP3* would also not be affected by the problem with *S1* and would also work correctly.

The ASS is an independent component that takes information from the DIR, interprets it and proposes corrective actions. The architecture has been designed with the objective that the ASS has a low coupling, this allows to have several implementations of the ASS with different AI mechanisms facilitating to experiment with different AI techniques. The complete design of the ASS is part of our future work. In this work, we propose using the Jena[4] rule-based system, which allows easy integration of ontologies and rules. The proposed rules are activated depending on the information about the current status of the infrastructure (especially malfunctioning issues) contained in the DIR. When a rule is fired the specified actions are executed, which typically define the changes to be applied in the system.

VI. Conclusion

In this work, we have proposed a solution approach that aims at detecting and eventually resolving anomalous situations or malfunctions in IoT systems. The approach uses several mechanisms distributed on independent intelligent agents that collaborate with each other. These agents process the log registers generated by software installed in IoT devices and detect problems and malfunctions that may compromise the operation of the IoT system. The ability to understand messages contained in a log register is complex. For this reason, we propose using a list of parameters that help to identify and describe undesired situations of the elements that compose an IoT system. The Lightweight Management Agent generates events from messages contained in log registers and each event contains information about the status of an IoT device. LMAs send those events to Complex Management Agents, which process them in order to identify problems. CMAs use knowledge graphs (based on the ECO ontology) to structure the system information such as the topology, the deployed software and possible problems (undesired situations). They use this knowledge to infer new information, in particular, to identify the scope to which an identified problem affects the entire IoT system. Based on this information, corrective actions can be carried out to bring back the IoT system to a desired state. All these mechanisms provide a viable solution for the auto-maintenance of IoT systems. The proposed approach can be deployed in conjunction with third-party IoT systems since it can be adapted and integrated with existing solutions that have been designed and deployed for specific tasks.

Our work is subject to some limitations that we plan to address in future work. In a first step, we will extend the list of parameters proposed in this work. We will also apply our solution to more complex real-world environments so as to further back its versability and analyse its performance. To this end, we rely on the Mininet5 simulator for large-scale experiments. Also, in this work, we focused on problem detection. As a next step, we will concentrate on analysing the automatic execution of corrective actions to resolve detected problems (Action Schedule Service in our architecture).

[4] https://jena.apache.org

Acknowledgment

This work has been supported by grant VAE: TED2021-131295B-C33 funded by MCIN/AEI/ 10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR", by grant COSASS: PID2021-123673OB-C32 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe", and by the AGROBOTS Project of Universidad Rey Juan Carlos funded by the Community of Madrid, Spain. Iván Bernabé has been funded by the Spanish Ministry of Universities through a grant related to the Requalification of the Spanish University System 2021–23 by the University Carlos III of Madrid.

References

[1] U. Cisco, "Cisco annual internet report (2018–2023) white paper," *Cisco: San Jose, CA, USA*, vol. 10, no. 1, pp. 1–35, 2020.

[2] S. Qiu, K. Cheng, T. Zhou, R. Tahir, L. Ting, "An eeg signal recognition algorithm during epileptic seizure based on distributed edge computing," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 5, pp. 6–13, 2022, doi: 10.9781/ijimai.2022.07.001.

[3] S. Pan, X. Gu, Y. Chong, Y. Guo, "Content-based hyperspectral image compression using a multi- depth weighted map with dynamic receptive field convolution," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 5, pp. 85–92, 2022, doi: 10.9781/ijimai.2022.08.004.

[4] M. M. Ogonji, G. Okeyo, J. M. Wafula, "A survey on privacy and security of internet of things," *Computer Science Review*, vol. 38, p. 100312, 2020, doi: https://doi.org/10.1016/j.cosrev.2020.100312.

[5] H. Mrabet, S. Belguith, A. Alhomoud, A. Jemai, "A survey of iot security based on a layered architecture of sensing and data analysis," *Sensors*, vol. 20, no. 13, p. 3625, 2020.

[6] K. Gulati, R. S. K. Boddu, D. Kapila, S. L. Bangare, N. Chandnani, G. Saravanan, "A review paper on wireless sensor network techniques in internet of things (iot)," *Materials Today: Proceedings*, vol. 51, pp. 161–165, 2022.

[7] S. Rani, A. Kataria, V. Sharma, S. Ghosh, V. Karar, K. Lee, C. Choi, "Threats and corrective measures for iot security with observance of cybercrime: A survey," *Wireless communications and mobile computing*, vol. 2021, pp. 1–30, 2021.

[8] J. Seeger, A. Bröring, G. Carle, "Optimally self-healing iot choreographies," *ACM Transactions on Internet Technology (TOIT)*, vol. 20, no. 3, pp. 1–20, 2020.

[9] D. Weyns, *Software Engineering of Self-adaptive Systems*, pp. 399–443. Cham: Springer International Publishing, 2019.

[10] O. Gheibi, D. Weyns, F. Quin, "Applying machine learning in self-adaptive systems: A systematic literature review," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 15, no. 3, pp. 1–37, 2021.

[11] C. Zhu, G. Pastor, Y. Xiao, Y. Li, A. Ylae-Jaeaeski, "Fog following me: Latency and quality balanced task allocation in vehicular fog computing," in *2018 15th Annual IEEE international conference on sensing, communication, and networking (SECON)*, 2018, pp. 1– 9, IEEE.

[12] Z. Liu, X. Yang, Y. Yang, K. Wang, G. Mao, "Dats: Dispersive stable task scheduling in heterogeneous fog networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3423–3436, 2018.

[13] G. Zhang, F. Shen, N. Chen, P. Zhu, X. Dai, Y. Yang, "Dots: Delay-optimal task scheduling among voluntary nodes in fog networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3533–3544, 2018.

[14] L. Sun, Y. Li, R. A. Memon, "An open iot framework based on microservices architecture," *China Communications*, vol. 14, no. 2, pp. 154–162, 2017.

[15] A. Celesti, L. Carnevale, A. Galletta, M. Fazio, M. Villari, "A watchdog service making container- based micro-services reliable in iot clouds," in *2017 IEEE 5th international conference on future internet of Things and Cloud (fiCloud)*, 2017, pp. 372–378, IEEE.

[16] A. Krylovskiy, M. Jahn, E. Patti, "Designing a smart city internet of things platform with microservice architecture," in *2015 3rd international conference on future internet of things and cloud*, 2015, pp. 25–30, IEEE.

[17] S. He, J. Zhu, P. He, M. R. Lyu, "Experience report: System log analysis for anomaly detection," in *2016 IEEE 27th international symposium on*

software reliability engineering (ISSRE), 2016, pp. 207–218, IEEE.

[18] S. Zhang, W. Meng, J. Bu, S. Yang, Y. Liu, D. Pei, J. Xu, Y. Chen, H. Dong, X. Qu, *et al.*, "Syslog processing for switch failure diagnosis and prediction in datacenter networks," in *2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS)*, 2017, pp. 1–10, IEEE.

[19] S. Messaoudi, A. Panichella, D. Bianculli, L. Briand, R. Sasnauskas, "A search-based approach for accurate identification of log message formats," in *Proceedings of the 26th Conference on Program Comprehension*, 2018, pp. 167–177.

[20] R. Vaarandi, "A data clustering algorithm for mining patterns from event logs," in *Proceedings of the 3rd IEEE Workshop on IP Operations and Management (IPOM 2003) (IEEE Cat. No. 03EX764)*, 2003, pp. 119–126, Ieee.

[21] R. Vaarandi, M. Pihelgas, "Logcluster-a data clustering and pattern mining algorithm for event logs," in *2015 11th International conference on network and service management (CNSM)*, 2015, pp. 1–7, IEEE.

[22] N. F. Noy, D. L. McGuinness, *et al.*, "Ontology development 101: A guide to creating your first ontology," 2001.

[23] T. Bittner, M. Donnelly, S. Winter, "Ontology and semantic interoperability," in *Large-scale 3D data integration*, CRC Press, 2005, pp. 139–160.

[24] R. Jasper, M. Uschold, *et al.*, "A framework for understanding and classifying ontology applications," in *Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management KAW*, vol. 99, 1999, pp. 16–21, Citeseer.

[25] J. Agbaegbu, O. T. Arogundade, S. Misra, R. Damaševičius, "Ontologies in cloud computing—review and future directions," *Future Internet*, vol. 13, no. 12, p. 302, 2021.

[26] S. Jaskó, A. Skrop, T. Holczinger, T. Chován, J. Abonyi, "Development of manufacturing execution systems in accordance with industry 4.0 requirements: A review of standard- and ontology-based methodologies and tools," *Computers in Industry*, vol. 123, p. 103300, 2020, doi: https://doi.org/10.1016/j.compind.2020.103300.

[27] A. Heidari, N. Jafari Navimipour, "Service discovery mechanisms in cloud computing: a comprehensive and systematic literature review," *Kybernetes*, vol. 51, no. 3, pp. 952–981, 2022.

[28] M. M. Al-Sayed, H. A. Hassan, F. A. Omara, "Cloudfnf: An ontology structure for functional and non- functional features of cloud services," *Journal of Parallel and Distributed Computing*, vol. 141, pp. 143–173, 2020, doi: https://doi.org/10.1016/j.jpdc.2020.03.019.

[29] V. Singh, S. Pandey, "Cloud security ontology (cso)," *Cloud Computing for Geospatial Big Data Analytics: Intelligent Edge, Fog and Mist Computing*, pp. 81–109, 2019.

[30] F. Moscato, R. Aversa, B. Di Martino, T.-F. Fortiş, V. Munteanu, "An analysis of mosaic ontology for cloud resources annotation," in *2011 federated conference on computer science and information systems (FedCSIS)*, 2011, pp. 973–980, IEEE.

[31] K. U. Sri, M. B. Prakash, J. Deepthi, "A frame work to dropping cost in passage of cdn into hybrid cloud," *Int.J. Innov. Technol. Res*, vol. 5, no. 2, pp. 5829–5831, 2017.

[32] E. Di Nitto, G. Casale, D. Petcu, *et al.*, "On modaclouds' toolkit support for devops," in *4th European Conference on Service Oriented and Cloud Computing Workshops (ESOCC)*, 2016, pp. 430–431.

[33] K. Taylor, A. Haller, M. Lefrançois, S. J. Cox, K. Janowicz, R. Garcia-Castro, D. Le Phuoc, J. Lieberman, R. Atkinson, C. Stadler, "The semantic sensor network ontology, revamped.," in *JT@ ISWC*, 2019.

[34] L. Daniele, F. den Hartog, J. Roes, "Created in close interaction with the industry: the smart appliances reference (saref) ontology," in *Formal Ontologies Meet Industry: 7th International Workshop, FOMI 2015, Berlin, Germany, August 5, 2015, Proceedings 7*, 2015, pp. 100–112, Springer.

[35] B. Ontology, "onem2m technical specification: Ts-0012- v3.7.3.."

[36] Q.-D. Nguyen, C. Roussey, M. Poveda-Villalón, C. de Vaulx, J.-P. Chanet, "Development experience of a context-aware system for smart irrigation using caso and irrig ontologies," *Applied Sciences*, vol. 10, no. 5, p. 1803, 2020.

[37] S. R. U. Kakakhel, L. Mukkala, T. Westerlund, J. Plosila, "Virtualization at the network edge: A technology perspective," in *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, 2018, pp. 87– 92.

[38] D. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems.* Boston, MA: Addison-Wesley, 2002.

[39] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, "Knowledge graphs," *ACM Computing Surveys*, vol. 54, pp. 1–37, jul 2021, doi: 10.1145/3447772.

[40] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, J. Taylor, "Industry-scale knowledge graphs: Lessons and challenges," *Communications of the ACM*, vol. 62 (8), pp. 36–43, 2019.

[41] I. Bernabé, A. Fernández, H. Billhardt, S. Ossowski, "Towards semantic modelling of the edge-cloud continuum," in *Highlights in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation. The PAAMS Collection*, Cham, 2022, pp. 71– 82, Springer International Publishing.

[42] M. Lefrançois, J. Kalaoja, T. Ghariani, A. Zimmermann, *The SEAS Knowledge Model.* PhD dissertation, ITEA2 12004 Smart Energy Aware Systems, 2017.

[43] G. Salgueiro, V. Gurbani, A. Roach, "Format for the session initiation protocol (sip) common log format (clf)," 2013.

[44] R. Gerhards, "Rfc 5424: The syslog protocol," 2009.

[45] C. Lonvick, "The bsd syslog protocol," 2001.

[46] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, J. Widom, "A pipelined framework for online cleaning of sensor data streams," in *22nd International Conference on Data Engineering (ICDE'06)*, 2006, pp. 140–140, IEEE.

[47] D. F. Barbieri, D. Braga, S. Ceri, E. D. Valle, M. Grossniklaus, "Querying rdf streams with c- sparql," *ACM SIGMOD Record*, vol. 39, no. 1, pp. 20–26, 2010.

[48] B. DuCharme, *Learning SPARQL: querying and updating with SPARQL 1.1.* " O'Reilly Media, Inc.", 2013.

Iván Bernabé-Sánchez

Iván Bernabé-Sánchez received a degree in information technology engineering from the Carlos III University of Madrid, Leganés, Spain, in 2007 and a PhD degree in 2021. His research interests include the virtualization of devices and infrastructures defined through code, cloud and edge computing architectures, and self-configuration systems mechanisms based on knowledge representation and semantic technologies. He has participated in several nationally or internationally funded research projects.

Alberto Fernández

Alberto Fernández is a full professor at the University Rey Juan Carlos (URJC) in Madrid, where he is a member of the Artificial Intelligence Group of the CETINIA research centre. He obtained a PhD in Computer Science from the URJC. His main research lines are multi-agent systems, knowledge representation, semantic technologies, open systems, etc. He is especially interested in the application of previous technologies in domains such as intelligent transportation systems, fleet management, etc. He has participated in many national and international projects on the above topics and has published more than 80 articles in international journals, books and conferences.

