International Journal of Interactive Multimedia and Artificial Intelligence

March 2023, Vol. VIII, Number 1
ISSN: 1989-1660

unir LA UNIVERSIDAD EN INTERNET

"By far the greatest danger of Artificial Intelligence is that people conclude too early that they understand it."
Eliezer Yudkowsky

Special Issue on AI-driven Algorithms and Applications in the Dynamic and Evolving Environments

INTERNATIONAL JOURNAL OF INTERACTIVE MULTIMEDIA AND ARTIFICIAL INTELLIGENCE

ISSN: 1989-1660 –VOL. 8, NUMBER 1

EDITORIAL TEAM

Editor's Note

With the rapid development of information and communication technologies, artificial intelligence and IoTs, more and more advanced technologies, such as machine learning, reinforcement learning, neural networks and fuzzy systems, have been introduced into industrial practices. The application of advanced technologies has greatly promoted the process of industrial revolution. However, there is big gap between controlled simulation and real evolving environment, which results in the unsatisfactory performance of the typical algorithms in practical environments. For example, in Underwater IoTs, a dynamic and uncertain marine environment can cause equipment damage, resulting in huge financial losses. Therefore, improving the robustness and adaptability of algorithms and systems, and proposing new solutions in practical applications to meet the requirements of self-developing, self-organizing, and evolving systems is essential to promote intelligent industrial applications.

This Research Topic aims to collect researches focusing on addressing the problems of evolving system modelling, clustering, classification, prediction and control in non-stationary, unpredictable environments. The scope of this topic includes: (1) Robustness of environment modeling in evolutionary system, (2) Robustness of artificial intelligence algorithms, (3) Adaptability of neural networks and systems, (4) Prediction of intelligent algorithms in dynamic environments, (5) Improvement of robustness in deep learning algorithms, (6) Interpretability of predictive models in dynamic environments, (7) Application of AI technology in Industrial Internet of Things, (8) Uncertainty in Intelligent Transportation System, (9) The dynamic environment of Underwater Internet of Things, (10) Applications and migration of intelligent algorithms.

Specifically, the present Special Issue includes the topics described below.

Zhang et al. proposed a dataset containing a variety of elements. They construct a corresponding out-of-distribution test set. They explored the distribution characteristics of efficient datasets in terms of angle element, and confirmed that an efficient dataset tends to contain samples with different appearance.

Roy et al. discussed using the Internet of Medical Things in the COVID-19 crisis perspective. This paper suggested an ensemble transfer learning framework to predict COVID-19 infection, which predicted the COVID-19 infected people with an F1-score of 0.997 for the best case.

Hurtado et al. presented a novel approach for Human Activity Recognition (HAR) in healthcare to avoid the risk of mortality caused by physical inactivity. The model took advantage of the large amount of unlabelled data available by extracting relevant characteristics.The proposed approach can properly classify movement patterns in real-time conditions.

Saxena et al. proposed a network centrality based approach combined with graph convolution networks to predict the connections between network nodes. They also proposed an idea to select training nodes for the model based on high edge, which improved the prediction accuracy of the model.

Arroni et al. proposed an attention-based model that used the transformer to predict the sentiment expressed in tweets about hotels in Las Vegas. They crafted a transformer architecture model much simpler and smaller than the mentioned models for specific problems, which does no need a whole language representation.

Chen et al. proposed a new spatio-temporal attention graph convolution network (STAGCN) for sea surface temperature prediction. STAGCN can capture spatial dependance and temporal correlation. Experiments showed that the model can capture the spatio-temporal correlation of regional-scale sea surface temperature series and outperforms models under different sea areas and different prediction levels.

Andueza et al. used time series models, i.e., autoregressive integrated moving average and seasonal autoregressive integrated moving average to forecast the impact of COVID-19 on sales of cigarette in Spanish provinces.

Maestro et al. proposed a blockchain-based decentralized architecture for cloud resource management systems. They analyzed and compared the characteristics of the proposed architecture concerning the consistency, availability, and partition resistance of architectures that rely on Paxos/Raft distributed data stores. And they demonstrated that the proposed blockchain-based decentralized architecture noticeably increased the system availability.