Holger Billhardt

Holger Billhardt received his M.Sc. in computer science from the TH Leipzig, Germany, and his PhD in computer science at the Universidad Politécnica in Madrid. He is currently a full professor of computer science at Universidad Rey Juan Carlos in Madrid, where he is a member of the Artificial Intelligence Group at the Centre for Intelligent Information Technologies (CETINIA). His research is concerned with multi-agent systems, especially with the coordination of agents in distributed, open and dynamic environments. He is the author or co-author of more than 100 research papers and has participated in several nationally or internationally funded research projects.

Sascha Ossowski

Sascha Ossowski is a full professor of computer science and director of the CETINIA research centre at the University Rey Juan Carlos in Madrid. He received a MSc degree in informatics from U Oldenburg (Germany) and a PhD in artificial intelligence from TU Madrid (Spain). The main themes of his research refer to models and mechanisms for coordination in all sorts of agent systems and environments. He was co-founder of the European Association for Multiagent Systems (EURAMAS), chaired the European COST Action on Agreement Technologies, and is an emeritus board member of the International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).

Pollutant Time Series Analysis for Improving Air-Quality in Smart Cities

Raúl López-Blanco[1]*, Miguel Chaveinte García[2], Ricardo S. Alonso[2,3]*, Javier Prieto[1]*, Juan M. Corchado[1,2]

[1] Universidad de Salamanca, Salamanca (Spain)
[2] AIR Institute, Valladolid (Spain)
[3] Universidad Internacional de La Rioja, Logroño (Spain)

Received 31 March 2023 | Accepted 24 July 2023 | Published 27 August 2023

Abstract

The evolution towards Smart Cities is the process that many urban centers are following in their quest for efficiency, resource optimization and sustainable growth. This step forward in the continuous improvement of cities is closely linked to the quality of life they want to offer their citizens. One of the key issues that can have the greatest impact on the quality of life of all city dwellers is the quality of the air they breathe, which can lead to illnesses caused by pollutants in the air. The application of new technologies, such as the Internet of Things, Big Data and Artificial Intelligence, makes it possible to obtain increasingly abundant and accurate data on what is happening in cities, providing more information to take informed action based on scientific data. This article studies the evolution of pollutants in the main cities of Castilla y León, using Generative Additive Models (GAM), which have proven to be the most efficient for making predictions with detailed historical data and which have very strong seasonalities. The results of this study conclude that during the COVID-19 pandemic containment period, there was an overall reduction in the concentration of pollutants.

Keywords

Air Pollutants, Air Quality, Climate Change, Machine Learning, Public Health.

DOI: 10.9781/ijimai.2023.08.005

I. Introduction

The move towards Smart Cities is the evolution to which cities are tending, as they have become centres of population concentration that seek to maintain the quality of life of all their inhabitants. These increasingly overpopulated population centres. In fact, since 2008 and worldwide, there are more inhabitants in cities than in rural areas, and the trend continues to rise for cities (Fig. 1) [1].

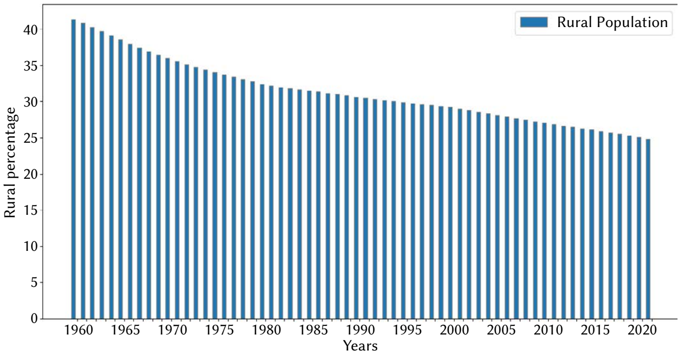


Fig. 1. Evolution of the rural population in Spain from 1960 to 2021. Source: World Bank [1].

* Corresponding author.

E-mail addresses: raullb@usal.es (R. López-Blanco), ralonso@airinstitute.com (R. S. Alonso), javierp@usal.es (J. Prieto).

People move to urban centres for many reasons, including to improve their quality of life, but maintaining the social indicators that people expect when they move to cities can be a difficult task for local, regional and other authorities involved in urban development [2]. This is partly due to some endemic problems in cities, such as air pollution, traffic (which costs €270 billion a year in Europe [3]), or the lack of green spaces (whose health benefits have been demonstrated in numerous studies [4]).

The maintenance of air quality in cities is one of the fundamental elements for the preservation and improvement of the quality of life of citizens. In fact, the World Health Organization (WHO) has a database by country that identifies the number of deaths attributable to pollution-related diseases. The WHO itself has determined that 99% of the population lives in places where the limits for pollutants suspended in the air are exceeded [5]. In Spain, for the year 2019, the estimated average percentage of deaths due to pollution was 3.32%.

A polluted environment can also influence the spread of respiratory diseases, with airborne particles acting as vectors of transmission [6] or even weakening the most vulnerable people, making them more susceptible to respiratory diseases.

At this point is where smart cities appear, seeking to improve the quality of life of citizens through the use of new technologies to achieve greater efficiency and sustainability of population services [7]. One of the main principles sought by the so-called Smart Cities is sustainability through the reduction of the environmental impact of the processes carried out in cities and the implementation of green technologies [8].

The success of the improvements introduced by Smart Cities consists of a balance between the quality of life perceived by citizens (for example, through the introduction of green areas near residential spaces [9]), the continuous actions carried out to obtain information about the environment [10] and to know which are the critical points on which action should be taken to maintain the citizens' perception of quality of life, as well as to avoid situations of eco-anxiety [11] and other disorders derived from climate change.

The set of technologies used to collect data from cities and to have more information about what is happening in them comprises a series of innovative technologies such as:

- **The Internet of Things (IoT)**. It allows to monitor the environment with different devices capable of capturing information from the surroundings, such as sensors to measure the pollutants present in the air and other magnitudes such as humidity, temperature, pressure [12].
- **Big Data**. Dealing with all the data produced by IoT devices requires a range of techniques to process and store it in the best way for later use [13].
- **Artificial Intelligence (AI)**. This discipline and its most important branches such as machine learning make it possible to create predictive models from data sets [14].
- **Blockchain**. Distributed ledger technologies such as blockchain are used in smart cities to improve the efficiency, transparency and security of data management systems and services [15].

The current work mainly combines: the Internet of Things (IoT), which are those devices or stations installed in cities and responsible for capturing data on pollutants present in the environment; Big Data, which compiles all the information obtained and makes it available to researchers to carry out this type of study; as well as Artificial Intelligence, which allows modelling what happens in the environment according to variations in the data.

All the data processing has been carried out using generative additive models (GAM) that have shown better performance than other machine learning models, such as Long Short Term Memory (LSTM) networks and Autoregressive Integrated Moving Average (ARIMA) models (used as predictive models in other works that point out that predictions with this type of networks can be improved [16]).

The rest of the article is structured as follows: Section II contains a series of related works that have carried out studies on air quality in cities and that use Artificial Intelligence models to carry them out. Section III performs a predictive and evolutionary analysis of the different pollutants found in suspension in some of the most important cities of the region of Castilla y León (Spain). Section IV gathers the most important conclusions drawn from the study of the evolution of these pollutants. Finally, Section V contains the future lines of work along which the present study could advance.

II. Related Works

This section reviews some of the most important works related to the study of air quality in different urban areas. This section deals with those works that have studied the effect of airborne pollutants in the environment by different methods and how they influence the quality of life in cities [17].

In most of the occasions, more than knowing the current situation of air quality in which a city is, it is sought through historical series [18] to know what will be the evolution in the future. All this in order to know whether the trend is upward or downward for each of the pollutants and to determine whether the corrective measures that can be applied have the expected effect.

These prediction actions are carried out by means of Machine Learning models that allow modeling the behavior of the evolution of these pollutants. Some of the most outstanding studies on pollutant evolution have been carried out using Long Short Term Memory (LSTM) networks and ARIMA models [19] and, more recently, generative additive models (GAM) [20]. Among the studies that use this type of models to predict the evolution of pollutants, those of Hasnain [21] and Shen [22] that study the evolution of pollutants in relevant cities of the Asian continent such as Seoul or regions such as Jiangsu in China, stand out.

Another area of interest on pollutants in cities is urban heat islands, areas where the temperature is significantly higher than the surrounding areas due to heat absorption and retention such as buildings and roads. Studies such as Swamy's [23] or Ngarambe's [24] have shown that heat islands can increase air pollution levels by increasing atmospheric stability and decreasing the height of the boundary layer, which limits the dispersion of pollutants.

Also important for pollutant dispersion are wind gusts, which can influence air quality by dispersing pollutants or transporting them to other areas. Studies have shown that wind gusts can influence the dispersion of fine particles in the atmosphere [25]. In addition, the presence of wind gusts can also influence the formation of pollution clouds, which can increase the levels of ozone and other pollutants in the air [26].

Air quality is directly dependent on human actions such as road traffic and industry. In fact, some studies have shown that during times of home confinement during the COVID-19 pandemic, air quality improved as virtually all air pollutants decreased [27].

From the studies reviewed, it is determined that air quality is not something that depends only on the pollutants themselves that are present in the air, but that their dispersion and concentration can be influenced by air gusts or be part of the heat island effect. The presence of these pollutants can be modeled by different Machine Learning models, being more accurate those that handle the concept of seasonality such as GAM models.

III. Experiment and Results

The experiment was based in the expansion of the task already proposed by López-Blanco et al. [20], where it was proved that the model based in the implementation of Generative Additive Model obtained better prediction results of pollutants than those obtained by LSTM and ARIMA.

To exemplify this statement, Fig. 2 and Fig. 3 demonstrate the application of LSTM recurrent networks. The main characteristic is that information can persist in the layers of the network, generating loops that allow the recall of previous states, thus creating long-term memory, which makes them ideal for learning from situations and making predictions. However, it requires data with a highly pronounced seasonality [28], which is not present in the current dataset.

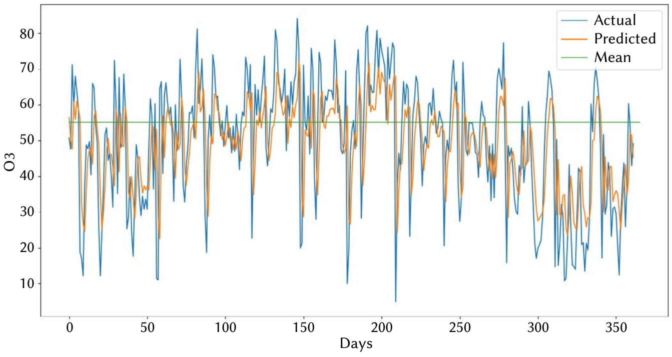


Fig. 2. Evaluation of LSTM network of pollutant O_3 in Valladolid (Spain).

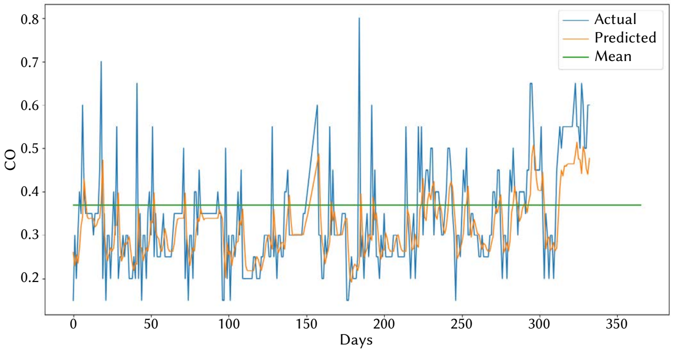


Fig. 3. Evaluation of LSTM network of pollutant CO in Valladolid (Spain).

As observed in these images, the obtained results from their evaluation do not provide predictive capability, as the networks either suffer from overfitting or impute the value of the previous day based on the considered time window.

Hence, this proposal employed the suitability of GAM as a criterion and applied it to the most populous urban areas in Castilla y León, namely: Ávila, Burgos, León, Palencia, Ponferrada, Salamanca, Segovia, Soria, Valladolid and Zamora.

Due to the previous analysis, a possible effect of the lockdown on air quality has been detected. Therefore, the spatiotemporal impact of COVID-19 lockdown measures have been evaluated in these population centers, to establish a comparison and determine the variation in atmospheric pollutant concentrations from the three years prior to the lockdown period.

A. Analysis and Forecasting Model

1. Description of the Dataset

The pollutants used in the study are: CO, NO_2, O_3 and $PM_{2.5}$. PM_{10} has also been taken into account, either as a predictor or indicator of particles, in those provinces where $PM_{2.5}$ had missing values. The CO pollutant is measured in mg/m^3, while the rest are measured in $\mu g/m^3$.

The presence of these pollutants in the air is a problem for human health, as many respiratory diseases have been shown to be caused by air pollution. Cancer of the respiratory tract is one of them caused in part by the presence of airborne PM [29].

All the pollutants studied affect human health, for example carbon monoxide (CO), produced by incomplete combustion of fossil fuels, reduces the blood's ability to carry oxygen; NO_2 and O_3 can cause airway irritation, respiratory problems and aggravation of asthma, hence the decision to include them in the study.

The dataset used contains daily concentration data of pollutants recorded at the air quality control stations of the Regional Government of Castilla y León [30]. The period of this data ranges from 1997 to 2020 (both included), during which there are certain periods of missing values in the different population centers studied. In general, PM2.5 and CO pollutants have large temporal gaps without data in most provinces, leading to various situations, which has led us to analyze each population center separately to examine their data and possible correlations between pollutants.

These facts can be observed in Fig. 4 and Fig. 5, showing the mentioned temporal evolution in the population centers of León and Ponferrada, respectively.

2. Proposed Model

For the analysis of the temporal series taking tendencies, seasonality and holidays into account, the Prophet package was used. Prophet is a tool to carry out precise and efficacious predictions, with a time of seconds to adjust the model. Equation (1) shows the expression followed by the model.