Sinha et al. proposed a method to select web data sources for web data warehouse. This work was based on the probabilistic analysis of SAW and TOPSIS, which deal more efficiently with the dynamic and complex nature of web.

We would like to thank all of the contributors to the research topic including authors, reviewers, and the IJMAI editorial and production offices.

Jiachen Yang[1]
Houbing Song[2]
Muhammad Khurram Khan[3]

[1] Tianjin University (China)
[2] University of Maryland, Baltimore County (USA)
[3] King Saud University (Saudi Arabia)

DOI: 10.9781/ijimai.2023.02.013

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.*

Dataset and Baselines for IID and OOD Image Classification Considering Data Quality and Evolving Environments

Zhuo Zhang[1], Yang Li[1,2]*, Yicheng Gong[1], Yue Yang[1], Shukun Ma[1], Xiaolan Guo[1], Sezai Ercisli[3]

[1] School of Electrical and Information Engineering, Tianjin University, Tianjin (China)
[2] College of Mechanical and Electrical Engineering, Shihezi University, Shihezi (China)
[3] Department of Horticulture, Faculty of Agriculture, Ataturk University, Erzurum (Turkey)

Received 20 March 2022 | Accepted 3 October 2022 | Early Access 24 January 2023

Abstract

At present, artificial intelligence is in a period of rapid development, and deep learning has begun to be applied in various fields. Data, as a key part of the deep learning, its efficiency and stability, will directly affect the performance of the model, so it is valued by people. In order to make the dataset efficient, many active learning methods have been proposed, the dataset containing independent identically distribution (IID) samples is reduced with excellent performance; in order to make the dataset more stable, it should be solved that the model encounters out-of-distribution (OOD) samples to improve generalization performance. However, the current active learning method design and the method of adding OOD samples lack guidance, and people do not know what samples should be selected and which OOD samples will be added to better improve the generalization performance. In this paper, we propose a dataset containing a variety of elements called a dataset with Complete Sample Elements(CSE), the labels such as rotation angle and distance in addition to the common classification labels. These labels can help people analyze the distribution characteristics of each element of an efficient dataset, thereby inspiring new active learning methods; we also construct a corresponding OOD test set, which can not only detect the generalization performance of the model, but also helps explore metrics between OOD samples and existing dataset to guide the selected method of OOD samples, so that it can improve generalization efficiently. In this paper, we explore the distribution characteristics of efficient datasets in terms of angle element, and confirm that an efficient dataset tends to contain samples with different appearance. At the same time, experiments have proved the positive influence of the addition of OOD samples on the generalization performance of dataset.

Keywords

Active Learning, Data Quality, Efficient Dataset, Evolving Environments, Generalization.

DOI: 10.9781/ijimai.2023.01.007

I. Introduction

In recent years, with the development of artificial intelligence, deep learning is widely used in various fields, such as the detection and prevention of plant diseases and pests, the intelligent recognition of medical images and so on [1]–[6]. However, deep learning needs a large amount of data as the basis, which brings problems such as high data cost, difficult data acquisition and so on. In order to solve the demand problem of a large amount of data, people put forward few-shot learning, which is committed to learning from a small amount of labeled data and obtaining generalization ability [7]–[10]. Methods in the field of few-shot learning have solved the problem of large-scale data dependence to a certain extent. However, although selecting samples with high information quality for training can effectively help neural networks improve performance, little attention has been paid to the quality of sample information. A sample with high information quality not only has less noise, but also has a large difference with the existing samples in the data set. That is, conducting data quality assessments can improve the model performances under the same budget, and reduce the sample collection budget with the same model performances. Therefore, it is of great significance to carry out data quality assessment and establish relevant baselines in typical applications such as identification and classification.

In addition to considering the problem of data quality, the changeable test environment is a practical problem that can not be ignored, which widely exists in industrial processing and manufacturing, automatic driving, field environment and so on [11]. For this problem of poor generalization of the model caused by the change of test data, also known as the difference of out of distribution(OOD), it is necessary to establish a data set that fully considers the change of scene factors to provide a fair comparison platform for relevant research. Although there have been many studies on the generalization of model algorithms, there is still a lack of data set construction. In particular, considering the changes of environmental factors, the construction of high-quality data sets without watermark and error label is of great significance to the promotion of subsequent related research.

* Corresponding author.
E-mail address: liyang328@shzu.edu.cn

In order to solve the above problems, this work has built an all element image acquisition platform and formed a Complete Sample Elements (CSE) data set, which can support the research and analysis of independent identically distribution (IID) and OOD. Probability entropy and distance entropy are proposed to evaluate the quality of the data set and establish relevant test baselines. In the aspect of OOD test, taking the real shooting data in the actual dynamic environment as the test, the relevant baseline is established, and the impact of distribution differences on the performance of the algorithm is explored.

The structure of this document is as follows: section II introduces the relevant work at home and abroad, section III introduces the CSE data set, section IV is the experimental part of this paper, and section V presents the conclusions and future works.

II. Related Works

Image quality evaluation is a basic and challenging problem in the field of image processing. Traditional image quality evaluation is realized by human visual system (HVS) or objective image quality assessment (IQA) [12]–[14]. It can evaluate the distorted images such as blur, JPEG compression and noise, and realize the discrimination of distortion types. However, these quality evaluation criteria serve human subjective visual perception and have nothing to do with the improvement of machine vision task performance. The development of artificial intelligence has promoted people to pay attention to data quality from the perspective of improving task performance. Recently, some work has also paid attention to the data quality of classification task guidance, such as the active cleaning of data labels proposed by Bernhardt in 2021 [15]. In addition, some works have paid attention to the influence of sample information quality on model performances, and they have proposed some sample information quality assessment methods on this basis [16], [17]. However, current methods lack validation on large-scale datasets containing constituent elements.

In the real scene, there are differences between train data and test data. How to effectively improve the test effect is a very valuable research direction. Under the guidance of this research direction, a variety of theoretical research methods on task generalization represented by transfer learning have been formed, so as to reduce the dependence on a large number of target domain data [18]. From the perspective of research subjects, transfer learning can be divided into data-based transfer learning, feature-based transfer learning, model parameter based transfer learning and so on. The data-based transfer learning method focuses on the transfer of knowledge through the adjustment and transformation of data; the feature-based method transforms each original feature into a new feature representation; model based transfer learning uses sub modules such as classifier, extractor or encoder to make accurate prediction results for the target domain, such as classification or clustering results [19]–[23]. However, these works focus more on theoretical research, and the data used are still quite different from the real scene.

III. CSE - A Dataset With Complete Sample Elements

In this section, we propose a dataset with complete sample elements, abbreviated as CSE. This dataset will facilitate the following two research topics:

- In addition to the common classification labels, the dataset also labels the remaining elements. When the train and test sets exhibit IID distributions, the element distribution characteristics of efficient datasets can be analyzed.
- When the train and test sets exhibit OOD distributions, the influence of OOD samples on generalization performance can be analyzed.

The dataset is divided into 11 categories, each class can be subdivided into 5 subclasses, so there is a total of 55 subclasses, each subclass is sampled from the same object. In Fig. 1, we show a representation of the images in it. Each subclass has 216 images, with 72 degrees and 3 distances. The background of the dataset is unified as a large checkerboard, and the size of the collected data is unified as 640×480. Since there are objects with small size in this dataset, in order to ensure that the collected sample subject is located in the center, we use a cylindrical heightening pad to support tiny objects. The dataset is publicly available for researchers to download and study[1].