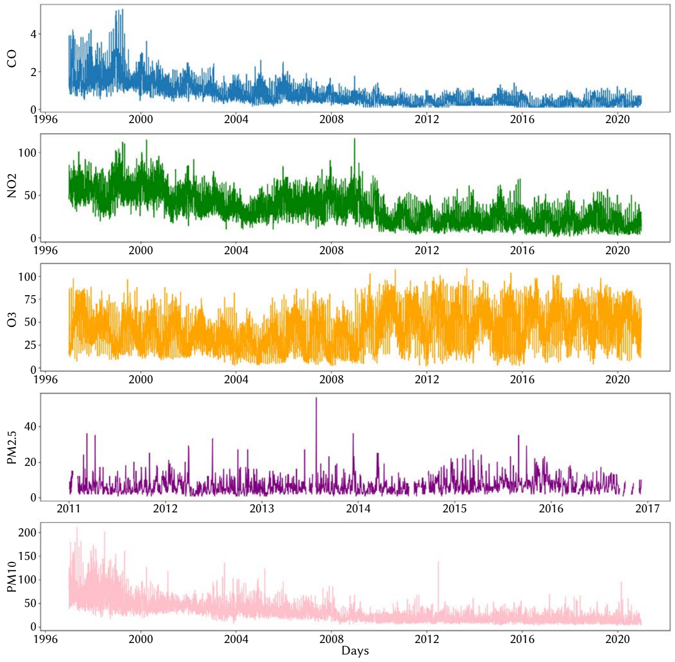


Fig. 4. Historical evolution of pollutants in the population center of León (Spain).

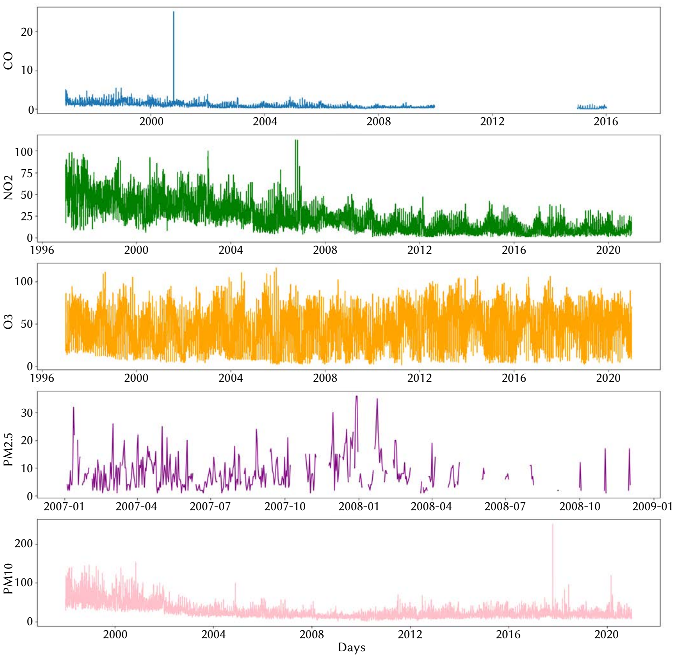


Fig. 5. Historical evolution of pollutants in the population center of Ponferrada (Spain).

$$y(t) = g(t) + s(t) + h(t) + e(t) \quad (1)$$

The assessment of the Prophet model's performance uses (1), where $y(t)$ is the predicted value determined by a linear or logistic equation; $g(t)$, as can be seen in (1) represents non-periodic changes; seasonality is given by $s(t)$, which represents periodic changes (weekly, monthly, annual); the $h(t)$ component contributes with information about holidays and events; and finally, $e(t)$ covers the noise portion of the time series, indicating random fluctuations that cannot be predicted [31].

This results in a model composed of three sub-models: the Trend model, Seasonality model, and Holidays model [32].

The trend model, called Nonlinear Saturating Growth, is represented by the logistic growth model expressed in (2).

$$g(t) = \frac{C}{1 + \exp(-k(t - m))} \tag{2}$$

where C is the maximum capacity (the maximum value of the curve), k is the growth rate (representing the "slope" of the curve), and m is an offset parameter.

The seasonality model employs Fourier series for approximations, based on (3). The seasonal component $s(t)$ provides a flexible model of periodic changes due to weekly and annual seasonality.

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right)\right) \tag{3}$$

Seasonality is key when predicting new values. Prophet offers components to plot the seasonality in weekly, annual or trendy intervals taking into account the historic series. The Fig. 6 shows the seasonality plot of the O3 concentration in the population center of Valladolid in historic, annual and weekly trends. The data of the first 10 years (2011-2020) comprised the initial training data. To validate the model, we used the data from 2020 as a test for the model and were able to compare and analyze the predicted values generated by the model with the actual values. In this process, we utilized wind velocity as a regressor variable. Finally, Prophet was used to predict the air quality for the different pollutants in the years 2021 and 2022.

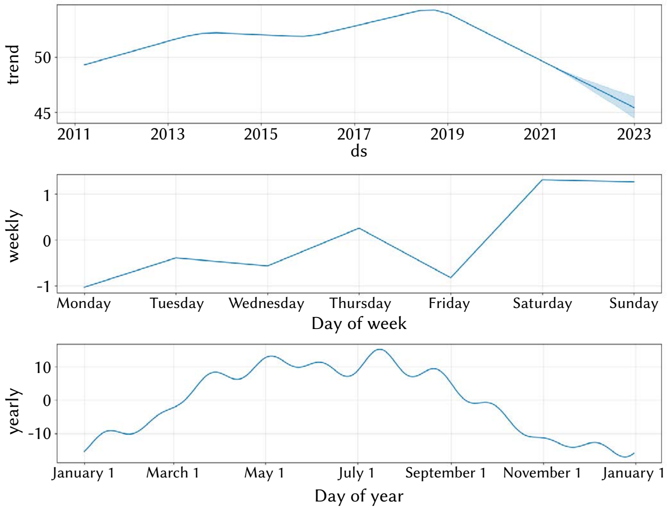


Fig. 6. Components of the Model of Valladolid (O3 trends graph, overall trend, yearly and weekly).

Therefore, this study applied GAM networks implemented in Prophet to forecast air quality. The air quality data from the Regional Government of Castilla y León [30] and the meteorological data from AEMET [33] were used as inputs. These data were preprocessed to deal with errors and missing values, using interpolation or other highly correlated pollutants as regressors. Then, the Prophet model was trained with these data to predict the values for the year 2020, incorporating wind velocity as a regressor. The predicted values were compared with the actual values, and a statistical analysis of the model performance, trend and seasonality was performed.

Finally, a two-year forecast for each pollutant was generated, examining the trends and seasonality patterns. The proposed model can be observed in Fig. 7, and its implementation in each population center and pollutant is described in Section III-A-4.

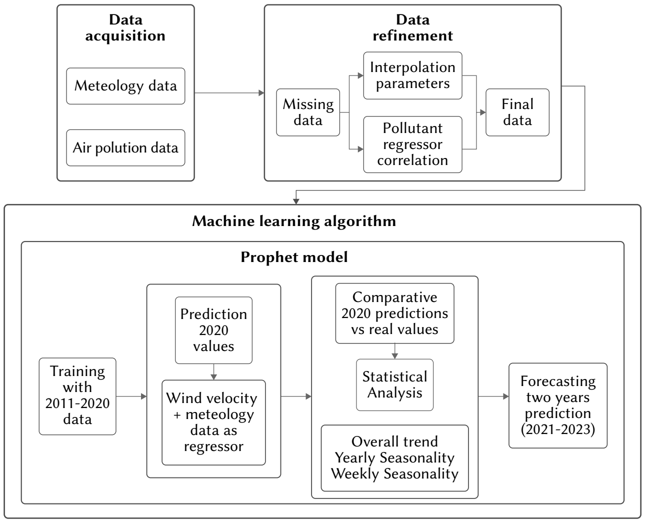

Data acquisition
Meteology data
Air polution data
Data refinement
Missing data
Interpolation parameters
Pollutant regressor correlation
Final data
Machine learning algorithm
Prophet model
Training with 2011-2020 data
Prediction 2020 values
Wind velocity + meteology data as regressor
Comparative 2020 predictions vs real values
Statistical Analysis
Overall trend Yearly Seasonality Weekly Seasonality
Forecasting two years prediction (2021-2023)

Fig. 7. The proposed architecture of prediction model for air quality.

3. Statistical Analysis

In assessing the model's efficacy, various statistical measures were computed: Pearson's correlation coefficient, mean squared error, root mean squared error, and mean absolute error. The Pearson correlation coefficient (R) was employed to ascertain if the model exhibited overfitting or underfitting. Optimal values are approximately 0.5, indicating that the model adheres to the series' overall pattern without overfitting.

The mean squared error (MSE) represents the average squared discrepancy between estimated and predicted values. The root mean squared error ($RMSE$) is the square root of MSE. The mean absolute error (MAE) is determined by averaging the differences between given and actual values. As $RMSE$ assigns greater weight to outliers compared to MAE, the disparity between the two reflects the influence of outliers within the dataset [21].

$$R = \frac{n\sum_{i=1}^{n}(x_i \hat{x}\, i) - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} \hat{x}\, i)}{\sqrt{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}\sqrt{n\sum_{i=1}^{n} \hat{x}\, i^2 - (\sum_{i=1}^{n} \hat{x}_i)^2}} \tag{4}$$

$$MSE = \sum_{i=1}^{n} (x_i - \hat{x}_i)^2 \tag{5}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |x_i - \hat{x}_i| \tag{6}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (x_i - \hat{x}_i)^2} \tag{7}$$

where x_i and $\hat{x}_i$ are the actual and predicted values respectively, and n represent the number of samples.

4. Results and Discussion

After selecting the algorithm to be employed, it is necessary to mention that it has been decided to implement the Generative Additive Model (GAM) from Prophet, which results highly convenient for the data series which encompass extense periods of detailed historic observations, with pronounced seasonalities which involve previously identified relevant, although irregular elements, as well as data points with significative outliers whose non linear growth trends approach a limit.

The prediction of the temporal series can be observed in Fig. 8, along with the seasonality in Fig. 6, and later in the detailed analysis which is explained after each pollutant. To carry out this prediction and due to the casuistics of the previously commented data, it has been opted to use the NO_2 as additional regressor to predict the missing values in the $PM_{2.5}$ and CO series that had a strong linear correlation (Pearson correlation coefficient). This method has been carried out in the population center of Salamanca (0.71 and 0.59, respectively) and León (0.54 and 0.73, respectively).

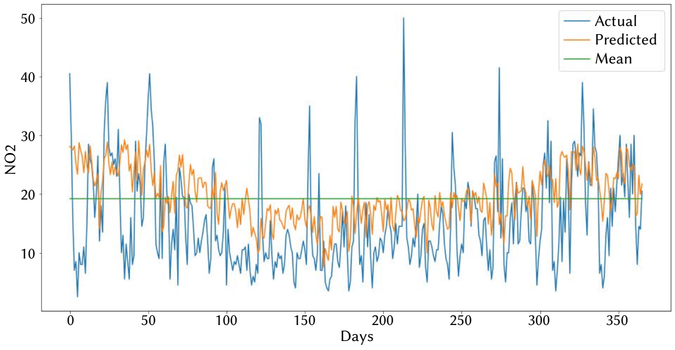


Fig. 8. NO_2 $(\mu g/m^3)$ forecasting vs. real values 2020 in León (Spain).

In Valladolid it was only used for the pollutant $PM_{2.5}$ (0.65).

In Burgos and Ponferrada, the PM_{10} was previously used as regressor for the $PM_{2.5}$ due to its high correlation (0.75 and 0.88 respectively). Subsequently in Burgos (0.77 and 0.76 respectively) the previous method was used for the prediction of the $PM_{2.5}$ and CO.

In Soria, Zamora, Palencia and Segovia the same method is used, but working with the PM_{10} since there is not enough data to predict the $PM_{2.5}$.

In Ávila, for the CO pollutant there are neither data or ways to correlate them to make a prediction. And for the particle analysis the PM_{10} was used due to the $PM_{2.5}$ not having enough data.

In constructing the models, various seasonalities have been employed: weekly and annual. This is done to account for the impact of predefined Spanish holidays and the influence of weekly traffic patterns, which are higher on weekdays and lower on weekends. With these considerations, a one-year prediction is made; and for this purpose, from the initial dataset,we retain the part of the historical series with the least noise and the longest length possible. Thus, we truncate the data's beginning, as seen in Fig. 4 and Fig. 5, which exhibits more noise and has a higher value difference compared to more recent data. Consequently, the first 14 years for NO_2 and O_3 and the first 5 years for CO and particle analysis, whether

$PM_{2.5}$ or PM_{10} in provinces, have been removed due to the aforementioned casuistry.

Within each pollutant, the followed seasonality will be examined in detail. To verify the goodness of the model's performance, the statistics mentioned earlier in Section 3 have been utilized.

NO_2

The results of the analysis are shown in Table I, where the Pearson correlation coefficient, the *MSE*, the *RMSE* and the *MAE* can be seen. These values denote the appropriateness of the model in fitting the historical data, as well as the accuracy of predictions; for instance, Salamanca exhibits both the highest Pearson correlation coefficient (0.68) and the lowest RMSE (3.46$\mu g/m^3$) which implies that its model best catches the overall trend and the lowest forecast error. On the other hand, Soria has the lowest Pearson correlation coefficient (0.37) as well as the highest RMSE (10.12μg/m3), which means that its model has the worst fit to the general trend and the highest forecast error as well as a wide confidence interval. This results might be due to Soria's series having more noise, more variability or more external factors affecting its behavior.

TABLE I. NO_2 2020 Model Performance Statistics for the Different Population Centers

Pop. Center	R	MSE	RMSE	MAE
Ávila	0.43	19.33	4.40	3.13
Burgos	0.37	38.56	6.21	4.66
León	0.48	72.13	8.49	6.93
Palencia	0.38	28.33	5.32	4.16
Ponferrada	0.48	19.51	4.42	3.03
Salamanca	0.68	11.98	3.46	2.74
Segovia	0.36	23.92	4.89	3.72
Soria	0.37	102.40	10.12	8.45
Valladolid	0.41	58.91	7.68	6.19
Zamora	0.47	31.84	5.64	4.22

In addition to the statistical data from conducting the proposed analysis, it has been decided to plot a comparison between the actual values and the predicted values in the year 2020 and analyze the behavior of the model visually.

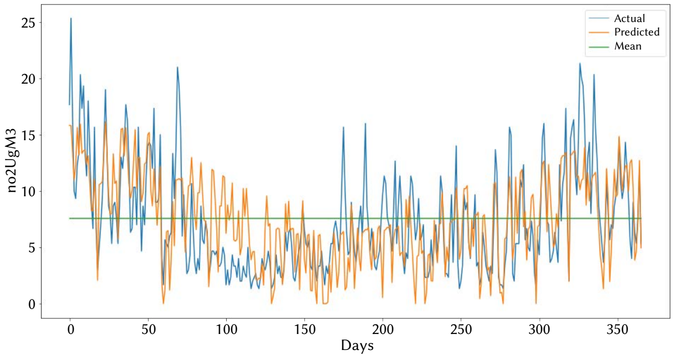


Fig. 9. NO_2 $(\mu g/m^3)$ forecasting vs. real values 2020 in Salamanca (Spain).

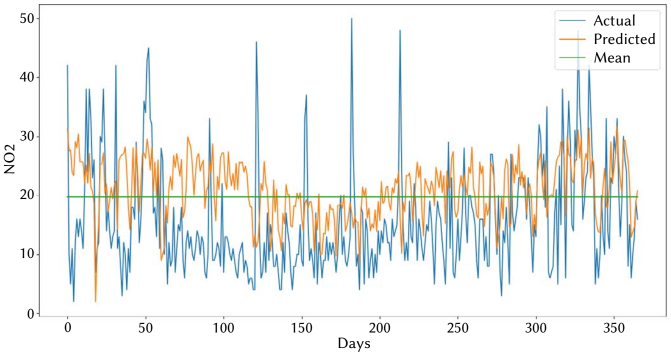


Fig. 10. NO_2 $(\mu g/m^3)$ forecasting vs. real values 2020 in Soria (Spain).

We can see in Fig. 8, Fig. 9 and Fig. 10 that the adjustment in the examples is good, even predicting peaks in the series; which confirms the statistical values of this pollutant discussed. We can also observe that between the periods of March and May 2020, the predicted values remain above the actual values, which is a general fact in all the analyzed population centers.

Finally, we performed a two-year prediction in which the prediction is displayed alongside the error margins, where the black points represent the actual values, the dark blue trendline is the temporal pattern that the model learns from and uses for predictions, and the lighter blue areas represent the error margins of the two-year prediction, where the actual values are no longer shown.

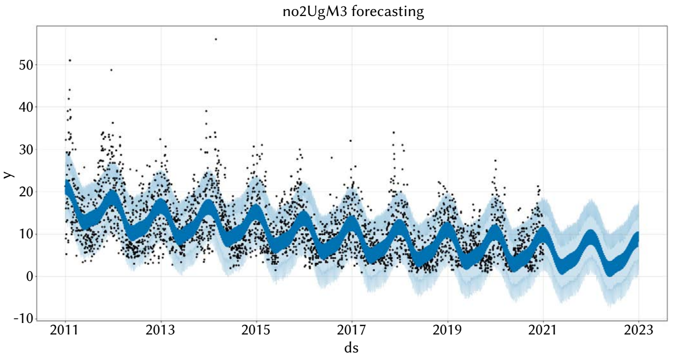


Fig. 11. NO_2 ($\mu g/m^3$) two years prediction in Salamanca (Spain).

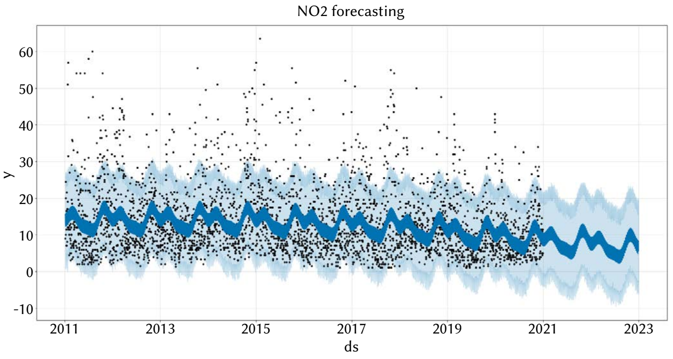


Fig. 12. NO_2 ($\mu g/m^3$) two years prediction in Burgos (Spain).

In Fig. 11 and Fig. 12 we can verify how the statistical results translate into the prediction, the trend of the series and the possible outliers, and their effect on the model.

Along with this, we obtain the components of the model, in which seasonality plays a prominent role, as discussed in the model (2).

In all the analyzed population centers, the trend observed in Fig. 13 is followed, which is exemplified by Valladolid. In this figure, we can see how the trend in recent years for NO_2 concentration is decreasing, and the prediction is that it will continue this pattern in the coming years. Weekly, it follows a stable pattern during workdays, declining on weekends. As for the annual trend, it experiences a decrease from March to August, with the highest values occurring during the first and third four-month period of the year.

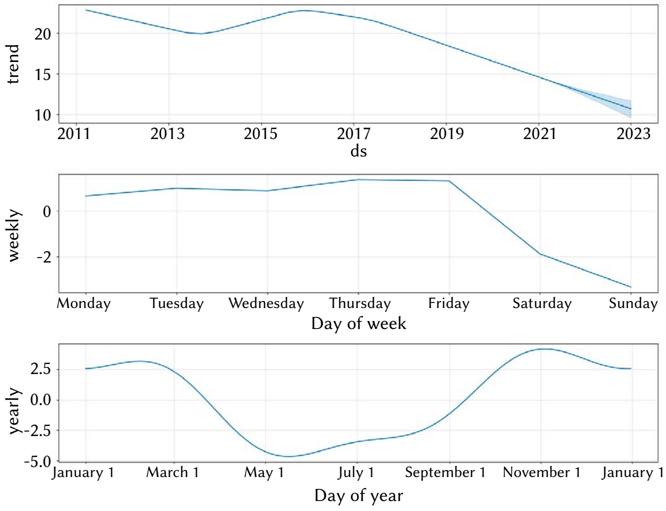


Fig. 13. Components of the Model of Valladolid (NO_2 trends graph, overall trend, yearly and weekly).

$PM_{2.5}$

From the results in Table II corresponding to the analysis of the $PM_{2.5}$, in the population centers where data or correlation was available, the following conclusions can be drawn:

TABLE II. $PM_{2.5}$ 2020 Model Performance Statistics for the Different Population Centers

Pop. Center	R	MSE	RMSE	MAE
Burgos	0.25	6.69	2.59	1.98
León	0.40	6.25	2.50	2.04
Ponferrada	0.17	89.51	9.46	5.50
Salamanca	0.30	12.45	3.53	2.65
Valladolid	0.24	23.02	4.80	3.90

The effectiveness of the prediction models varies between different population areas, as demonstrated by the *R*, *MSE*, *RMSE*, and *MAE* values. In general terms, the $PM_{2.5}$ prediction models show variable performance in different population areas, with a moderate fit in most cases. This suggests that the models capture the general trend in $PM_{2.5}$ pollution levels but are not overfitted. Greater prediction accuracy is observed in areas such as León, while in areas like Ponferrada, the model's performance is lower. Pearson correlation coefficients (R) range between 0.17 and 0.40, suggesting that there is some variability in the quality of the predictions between different population areas.

It is observed that some areas, such as León, have a higher Pearson correlation coefficient (0.40) and a lower error (*RMSE* of 2.50), indicating that the prediction model is more accurate in these areas (Fig. 14). On the other hand, areas like Ponferrada show a lower correlation coefficient (0.17) and a higher error (*RMSE* of 9.46), suggesting a lower performance of the model in this area (Fig. 15).

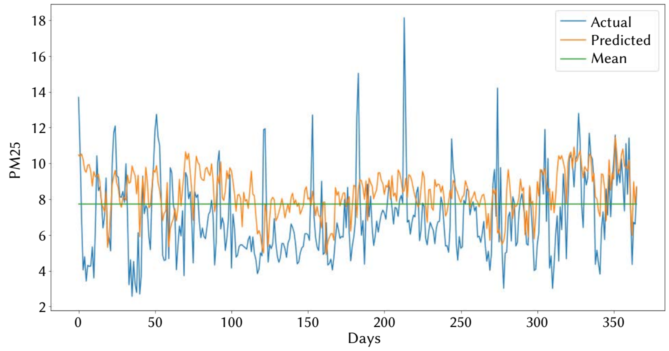


Fig. 14. $PM_{2.5}$ ($\mu g/m^3$) forecasting vs. real values 2020 in León (Spain).

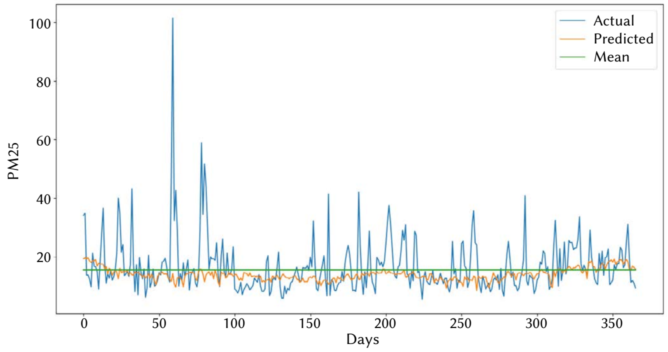


Fig. 15. $PM_{2.5}$ ($\mu g/m^3$) forecasting vs. real values 2020 in Ponferrada (Spain).

In Fig. 16 and Fig. 17, we observe the two-year predictions for these pollutants in the cases of Burgos and León respectively. We see that outliers appear, but as observed in the statistical analysis, the behavior in León is superior, adjusting to the stationary trend.

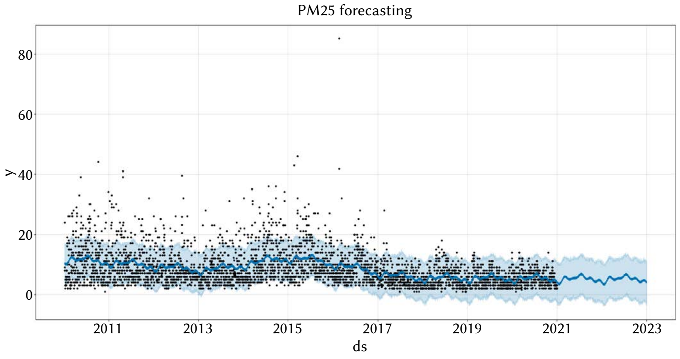


Fig. 16. $PM_{2.5}$ ($\mu g/m^3$) two years prediction in Burgos (Spain).

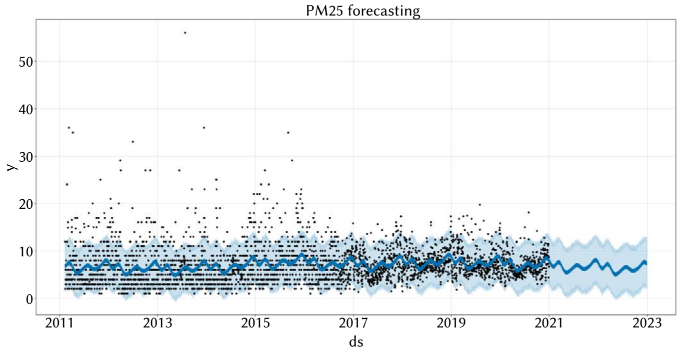


Fig. 17. $PM_{2.5}$ ($\mu g/m^3$) two years prediction in León (Spain).

Regarding the trend followed by this pollutant (Fig. 18), a decrease is observed in relation to recent years, predicting that this pattern will continue in the coming years. Weekly, it reaches its maximum peak during the middle of the week, being lower during the first and last days of the week. The same occurs inversely in the annual trend, with the first quarter and the last four months of the year being the highest points, and oscillating during the second quarter at the lowest values.

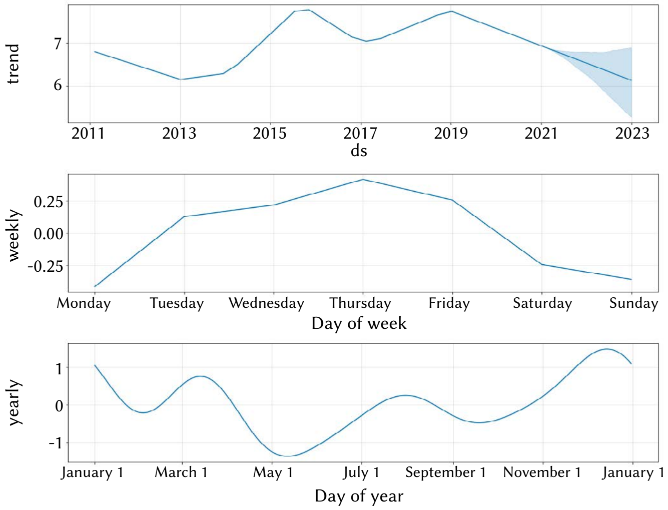


Fig. 18. Components of the Model of León ($PM_{2.5}$ trends graph, overall trend, yearly and weekly).

PM_{10}

The PM_{10} prediction models also show variable performance among the analyzed population areas (where $PM_{2.5}$ analysis was not possible), with a moderate fit in most cases. This indicates that the models capture the general trend in PM_{10} pollution levels without being overfitted. Differences in *R*, *RMSE*, and *MAE* values between population areas suggest variability in the quality of predictions across different areas. Soria serves as an example of the best-performing case. In Soria, the PM_{10} prediction model exhibits a Pearson correlation coefficient (*R*) of 0.29, an *RMSE* of 8.63, and an *MAE* of 5.96. These values indicate a moderate fit and acceptable performance in predicting pollution levels in this area. This contrasts with Segovia, where the PM10 prediction model displays a Pearson correlation coefficient (*R*) of 0.23, an *RMSE* of 10.65, and an *MAE* of 7.33. Although the model's fit is moderate, its performance is inferior compared to the case of Soria. The lower correlation and higher error values indicate that the model may not be as accurate in predicting PM10 pollution levels in Segovia (Table III).