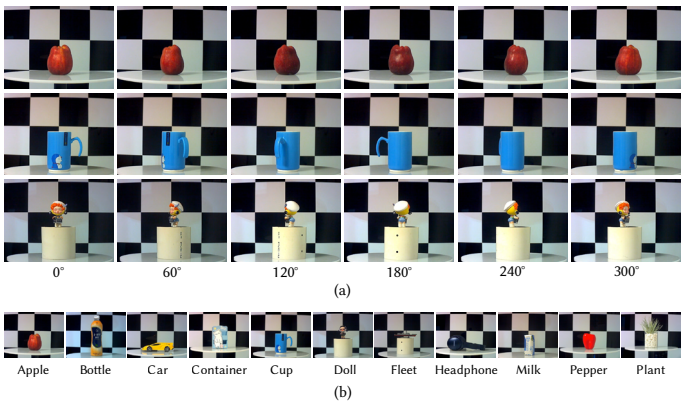


Fig. 1. Some samples of the CSE train set. (a) The examples of rotation; (b) All classes of the CSE dataset.

A. The Need to Structure Element-complete Datasets

1. For IID

For the train set and test set of IID distribution, there is redundancy within the train set. According to our test situation on the CIFAR-10 dataset, the sub-train set selected by the active learning method can obtain performance close to the entire dataset on fewer datasets, which will greatly improve the training efficiency. However, the current active learning methods has shortcomings such as relying on the selection of base classes, and the results are not robust. Therefore, if we can provide a dataset with complete elements, and understand why these samples will improve the performance of the dataset from the distribution level of elements, this will help us in the evaluation of sample information selecting. It should be noticed that in the class of doll and fleet, the objects are too small that it should be supported by a cylinder. To minimize the impact caused by the cylinder, we crop these samples, so it would be like Fig. 2.

Fig. 2. Some samples of the "doll" class in the CSE dataset. Objects in the "doll" class are tiny, so all samples of this class are cropped. The front, side, and back of this class of samples look very different, so if the network has only seen some of them (such as the front and the side), the rest of the samples (the back) are high-informative.

2. For OOD

For the train set and test set of OOD distribution, the information about the test set provided by the train set is insufficient, which will lead to a plummeting performance of the network model. If OOD

[1] Here: http://aimip.tju.edu.cn/rgzn.htm

samples are added to the train set, the network performance of the model will be improved; in future research, we will also try to calculate the distance between the train set and the test set and measure the similarity, and use the parts with high similarity to train the neural network, which will make the network more generalizable to the OOD test set.

B. How to Obtain a Dataset With Complete Elements

In order to make the sampling process automatic and controllable, we built a multi-DOF sampling platform, as shown in Fig. 3. The device is composed of three servo motors, which can realize front and rear, left and right, and up and down transforms, thereby changing the angle of the view captured by the camera.

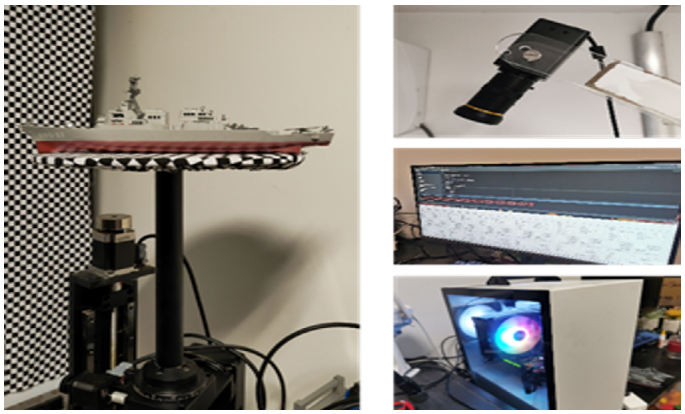


Fig. 3. The platform to sample the CSE dataset.

C. Variable Settings Supported by IID Train Set

After selection, we decided to design the following three variables to control:

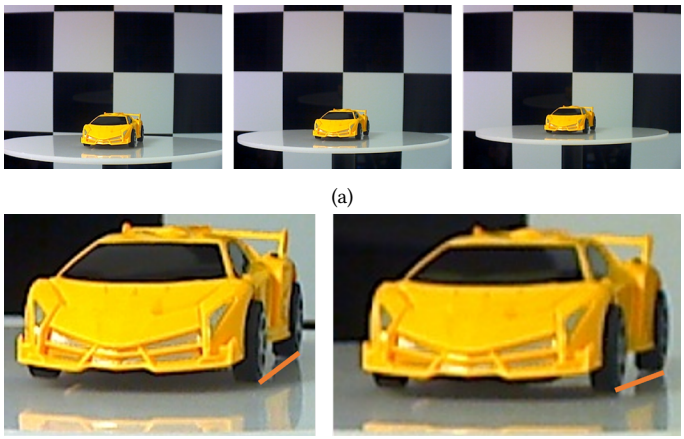
1. Rotation Angle

It is well known that for most asymmetric objects, the difference in viewing angle affects the observation results, and the same is true for neural networks. For example, as shown in Fig. 2, this figure shows the different angles of the front, side and back of the doll. For neural networks, especially those without any pre-training, they tend to think that these three angles are of different samples. The process of training is also the process of grouping samples of the same object from different angles into one group. The original intention of our design of the angle variable is how to choose an appropriate angle to reduce the number of samples in the train set as much as possible and improve the test accuracy as much as possible. We believe that this topic will be very helpful for future dataset simplification and dataset information Quantitative work.

When constructing the CSE dataset, we collect a sample every 5″, and each sample can collect 72 images at the same camera distance. As for why 5″ was chosen instead of 10″ or 1″, we have the following considerations. First, we believe that a full-featured dataset should be linear and smooth, so the angle of acquisition should be as small as possible, or as imperceptible as possible. However, collecting samples from an angle that is too small will greatly increase the number of samples, and many problems will follow: first, an excessively large number of samples will cause a serious burden on storage; second, an excessively large number of samples will prolong the training time; third, and most important point, repeatedly feeding a large number of samples with the same background and similar appearance to the model can easily make the model overfit, and even learn the background, object tray, and other elements incorrectly, which will cause a serious problem with the network model on which the object selecting method(distance entropy, probability entropy, etc.) relies. Therefore, considering the reasons above, we choose to collect a sample every 5″.

2. Distance Between Object and Camera

The distance of the camera determines the proportion of the object in the sample. The farther the distance between the object and the camera is, the smaller the area of the object in the sample and the larger the area of the background. At the same time, the distance of the camera will also affect the viewing angle, as shown in Fig. 4. The farther the object is, the narrower the range that can be seen, which may bring an extra amount of information to the learning of the network model.



(a)

(b)

Fig. 4. The distance between object and camera also affects the sample. (a) Three distances when sampling a car; (b) Details of the farthest sample and the nearest sample under the same placement angle. As the text says, the connection line (orange) on the underside of the wheel varies with distance.

D. Design of OOD Test Set

Out-of-distribution test samples are encountered in some specific tasks. There are many reasons for the existence of OOD samples. For example, the initial design of the data set is not well thought out, or the data itself is difficult to collect in large quantities, and the train set can only be generated by simulation. How does the model make use of the OOD sample information it encounters? This is also the original intention of our design of the OOD test set.

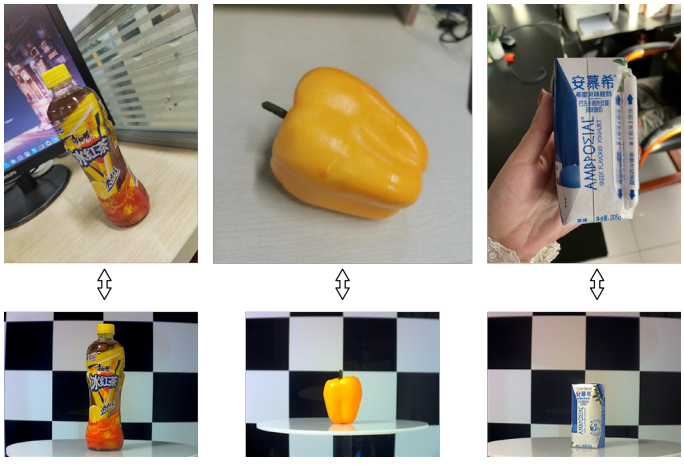


Fig. 5. The comparison between train samples and the corresponding OOD sample.

As shown in Fig. 5, when shooting the OOD test set, we randomly changed the background, randomly rotated the shooting angle, and randomly raised the shooting angle. These operations are very similar to the situation of OOD samples encountered in industry, since there are no such samples in the dataset. At the same time, the OOD samples are taken by mobile phones. Compared with the camera used to collect the train set, the default focal length of the mobile phone is shorter, and the captured samples will have some deformation compared with the train set, which is also a feature of the OOD samples.

IV. Results and Analysis

A. List of Backbones and Training Configuration

In order to verify whether the dataset we construct can effectively reflect the IID distribution and OOD distribution, we conducted several experiments with multiple backbone networks: ResNet [24], VGG [25] and WRN [26]. When we verify IID distribution, we use a uniform sampling of 25% of the train set as the IID test set. The purpose of this is to make the IID test set show a uniform distribution, and because the original train set is uniformly sampled, this sampling method will make this new IID test set present a similar distribution to the original train set, so the new IID train and test set can be approximated. When we verify OOD distribution, we directly use the test set and train set for OOD training and testing. All subsequent experiments are conducted under a single server with an Intel Core i7-12700KF CPU, dual nVidia GeForce RTX 3080Ti GPU and 128 GB memory with PyTorch.

B. Backbone Network Performances on IID and OOD Test Sets

The results obtained by training and testing on the backbone network are shown in Table 1. Note that the three backbone networks we listed all get 100% test accuracy on the IID combination, which proves that the IID train and test sets are IID distributions of each other, which is consistent with our assumption in IV.A. However, the three backbone networks perform poorly in the OOD combination. Given the undisputed high performance of the three backbone networks, it can also be shown that the OOD train and test sets are distributed in OOD.

In particular, we add a set of pre-trained comparison experiments in Table I. The control group has essentially the same parameters as the experiments using the three backbones, but with the ImageNet pre-trained model. It can be seen that a group of experiments using the ImageNet pre-training model is significantly higher in testing accuracy than the group that does not use ImageNet, which confirms the prediction in III.C.1. Using a train set with an interval of 5° has caused the model to overfit, as explained below. The result of VGG is a little bit lower, it is because the performance is weaker than ResNet and WRN.

TABLE I. The Test Accuracy of Several Backbones

settings	ResNet	VGG	WRN
IID, non-pre-training	100%	100%	100%
OOD, non-pre-training	18.808%	17.636%	20.268%
OOD, pre-training	29.256%	26.343%	36.696%

Experiment parameters: batch size: 32; Epoch: 50; initial learning rate: ResNet & WRN: 0.01; VGG:0.1; step learning rate: decay epoch: [20, 30, 40], gamma: 0.1

First, using ImageNet and reducing the learning rate is to make the model "remember" the ImageNet distribution as much as possible. Second, ImageNet is similar to our OOD test set collection method, and the background environment is more variable, which it also does in OOD test set. It can be considered that ImageNet has a similar distribution to the OOD test set. Third, since the pre-trained model achieves convergence in the later stage, it means that the model has also learned the distribution of the OOD train set, and so it does in non-pre-training group. In the comprehensive comparison experiment, the test performance of the ImageNet group is higher than that of the non-pre-training group. Therefore, we have reason to believe that the use of ImageNet pre-trained network can effectively suppress the overfitting phenomenon. The information of ImageNet makes the model not affected by factors such as background and thus overfit. While the non-pre-training group appears some overfitting phenomenon, which further deteriorates the performance on OOD test sets. When we use the Grad-CAM [27] method to visualize the attention of the network, we can see that the non-pre-training group totally cannot pay attention to the objects, but the pre-training group can accurately recognize them, as it can be seen in Fig. 6.