TABLE III. PM_{10} 2020 Model Performance Statistics for the Different Population Centers

Pop. Center	R	MSE	RMSE	MAE
Ávila	0.28	135.23	11.63	6.26
Palencia	0.27	82.71	9.09	5.99
Segovia	0.23	113.32	10.65	7.33
Soria	0.29	74.51	8.63	5.96
Zamora	0.27	66.23	8.14	5.13

PM_{10} exhibits a moderate and variable adjustment, similar to $PM_{2.5}$, depending on the analyzed population center. The model's prediction for 2020 is shown in Fig. 19 and Fig. 20, taking the cases of Soria and Segovia, respectively. We see that, like $PM_{2.5}$, the adjustment is highly sensitive to possible outliers and changes in peaks reached by the series. This can be explained by the fact that NO_2 does not work well as a regressor for filling missing values in the series in all population centers.

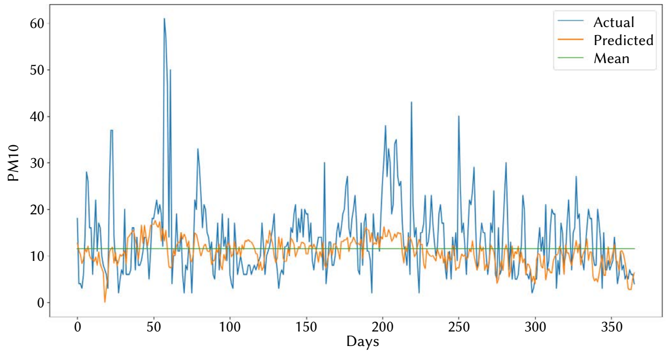


Fig. 19. PM_{10} ($\mu g/m^3$) forecasting vs. real values 2020 in Soria (Spain).

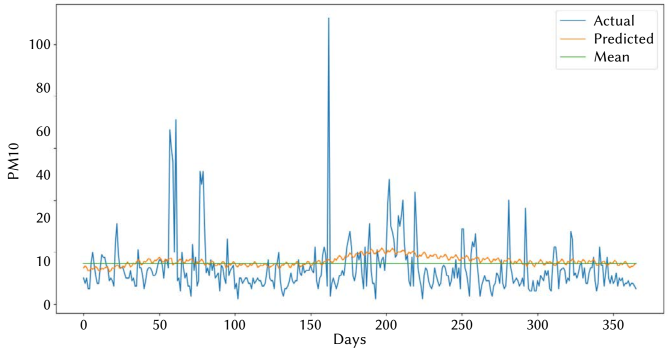


Fig. 20. PM_{10} ($\mu g/m^3$) forecasting vs. real values 2020 in Segovia (Spain).

Similarly, the two-year prediction and its adjustment to the trend and the impact of outliers are presented in Fig. 21 and Fig. 22.

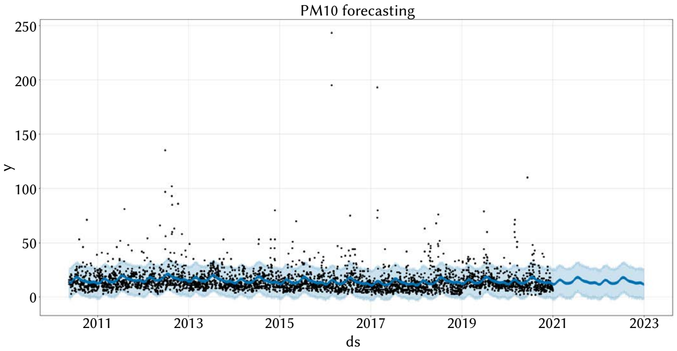


Fig. 21. PM$_{10}$ ($\mu g/m^3$) two years prediction in Segovia (Spain).

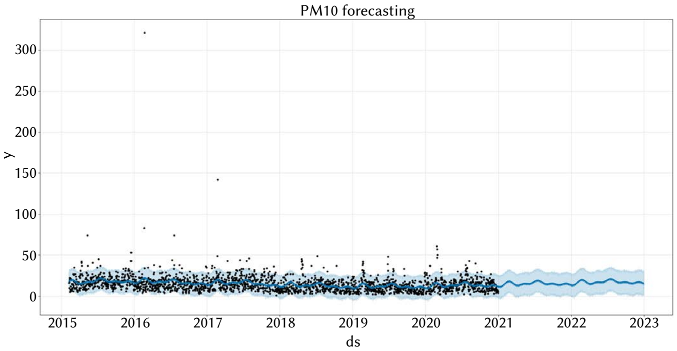


Fig. 22. PM$_{10}$ ($\mu g/m^3$) two years prediction in Soria (Spain).

The trend followed in the urban centers where PM$_{10}$ has been analyzed shows an upward tendency in recent years, which is expected to continue growing (Fig. 23). On a weekly basis, it follows the pattern of the highest values during workdays, decreasing to the minimum values on weekends. Annually, the stationary trend oscillates throughout the year, reaching maximums in March and in July-August, and decreasing to minimums at the end of the year.

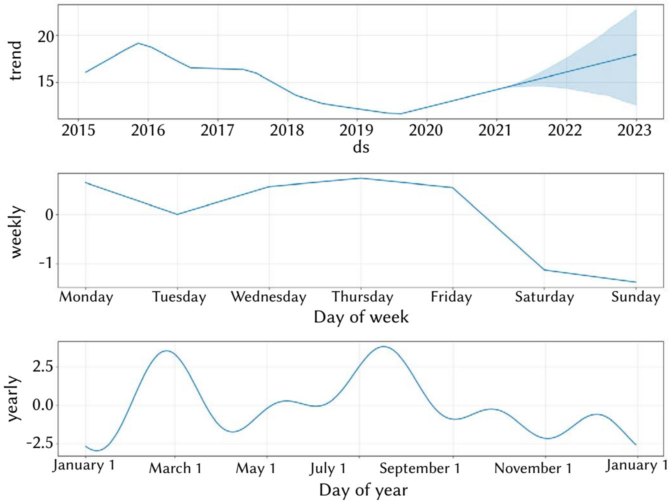


Fig. 23. Components of the Model of Soria (PM$_{10}$ trends graph, overall trend, yearly and weekly).

O$_3$

The performance of time series prediction models for the O$_3$ pollutant varies depending on the population centers under consideration, as evidenced by the calculated evaluation measures (Table IV). It is noted that Salamanca exhibits the best model fit, with high accuracy and low variability, followed by Burgos and Segovia, which display low-to-medium accuracy and low-to-medium variability. In contrast, Zamora presents the worst model fit, with moderate-to-high accuracy and very high variability, followed by Palencia and Ponferrada, which demonstrate moderate-to-low accuracy and high variability. The remaining population centers show intermediate values between these two groups. These differences can be attributed to various factors that influence the nature of the time series for each city, such as data quality, seasonality, the cyclical component, complexity, and heterogeneity.

TABLE IV. O$_3$ 2020 Model Performance Statistics for the Different Population Centers

Pop. Center	R	MSE	RMSE	MAE
Ávila	0.45	277.98	16.67	12.69
Burgos	0.36	221.97	14.90	11.48
León	0.46	266.98	16.34	12.26
Palencia	0.43	312.71	17.68	13.32
Ponferrada	0.47	288.22	16.98	13.50
Salamanca	0.61	175.88	13.26	10.44
Segovia	0.47	231.92	15.23	11.91
Soria	0.42	214.86	14.66	11.18
Valladolid	0.49	278.60	16.91	12.69
Zamora	0.49	335.15	18.31	14.40

In Fig. 24 and Fig. 25, we can observe the model fit in the 2020 prediction alongside the actual values. Salamanca and Zamora are shown, as previously mentioned, as examples of the results of the statistical analysis. In these images, it is demonstrated that Salamanca's fit is better, even successfully predicting maximum peaks accurately.

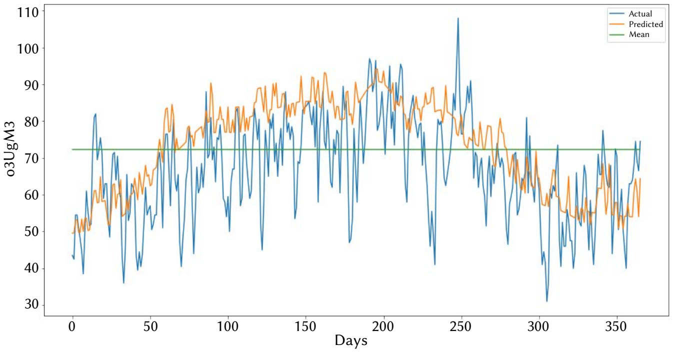


Fig. 24. O$_3$ ($\mu g/m^3$) forecasting vs real values 2020 in Salamanca (Spain).

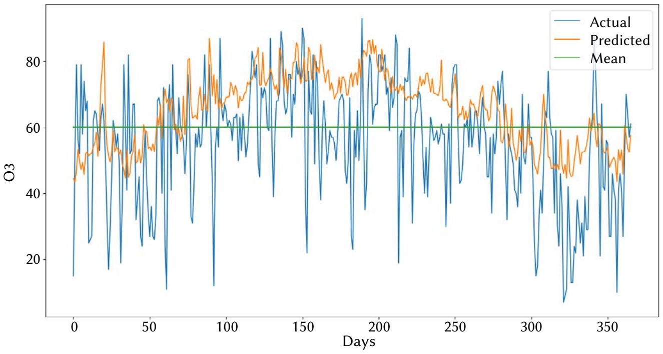


Fig. 25. O$_3$ ($\mu g/m^3$) forecasting vs real values 2020 in Zamora (Spain).

Additionally, the two-year predictions for the other pollutants are included. In Fig. 26 and Fig. 27, we can see in more depth the trend fit and the differences in the model fit between both provinces, and the reason for the variability detected with the higher RMSE in Zamora, due to the presence of a larger number of outliers resulting in a slightly worse prediction fit.

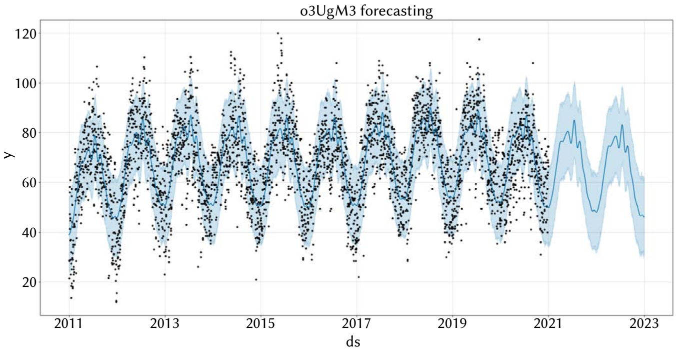


Fig. 26. O$_3$ ($\mu g/m^3$) two years prediction in Salamanca (Spain).

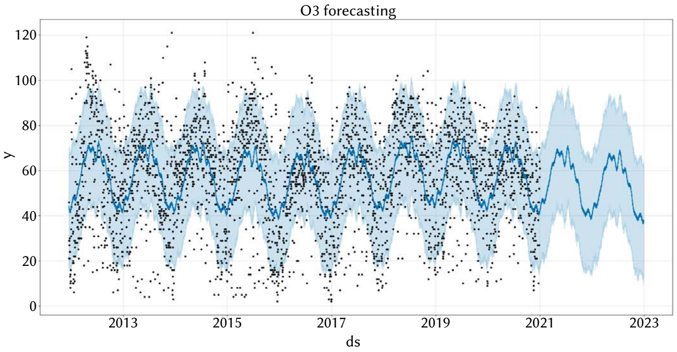


Fig. 27. O$_3$ ($\mu g/m^3$) two years prediction in Zamora (Spain).

Regarding the stationary trend of the O$_3$ pollutant, the graphs in Fig. 6 and Fig. 28 of Valladolid and Segovia urban centers are included (as previously mentioned, one urban center is used as an example of the general pattern). In these figures, it can be seen that since 2019, the trend has been decreasing and is expected to continue. Weekly, a pattern similar to that presented by NO$_2$ is found, with an increase on weekends. Moreover, in O$_3$, there is a certain midweek peak. Annually, it is observed that the highest values are reached in the months of the second quarter of the year, while the lowest values occur in the rest of the quarters.

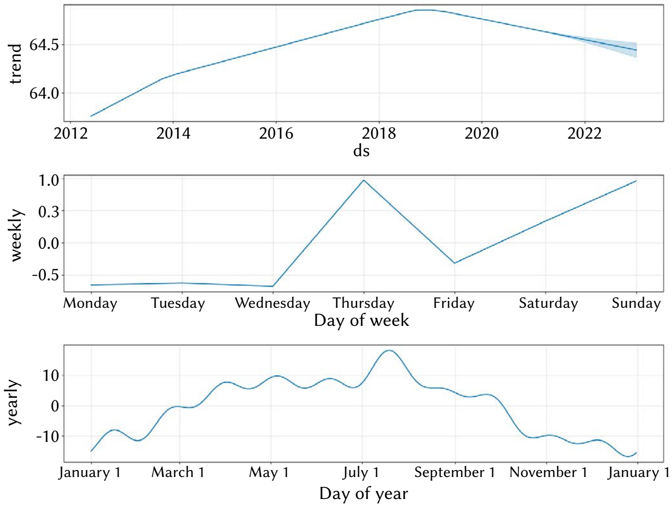


Fig. 28. Components of the Model of Segovia (O$_3$ trends graph, overall trend, yearly and weekly).

CO

Upon analyzing the results Table V corresponding to the CO pollutant in different population centers, we can draw several conclusions. Firstly, it can be seen that in Avila there are no adequate data or correlations available to predict CO levels. Regarding the performance of the models in other areas, significant variations are noticed in terms of Pearson correlation coefficient (*R*), *MSE*, *RMSE*, and *MAE*.

TABLE V. CO 2020 Model Performance Statistics for the Different Population Centers

Pop. Center	R	MSE	RMSE	MAE
Ávila	-	-	-	-
Burgos	0.06	0.03	0.17	0.14
León	0.25	0.05	0.22	0.19
Palencia	0.01	0.01	0.08	0.07
Ponferrada	0.88	0.01	0.07	0.05
Salamanca	0.17	0.06	0.24	0.19
Segovia	0.09	0.02	0.13	0.11
Soria	0.28	0.005	0.07	0.05
Valladolid	0.02	0.02	0.16	0.13
Zamora	0.77	0.005	0.07	0.06

For example, in Ponferrada and Zamora, the models seem to be overfitted, as they exhibit very high Pearson correlation coefficients (0.88 and 0.77, respectively). This could be due to the use of NO2 as a regressor for the missing CO values in their respective series.

In other areas, such as Burgos, León, Palencia, Salamanca, Segovia, Soria, and Valladolid, the results vary in terms of fit and accuracy. Some areas like León and Soria (Fig. 29 and Fig. 30) show moderate correlation coefficients (0.25 and 0.28, respectively), while others such as Palencia and Valladolid display very low correlations (0.01 and 0.02, respectively).

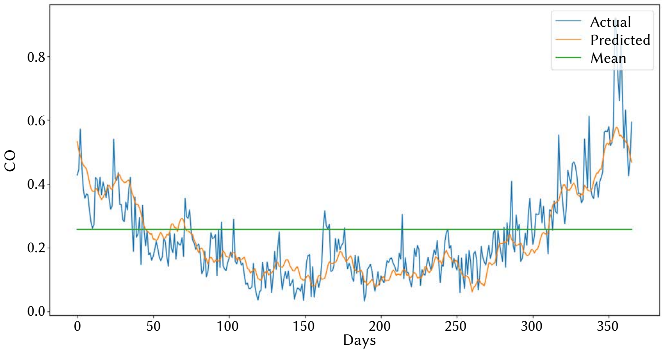


Fig. 29. CO$_3$ (mg/m^3) forecasting vs. real values 2020 in Ponferrada (Spain).

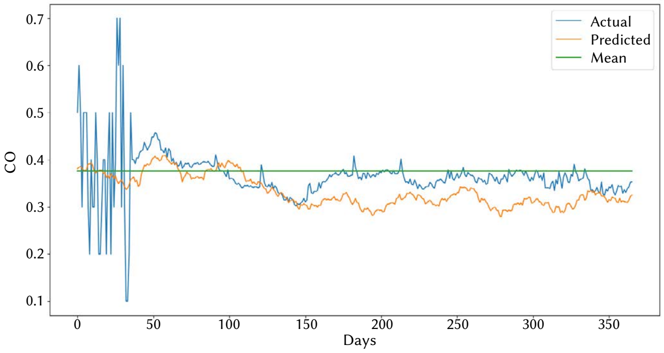


Fig. 30. CO$_3$ (mg/m^3) forecasting vs. real values 2020 in Soria (Spain).

This effect is even more pronounced in the two-year prediction, which partly explains the obtained statistical values. In Fig. 31, it can be seen how the model follows the series trend and is capable of approximating the periods with missing values since it has enough data and does not present a large number of outliers that might confuse it. Meanwhile, in Fig. 32, the prediction is not entirely accurate due to the large number of outliers and some periods in the series that lack of data.

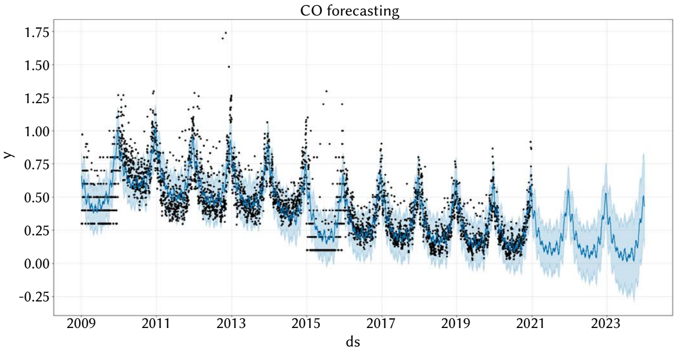


Fig. 31. CO_3 (mg/m^3) two years prediction in Ponferrada (Spain).

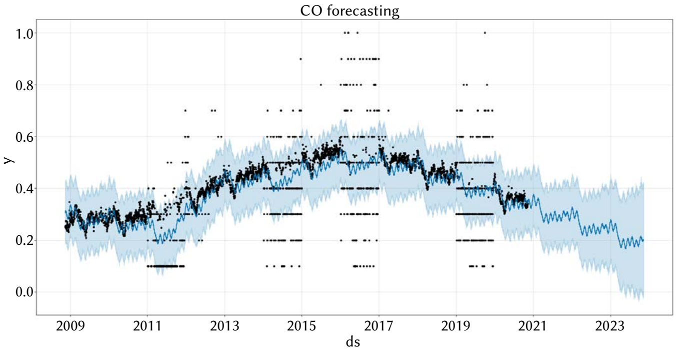


Fig. 32. CO_3 (mg/m^3) two years prediction in Soria (Spain).

In summary, the table results indicate that the models used to predict CO levels in different population centers present variations in their performance and accuracy. These variations can be attributed to differences in modeling approaches, data quality, and correlations between the pollutants used as regressors. In future research, it would be advantageous to investigate alternative modeling approaches and additional factors, such as wind gusts or the so-called heat island effect, with the aim of enhancing the accuracy of CO predictions in these population areas.

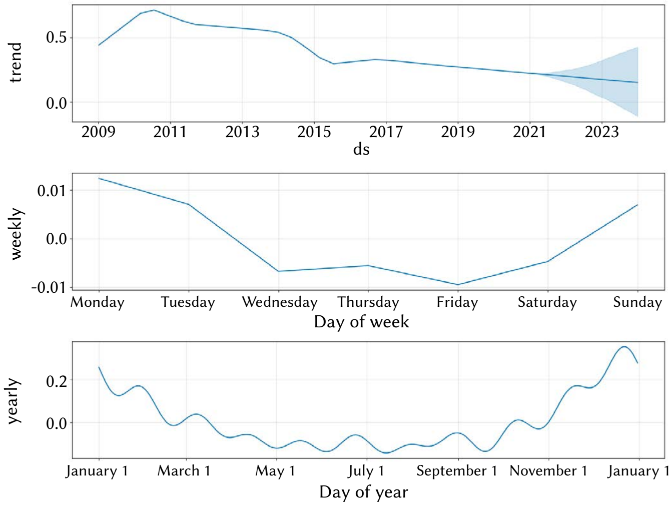


Fig. 33. Components of the Model of Ponferrada (CO trends graph, overall trend, yearly and weekly).

Finally, regarding the trend, the study of the stationary trend is presented, specifically for Ponferrada, but generalizing to the rest of the urban centers. In Fig. 33, it can be observed that there has been a downward trend in recent years, although it has slowed down and stagnated. Weekly, during the weekend, the values increase, reaching the highest peak between Sunday and Monday, and then decreasing after Monday, taking the lowest values between Wednesday and Friday. Annually, it can be observed that the trend begins to increase from October until the end of the year. During the first two quarters of the year, the trend is decreasing.

B. COVID-19's Impact on Air Quality

As analyzed in the one-year predictions in Section 4, the predicted value of pollutants in general, although particularly notable in NO2, is higher than the actual values between the periods of March and May 2020, while it adjusts for the rest of the year's prediction, even to the highest peaks. According to numerous studies, a sudden decrease in pollutant concentrations has been observed worldwide: Malaysia [34], northern China [35] and Brescia (Lombardy) [36].

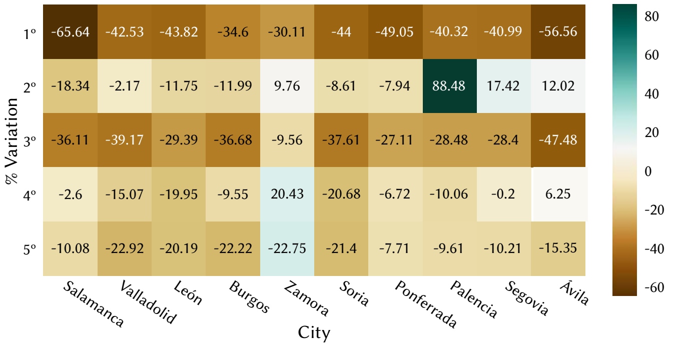
This has led us to investigate this period in depth and how it fits within the historical time series of each population centers. To this end, in this part of the research, we partitioned the time series data into six distinct periods for analysis. The pre-lockdown phase spanned from December 1, 2019, to March 13, 2020 (103 days), while the lockdown period extended from March 14, 2020, to June 21, 2020 (99 days). The post-lockdown phase occurred between June 22, 2020, and September 30, 2020 (100 days). Additionally, we included three comparative periods (P4-P6), which corresponded to the same lockdown dates in the years 2017, 2018 and 2019.

The results are shown by pollutant with their respective spatiotemporal variations in each population centers. To perform the analysis, with the mentioned dates, they have been combined to provide a perspective on air quality during the lockdown period. The following variations (in%) were considered in averaged over the periods detailed below(the order followed is important, as it corresponds to the row number of the variation in the heatmap):

1. Variation between the lockdown period and the pre-lockdown period.
2. Variation between the post-lockdown period and the period ranging from the beginning of the pre-lockdown to the end of the lockdown.
3. Variation between the 2020 lockdown period and the average of the same dates in 2017, 2018, and 2019.
4. Variation between the average of the entire year 2020 and the average of the entire year 2019.
5. Variation between the average of the entire year 2020 and the average of the years 2017, 2018, and 2019.

NO_2

As previously discussed, one of the most notable effects of this decrease occurs in NO_2. It has been decided to display the analysis of the different proposed variations in a heatmap, as shown in Fig. 34. In



	Salamanca	Valladolid	León	Burgos	Zamora	Soria	Ponferrada	Palencia	Segovia	Ávila
1º	-65.64	-42.53	-43.82	-34.6	-30.11	-44	-49.05	-40.32	-40.99	-56.56
2º	-18.34	-2.17	-11.75	-11.99	9.76	-8.61	-7.94	88.48	17.42	12.02
3º	-36.11	-39.17	-29.39	-36.68	-9.56	-37.61	-27.11	-28.48	-28.4	-47.48
4º	-2.6	-15.07	-19.95	-9.55	20.43	-20.68	-6.72	-10.06	-0.2	6.25
5º	-10.08	-22.92	-20.19	-22.22	-22.75	-21.4	-7.71	-9.61	-10.21	-15.35

Fig. 34. Variation of NO_2 in different time periods for the different population centers.

this heatmap, it can be observed that the impact of the lockdown has led to a significant reduction in the 2020 lockdown period compared to the average of previous years in all studied population centers, in more detail in Fig. 35. This has resulted in a generalized decrease in this pollutant in 2020 compared to previous years.

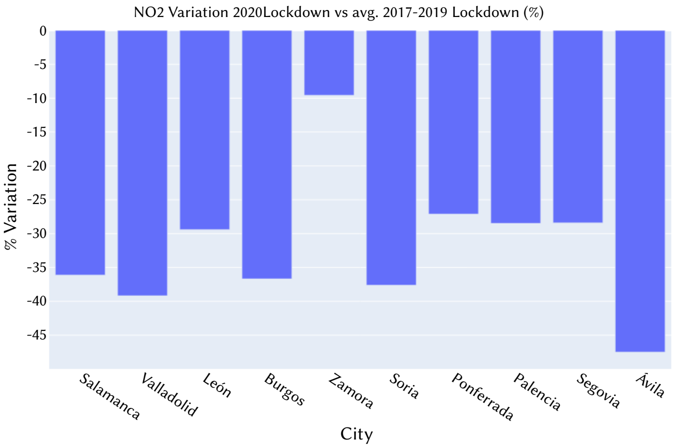


Fig. 35. Variation of NO_2 between the 2020 lockdown period and the average of the same dates in 2017, 2018, and 2019 for the different population centers.

The largest percentage decreases are found in the first of the proposed variations. In this case, in addition to the "lockdown factor", the decrease in values is due to the stationary trend of the pollutant (Fig. 13) that occurs during the lockdown period.

O_3

In the case of the O_3 pollutant, a similar behavior is observed during the lockdown period, as shown in Fig. 36, although its decrease is not as pronounced.

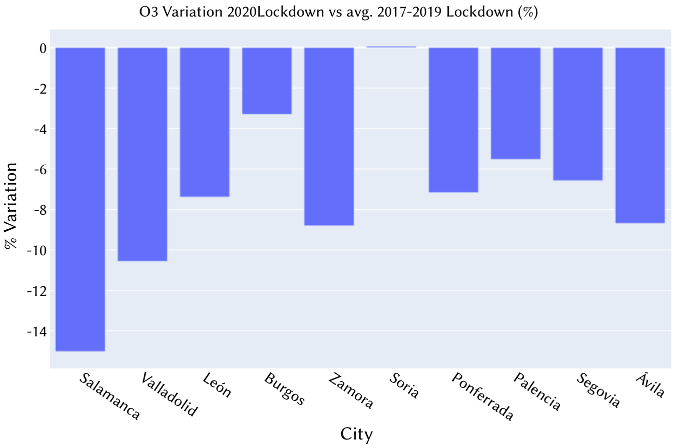


Fig. 36. Variation of O_3 between the 2020 lockdown period and the average of the same dates in 2017, 2018, and 2019 for the different population centers.

At the same time, as can be seen in the heatmap of this pollutant in Fig. 37, this leads to a decrease in the average values in 2020 across all population centers compared to the other years analyzed.

The increase observed during the lockdown period, compared to the period preceding it, is due to the stationary trend in which the highest peaks are reached during the lockdown, as can be seen in Fig. 28.

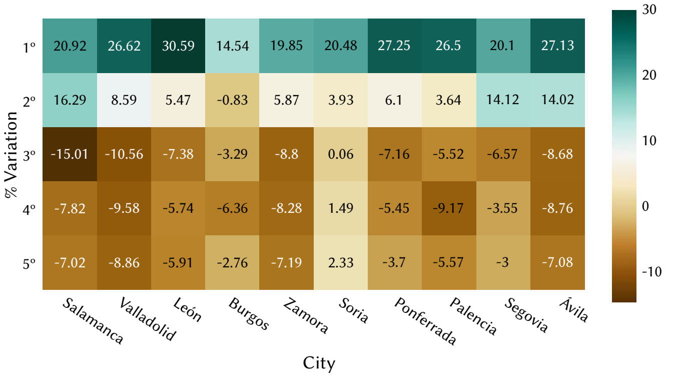


Fig. 37. Variation of O_3 in different time periods for the different population centers.