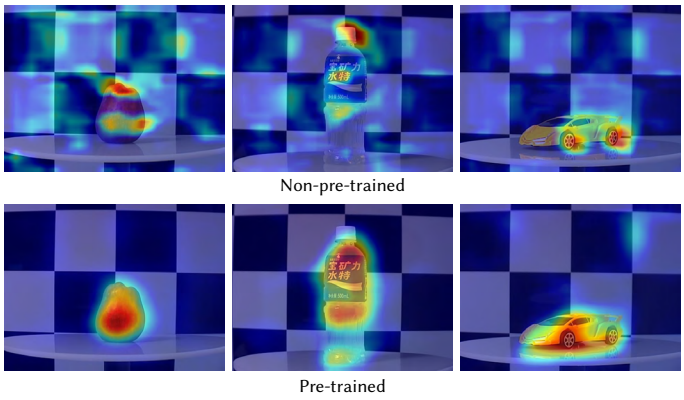


Fig. 6. When visualizing the model by Grad-CAM, we can see clearly that the non-pre-trained group has a stronger overfitting phenomenon than the pre-trained group.

C. Rotation Angle Distribution Features of Efficient Datasets

In order to explore the element distribution characteristics of efficient datasets derived from IID train set, we use two methods in active learning: distance entropy [28] and probability entropy [29]. We design a series of experiments: first, select 88(1%) IID samples as the base, add 88(1%) samples in each round of experiments, a total of 9 rounds. The subsets selected by these methods are re-trained on the ResNet18 network, and the IID test accuracy is shown in Fig. 7. It can be seen that the test accuracy of the subset obtained by selecting 528(6%) samples can already reach 99%. We take the subset with 528(6%) samples selected by distance entropy as an efficient train set for subsequent analysis.

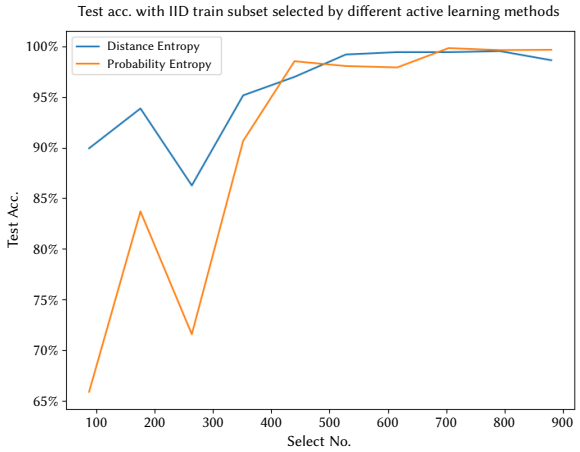


Fig. 7. The IID test accuracy of models trained on subsets selected by distance entropy method and probability entropy method.

1. Irregular Objects

When an irregular object rotates around to collect several samples, each sample has a large change from other samples, such as samples of cars, ships, dolls, etc. We take the sample from the third doll in the nearest position as an example to explore the rotation angle element distribution characteristics of irregular samples in an efficient dataset.

As shown in Fig. 8, the samples with high information content are distributed at 110°-205° and 270°-330°. At these degrees, the object is basically at the front or back angle, and the sample taken after rotation changes greatly, so it brings more information; while the side angle of the object is almost the same as the 205°, 270° samples, so less information. In the repeated experiments, we also did a set of similar experiments with cars, and the results were similar, as shown in Fig. 9, except that the side is high information, and the front and back are low information. This shows that the wider surface with larger rotation variation of irregular samples has high information content.

Fig. 8. The dolls selected by active learning method. In the CSE dataset, the front and back samples of this doll are of much more information, but the side samples are of less information.

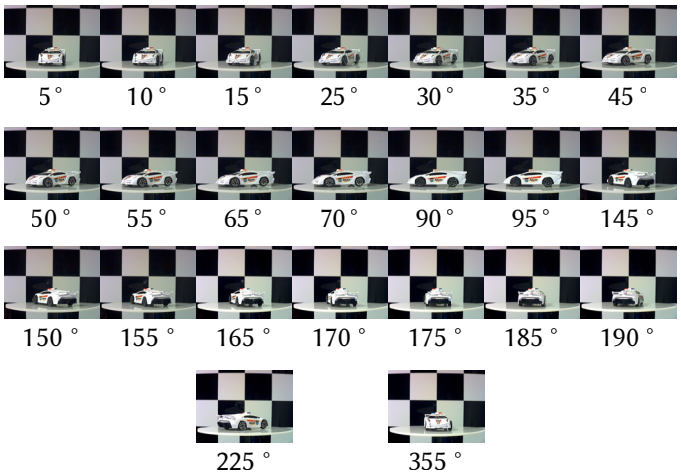


Fig. 9. The cars in the efficent dataset. Different from the dolls in Fig. 8, the side samples are of more information.