$PM_{2.5}$

In the population centers with data available for this pollutant, this analysis has been carried out. In this pollutant, we begin to see disparate behaviors among population centers during the lockdown period. In that period, compared to the average of previous years, only Valladolid and León experience a significant decrease (-12.47% and -10.48%, respectively) as show in the Fig. 38. Meanwhile, in the rest of the provinces, there is a slight increase in the following order: Ponferrada (+0.68%), Salamanca (+2.68%), and Burgos (+4.50%).

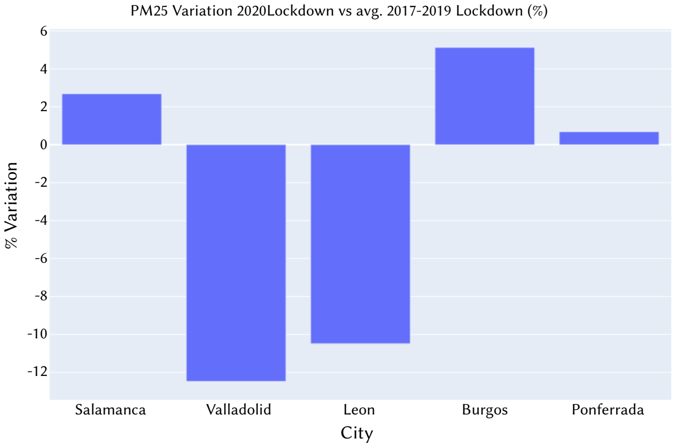


Fig. 38. Variation of $PM_{2.5}$ between the 2020 lockdown period and the average of the same dates in 2017, 2018, and 2019 for the different population centers.

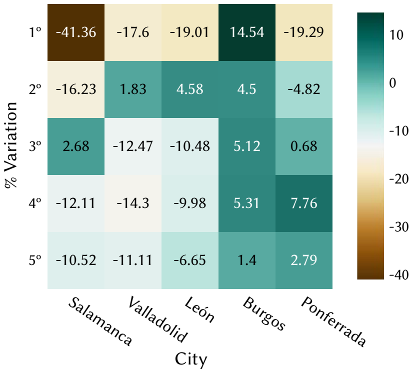


Fig. 39. Variation of $PM_{2.5}$ in different time periods for the different population centers.

Some of the most striking data are those presented by Salamanca and Burgos concerning the variation between the lockdown period and the one immediately preceding it (Fig. 39). Furthermore, both values are at the extremes of the scales and outside the range of the other population centers: While Salamanca shows a decrease of −41.36%, Burgos increases by 14.54% during that period. The data provided by Burgos is especially noteworthy, given that the stationary trend in that period is a decrease in pollutant values.

Finally, it should be noted that in the population centers of Salamanca, Valladolid, and León, there is a decrease in $PM_{2.5}$ values during 2020, while in Burgos and Ponferrada, there is an increase compared to 2019 and another one during the years 2017, 2018, and 2019 although at a lower rate.

PM_{10}

The analysis has been carried out for those population centers where it was not possible to do so with the $PM_{2.5}$ particle pollutant, due to the lack of data for that period and pollutant.

During the lockdown period, as seen in Fig. 40, all population centers manage to reduce their values compared to the average of previous years, achieving a significant decrease in Zamora (−27.93%) and Palencia (−27.46%); except for Ávila, which increased its values by 12.43% in this comparison.

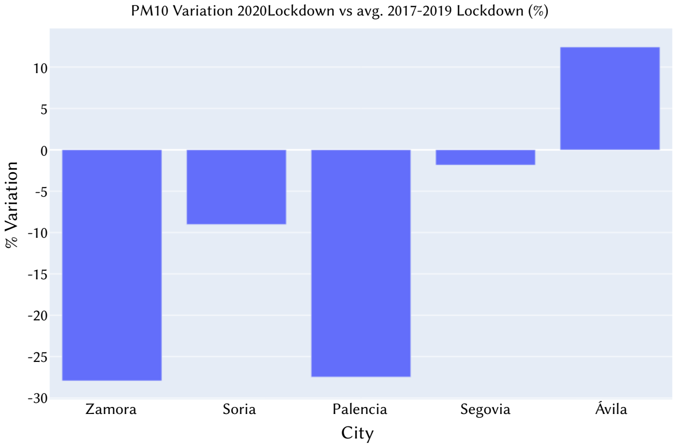


Fig. 40. Variation of PM10 between the 2020 lockdown period and the average of the same dates in 2017, 2018, and 2019 for the different population centers.

All population centers follow the stationary trend in PM_{10}, experiencing a decrease during the lockdown period compared to the previous period, as observed in Fig. 41.

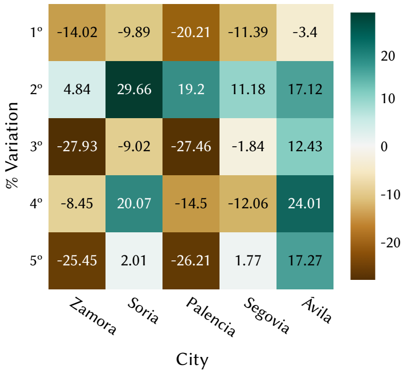


Fig. 41. Variation of PM_{10} in different time periods for the different population centers.

When comparing the data for the entire year 2020, Ávila is again found to be one of the provinces that stood out from the rest, registering an increase of up to 24% compared to 2019 and 17.27% compared to the averages of the years 2017, 2018, and 2019. Soria also stands out in the comparison of 2020 with 2019, with an increase in data of 20.07%.

A generalized decrease is observed in the rest of the population centers.

CO

The analysis of the CO pollutant has yielded diverse results (Ávila lacked data for the analysis pertaining to the periods intended to be scrutinized). On the one hand, it has shown a significant decrease during the lockdown period in the population centers of Valladolid, Soria, Ponferrada, and Segovia; and on the other hand, Salamanca, León, Burgos, Zamora, and Palencia experienced relevant increases (Fig. 42).

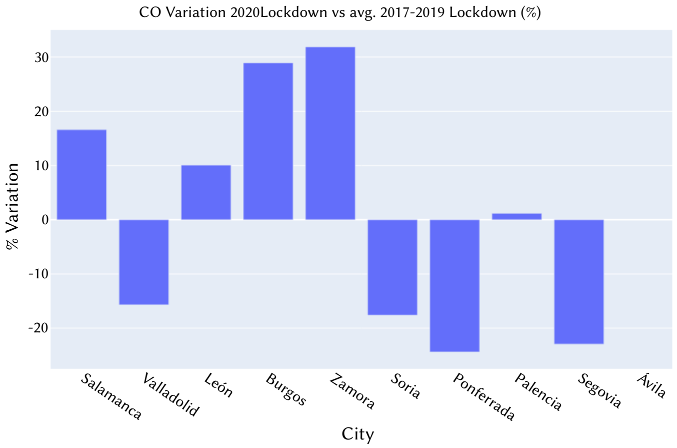


Fig. 42. Variation of CO between the 2020 lockdown period and the average of the same dates in 2017, 2018, and 2019 for the different population centers.

However, as seen in the heatmap in Fig. 43, Valladolid and Segovia recovered part of the lost values during the period following the lockdown, compared to the progress of the year so far. This is also due to the stationary trend, which causes the lowest levels to be reached during the period of the year in which the lockdown occurred (Fig. 33).

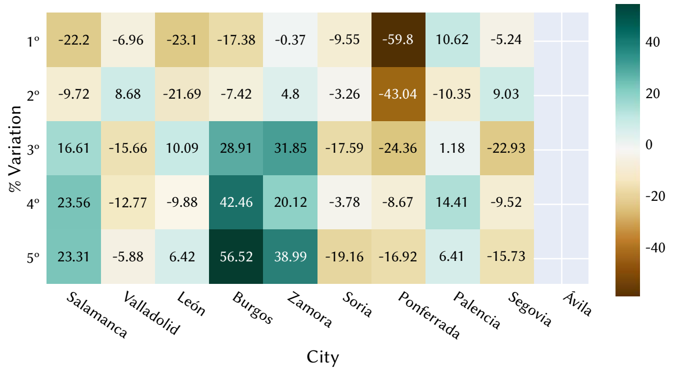


Fig. 43. Variation of CO in different time periods for the different population centers.

Delving into seasonality, it is seen that the variation between the lockdown period and the immediately preceding period, in general (except for Palencia), all population centers decrease their values, although in different ways depending on the impact during the lockdown.

Therefore, this leads to a decrease in 2020 compared to previous years in the population centers of Valladolid, Soria, Ponferrada, and Segovia; along with León, in the comparison of 2020 with 2019. The other group of provinces ended 2020 with a significant increase in their values compared to other years. This last point may be explained by the increasing trend of this pollutant over a few years, and one of the possible forecasts is that it will increase in the coming years, as can be seen in Fig. 33.

IV. Conclusions

The conclusions drawn from this study highlight the critical importance of having accurate predictions of pollutants, as this is essential for implementing measures to mitigate the damages caused by air pollution. Furthermore, it is important to investigate the causes, relationships, and trends of these pollutants in the short and long term, taking into account possible events that may alter their behavior, such as COVID-19. Accurate prediction allows for better information on air quality, enabling governmental organizations to prepare health plans that anticipate high levels of air pollution. Thus, they can adapt to any health event caused by atmospheric pollution phenomena.

The Prophet model has allowed us to make predictions that demonstrate a strong ability to forecast air quality in different spatial scenarios: various population centers with distinct regional characteristics; and temporal scenarios: in the short and long term, where attention should be paid to trends and seasonality. The possibilities for exploration with this approach are extensive in the field of air quality, surpassing ordinary prediction models such as LSTM or ARIMA. Particularly noteworthy are the cases of NO_2 y O_3 pollutants, where a high degree of accuracy is achieved, even for occasional peak levels. Moreover, they exhibit a precise prediction in any of the population centers according to the studied statistics. Furthermore, they show a precise forecast in any of the urban areas according to the statistics analyzed. This work therefore illustrates that Prophet has a broad capacity to forecast atmospheric pollution, and due to the fast training time and the lack of a complex system, it can be applied to other regions.

For the remaining pollutants discussed, several limitations regarding available data have hindered the model's ability to make accurate predictions, leading us to assess the importance of outliers, such as unanticipated meteorological events. Nonetheless, such intentionality was attributed to the constraints imposed by meteorological parameters on the adaptability of the model's prediction window.

In this study, emphasis is also placed on the analysis of trends for each pollutant and the seasonality they exhibit. This aids in achieving greater prediction accuracy and developing air quality plans that adapt accordingly.

Behavioral or restrictive events in society, such as COVID-19, disrupt the aforementioned factors, resulting in a significant impact on air quality and trends. In the current study, the implications of the lockdown due to the COVID-19 pandemic on air quality were assessed, in terms of variation and comparison among the studied population centers (the largest population centers in Castilla y León, Spain) during different periods surrounding the COVID-19 lockdown. It would be interesting to investigate, in future research, whether this event changed the behavior of the population and the interaction of pollutants with the environment.

The results showed a significant decrease in NO_2 y O_3 pollutants. This decline was not limited to the lockdown period, but the trend contributed to making 2020 one of the years with the lowest concentration of these pollutants in a long time. For the other pollutants, a decrease was also observed in most population centers, demonstrating how COVID-19 further emphasized the slope of the trend followed by these pollutants. It is worth noting that, in contrast to the other selected pollutants, CO experienced an increase in more than half of the studied population centers, confirming that its trend does not follow a decreasing pattern but rather stagnates. In general terms, with the exception of CO, a significant reduction in all atmospheric pollutants was observed during the closure period in the major population centers of Castilla y León. The findings of this study will be valuable for local municipal agencies and the administration of the Castilla y León region in order to establish rules and regulations aimed at enhancing and updating air quality in the future.

A. Limitations of the Study

The limitations of this study include geographical, as the study was limited to the provincial capitals and main cities of Castilla y León, and although the methodology of the experiment can be replicated, the results are only comparable and hardly extrapolable.

It should also be noted that the data are open data from government sources, where the accuracy of the sensors used to measure air quality is unknown. The study also does not include a review of meteorological data, such as wind gusts (speed and direction) or rainfall, which are relevant to pollutant dispersion.

V. Future Work Lines

Future research directions will focus on investigating the following aspects:

- To study and investigate the effects of wind gusts and their direction on the dispersion and concentration of these pollutants; identifying areas with higher pollutant concentrations would allow for the installation of green spaces in smart cities, which could improve air quality.
- To develop a federated learning architecture where different IoT devices for environmental monitoring can aggregate their readings and contribute to the training of models based on their location.
- To research Physics-Informed Neural Networks (PINNs) that are used to solve differential equations with applications in weather modeling, which may also help understand the movement of pollutant particles in the environment.
- To investigate long-term predictions based on the segmentation of time series into subseries that serve as input tokens to Transformer models and the independence of each channel. This approach would benefit from local information and long-term memory capabilities.

Acknowledgment

This work has been partially supported by the Institute for Business Competitiveness of Castilla y León, and the European Regional Development Fund under grant CCTT3/20/SA/0002 (AIR-SCity project).

References

[1] Bank, 2021. [Online]. Available: https://data.worldbank.org/indicator/SP.RUR.TOTL.ZS.

[2] G. Duranton, D. Puga, "The growth of cities," *Handbook of economic growth*, vol. 2, pp. 781–853, 2014.

[3] E. Commission, "Sustainable urban mobility in the eu: No substantial improvement is possible without member states' commitment," Jun 2020. [Online]. Available: https://op.europa.eu/webpub/eca/special- reports/urban-mobility-6-2020/en/.

[4] W. H. Organization, 2016. [Online]. Available: https://www.euro.who.int/ data/assets/pdf_file/ 0005/321971/Urban-green-spaces-and-health-review- evidence.pdf.

[5] W. H. Organization, "Air pollution," 2023. [Online]. Available: https://www.who.int/health-topics/air- pollution.

[6] Y. Zhu, J. Xie, F. Huang, L. Cao, "Association between short-term exposure to air pollution and covid-19 infection: Evidence from china," *Science of the total environment*, vol. 727, p. 138704, 2020.

[7] P. Chamoso, A. González-Briones, S. Rodríguez, J. M. Corchado, "Tendencies of technologies and platforms in smart cities: a state-of-the-art review," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[8] T. Yigitcanlar, R. Mehmood, J. M. Corchado, "Green artificial intelligence: Towards an efficient, sustainable and equitable technology for smart cities and futures," *Sustainability*, vol. 13, no. 16, p. 8952, 2021.

[9] V. Giannico, G. Spano, M. Elia, M. D'Este, G. Sanesi, R. Lafortezza, "Green spaces, quality of life, and citizen perception in european cities," *Environmental Research*, vol. 196, p. 110922, 2021.

[10] R. Casado-Vara, P. Novais, A. B. Gil, J. Prieto, J. M. Corchado, "Distributed continuous-time fault estimation control for multiple devices in iot networks," *IEEE Access*, vol. 7, pp. 11972–11984, 2019.

[11] P. Pihkala, "Eco-anxiety and environmental education," *Sustainability*, vol. 12, no. 23, p. 10149, 2020.

[12] R. Casado-Vara, A. Martín del Rey, R. S. Alonso, S. Trabelsi, J. M. Corchado, "A new stability criterion for iot systems in smart buildings: Temperature case study," *Mathematics*, vol. 8, no. 9, p. 1412, 2020.

[13] T. Yigitcanlar, N. Kankanamge, M. Regona, A. Ruiz Maldonado, B. Rowan, A. Ryu, K. C. Desouza, J. M. Corchado, R. Mehmood, R. Y. M. Li, "Artificial intelligence technologies and related urban planning and development concepts: How are they perceived and utilized in australia?," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 6, no. 4, p. 187, 2020.

[14] J. M. Corchado, F. Pinto-Santos, O. Aghmou, S. Trabelsi, "Intelligent development of smart cities: Deepint. net case studies," in *Sustainable Smart Cities and Territories*, 2022, pp. 211–225, Springer.

[15] Y. Mezquita, A. González-Briones, R. Casado-Vara, P. Wolf, F. de la Prieta, A.-B. Gil-González, "Review of privacy preservation with blockchain technology in the context of smart cities," in *Sustainable Smart Cities and Territories*, 2022, pp. 68–77, Springer.

[16] Y.-S. Chang, H.-T. Chiao, S. Abimannan, Y.-P. Huang, Y.-T. Tsai, K.-M. Lin, "An lstm-based aggregated model for air pollution forecasting," *Atmospheric Pollution Research*, vol. 11, no. 8, pp. 1451–1463, 2020.

[17] J. M. Corchado, "Technologies for sustainable consumption - researchgate. net," Apr 2021.

[18] R. López-Blanco, R. S. Alonso, J. Prieto, S. Trabelsi, "Automating the implementation of unsupervised machine learning processes in smart cities scenarios," in *Distributed Computing and Artificial Intelligence, Special Sessions, 19th International Conference*, 2023, pp. 71–80, Springer.

[19] T. Dimri, S. Ahmad, M. Sharif, "Time series analysis of climate variables using seasonal arima approach," *Journal of Earth System Science*, vol. 129, no. 1, pp. 1–16, 2020.

[20] R. López-Blanco, J. H. Martín, R. S. Alonso, J. Prieto, "Time series forecasting for improving quality of life and ecosystem services in smart cities," in *Ambient Intelligence—Software and Applications—13th International Symposium on Ambient Intelligence*, 2023, pp. 74–85, Springer.

[21] A. Hasnain, Y. Sheng, M. Z. Hashmi, U. A. Bhatti, A. Hussain, M. Hameed, S. Marjan, S. U. Bazai, M. A. Hossain, M. Sahabuddin, *et al.*, "Time series analysis and forecasting of air pollutants based on prophet forecasting model in jiangsu province, china," *Frontiers in Environmental Science*, p. 1044, 2022.

[22] J. Shen, D. Valagolam, S. McCalla, "Prophet forecasting model: A machine learning approach to predict the concentration of air pollutants (pm2. 5, pm10, o3, no2, so2, co) in seoul, south korea," *PeerJ*, vol. 8, p. e9961, 2020.

[23] G. Swamy, S. Nagendra, U. Schlink, "Impact of urban heat island on meteorology and air quality at microenvironments," *Journal of the Air & Waste Management Association*, vol. 70, no. 9, pp. 876–891, 2020.

[24] J. Ngarambe, S. J. Joen, C.-H. Han, G. Y. Yun, "Exploring the relationship between particulate matter, co, so2, no2, o3 and urban heat island in seoul, korea," *Journal of Hazardous Materials*, vol. 403, p. 123615, 2021.

[25] G. Miskell, W. Pattinson, L. Weissert, D. Williams, "Forecasting short-term peak concentrations from a network of air quality instruments measuring pm2. 5 using boosted gradient machine models," *Journal of environmental management*, vol. 242, pp. 56–64, 2019.

[26] W.-W. Li, R. Orquiz, J. H. Garcia, T. T. Espino, N. E. Pingitore, J. Gardea-Torresdey, J. Chow, J. G. Watson, "Analysis of temporal and spatial dichotomous pm air samples in the el paso-cd. juarez air quality basin," *Journal of the Air & Waste Management Association*, vol. 51, no. 11, pp. 1551–1560, 2001.

[27] S. Fei, R. A. Wagan, A. Hasnain, A. Hussain, U. A. Bhatti, E. Elahi, "Spatiotemporal impact of the covid-19 pandemic lockdown on air quality pattern in nanjing, china," *Frontiers in Environmental Science*, p. 1548, 2022.

[28] Y. Wang, S. Zhu, C. Li, "Research on multistep time series prediction based on lstm," 10 2019, pp. 1155–1159.

[29] M. C. Turner, Z. J. Andersen, A. Baccarelli, W. R. Diver, S. M. Gapstur, C. A. Pope III, D. Prada, J. Samet, G. Thurston, A. Cohen, "Outdoor air pollution and cancer: An overview of the current evidence and public health recommendations," *CA: a cancer journal for clinicians*, vol. 70, no. 6, pp. 460–479, 2020.

[30] "Datos abiertos de castilla y león. calidad del aire (por días)." https://datosabiertos.jcyl.es/web/jcyl/set/es/medio-ambiente/calidad_aire_historico/ 1284212629698. Accessed: 2023-07-07.

[31] W. Robson, "The math of prophet," Jun 2019. [Online]. Available: https://medium.com/future-vision/the- math-of-prophet-46864fa9c55a.

[32] S. J. Taylor, B. Letham, "Forecasting at scale," 09 2017, doi: 10.7287/peerj.preprints.3190v2.

[33] AEMET, "Opendata aemet." [Online]. Available: https://opendata.aemet.es.

[34] S. Abdullah, A. A. Mansor, N. N. L. M. Napi, W. N. W. Mansor, A. N. Ahmed, M. Ismail, Z. T. A. Ramly, "Air quality status during 2020 malaysia movement control order (mco) due to 2019 novel coronavirus (2019-ncov) pandemic," *Science of The Total Environment*, vol. 729, p. 139022, 2020, doi: https://doi.org/10.1016/j.scitotenv.2020.139022.

[35] R. Bao, A. Zhang, "Does lockdown reduce air pollution? evidence from 44 cities in northern china," *Science of The Total Environment*, vol. 731, p. 139052, 2020, doi: https://doi.org/10.1016/j.scitotenv.2020.139052.

[36] E. Bontempi, C. Carnevale, A. Cornelio, M. Volta, A. Zanoletti, "Analysis of the lockdown effects due to the covid-19 on air pollution in brescia (lombardy)," *Environmental Research*, vol. 212, p. 113193, 2022, doi: https://doi.org/10.1016/j.envres.2022.113193.

Raúl López Blanco

Predoctoral researcher, MSc in Web Site Management and Engineering (2021) by the International University of La Rioja (UNIR) (Spain) and BSc in Computer Engineering (2019) by the University of Salamanca (Spain). Researcher at the AIR Institute and member of the BISITE research group since 2019. Teacher in Internet of Things MSc and Industry 4.0 MSc at the University of La Rioja (UNIR) since 2022. Software developer with experience in programming languages such as C++, Java, C# JavaScript, web development, blockchain and databases, among others. His main research topics focus on traceability systems, databases, IoT and distributed systems. Raúl has published papers on IoT and AI and has participated in more than 10 research projects related to Smart Data, Smart Farming, Smart Healthcare, Deep Learning, Traceability, IoT and IIoT.

Miguel Chaveinte García

Last year student of the Degree in Computer Engineering at the University of Valladolid (UVa). Currently, he combines his studies with his work at Air Institute. His work within the organization focuses on several areas ranging from the development of backend systems to the research of new Deep Learning algorithms and explainable Artificial Intelligence, along with data analysis processes.

Ricardo S. Alonso, PhD.

Ricardo holds a PhD in Computer Engineering from the University of Salamanca (Spain), a MSc in Intelligent Systems from the same university and a Telecommunications Engineering degree from the University of Valladolid (Spain). He also holds a Master's degree in Business Administration (MBA) from the International University of La Rioja (Spain). On the one hand, he is currently a researcher at the AIR Institute and he is part of the BISITE Research Group (Bioinformatics, Intelligent Systems and Educational Technology) at the University of Salamanca. He has published more than 15 research articles in international journals and more than 35 in international conferences. During this career, he has collaborated in more than 40 regional, national, and European R&D projects, having been the Principal Investigator in regional and European projects funded by the Horizon 2020 program. His research interests include Internet of Things, Edge Computing, Distributed Ledger and Blockchain technologies, Embedded Systems, Indoor Location Systems, Cloud Computing and Artificial Intelligence. On the other hand, he has worked for more than 10 years in private enterprise as an entrepreneur, technologist, and researcher in the field of Wireless Sensor Networks and Real Time Location Systems, acting as Deputy R&D Director, R&D Director, Sales Manager and Co-Founder in different companies. He is also a lecturer at the International University of La Rioja, where he is Academic Director of the Interuniversity Master's Degree in AI for the Energy and Infrastructure Sector.

Javier Prieto, PhD

Full Professor at the University of Salamanca in the Department of Computer Science and Automation. He is a member of BISITE and ATA (University Institute for Research in Art and Technology of Animation). BSc in Telecommunications Engineering and PhD in Information and Communication Technologies (University of Valladolid). Since 2007, Javier Prieto has worked in diverse public and private research centres: Fundación Centro para el Desarrollo de las Telecomunicaciones en Castilla y León (CEDETEL), University of Valladolid (UVa), or the Massachusetts Institute of Technology (MIT) in Cambridge, MA (USA). He has received competitive grants as Postdoctoral Researcher by the University of Salamanca, or Postdoctoral Researcher Torres Quevedo by the Ministry of Science and Innovation. He has published over 150 articles in international journals, books, and congresses, and is the author of 2 national patents. He is the editor in chief of the Smart Cities journal, at its IoT Section, and a senior editor of the journals IEEE Communications Letters and Wireless Communications and Mobile Computing, a guest editor of numerous special issues in different journals, and a member of the scientific committee of the Advances in Distributed Computing and AI Journal. He is co-founder and Technical Secretariat at AIR Institute, IEEE Senior Member, board member of IEEE Spain Section, Head of the IEEE Blockchain Initiative, Operations Chair of the IEEE GLOBECOM 2021 congress, member of the TC Cognitive Networks (TCCN) Technical Committees of the IEEE ComSoc and RFID Technologies Committee (MTT-24) of the IEEE MTT-S. He has been president of the technical program committee of the IEEE ICUWB 2017. Teaching activity: Full Professor at the University of Salamanca and Associate Professor at the University of Valladolid. Javier has been professor of the Master in Big Data Science at the University of Valladolid and the Master in Industry 4.0 (IIoT) at the International University of La Rioja. Research interests: social computing and AI for smart cities, location and navigation technologies in both indoor and outdoor environments, and Bayesian inference techniques for improving social welfare and sustainable development. He has coordinated and participated in over 100 national and international R&D projects.

Juan M. Corchado, PhD.

PhD in Computer Sciences (University of Salamanca). PhD in Artificial Intelligence (University of the West of Scotland). Director of the BISITE Research Group at the University of Salamanca. Director of the IoT Digital Innovation Hub. Visiting Professor at the Osaka Institute of Technology. Visiting Professor at the University Malaysia Kelantan. Member of the Advisory group on Online Terrorist Propaganda (European Counter Terrorism Centre, EUROPOL). He is currently University Professor, Director of Postgraduate Studies in Security, Information Systems, social media, Mobile Telephony, Digital Animation, IoT and Blockchain, having directed 17 doctoral theses. He is also the director of the IoT Digital Innovation Hub and Academic Director of the University Institute for Research in Animation Art and Technology at the University of Salamanca. He is a Visiting Professor at the Osaka Institute of Technology (Japan). Previously he has been Vice-rector of Research and Transfer at the University of Salamanca (2013-2017), Director of the University of Salamanca Science Park (2013-2017), Deputy Director of the School of Computer Engineering at the University of Vigo (1999-00), researcher at the University of Paisley (1995-98) and Dean in the Faculty of Sciences at the University of Salamanca (2008-2013). He worked as a programmer at the Oceanographic Laboratory in Plymouth (United Kingdom) between 1993 and 1998. Juan M. Corchado is President of the Spanish chapter of the association IEEE Systems, Man and Cybernetics and member of the Advisory Group on Online Terrorist Propaganda of EUROPOL. He is also editor-in-chief of ADCAIJ (Advances in Distributed Computing and Artificial Intelligence Journal) and IJDCA (International Journal of Digital Contents and Applications). He is co-author of more than 350 books, book chapters, articles in international journals, etc., most of them presenting both practical and theoretical aspects related to Hybrid Artificial Intelligence Systems, Distributed Computing, Biomedicine, Ambient Intelligence, Wireless Network Systems, Multiagent Systems, etc. He has worked on more than 100 research projects (European, National and Autonomous) and research contracts (article 83). He has been the coordinator and Principal Investigator in different H2020 and Interreg PocTep project, as well as numerous national projects.

unir
LA UNIVERSIDAD
EN INTERNET

Rectorado
Avenida de la Paz, 137
26006 Logroño (La Rioja)
t (+34) 941 21 02 11

www.unir.net