2. Rotation-Invariant Objects

Some objects are rotationally invariant, such as apples, containers, etc. Their characteristic is that samples taken from any angle are similar. Our experiments show that the number of rotation-invariant samples in the efficient dataset is much less than the number of irregular samples, such as the first apple corresponding to only seven samples (in contrast, each subclass of dolls generally selects at least 50 samples), and the angle has no regularity. Similar to our previous proof, rotation-invariant samples only need to find a few samples as representatives to obtain most of the information.

3. Approximately Rotation-Invariant Objects

There are also some objects that are approximately rotationally invariant in this dataset. Their main parts are rotationally invariant, but they also have other components that make them rotationally invariant, such as the handle of a cup. The characteristic of this type of object is that when the components that affect its rotation invariance are occluded, the samples have high similarity, as shown in Fig. 10. We take the sample of the second cup at the farthest position as an example to explore the rotation angle element distribution characteristics of approximately rotation-invariant samples in an efficient dataset.

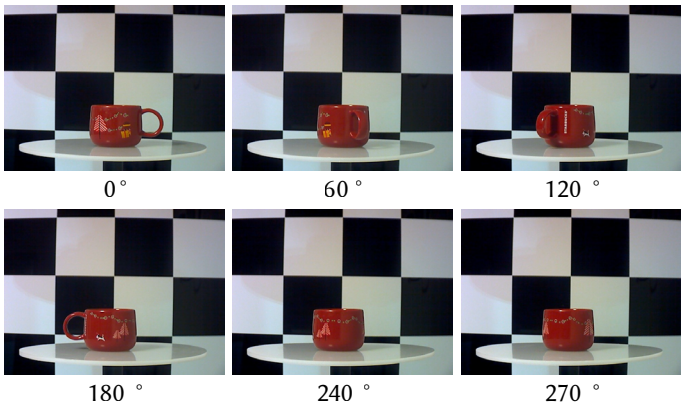


Fig. 10. Examples of approximate rotation-invariant objects. When the rotation angle comes from 240″ to 270″, the cup seems nearly the same, which means the low information in the samples.

As shown in Fig. 11, the samples with high information content are distributed between 45°- 70°, 185°-210°, and 305°-355°. Under these several degrees, the cup handle is on the side or front of the cup body, and the change is more obvious when rotating the object, and the samples of the cup handle behind the cup are relatively similar, so only a few samples can be selected. At the same time, when the handle is in front of the cup, since the color of the handle is closer to the cup, and the difference when it is rotated is smaller than that when the handle is on the side (only the 70° samples are sampled).

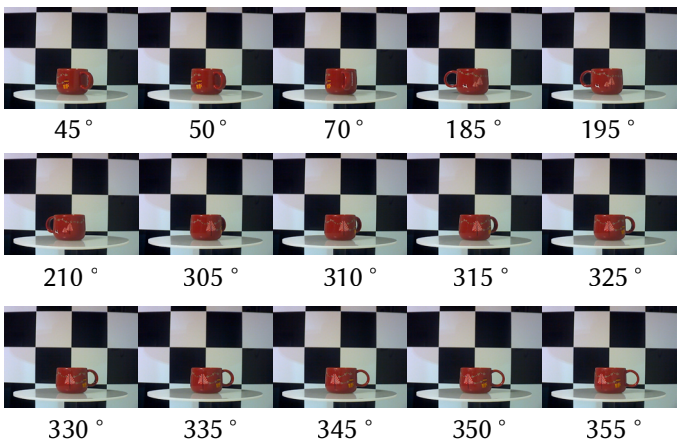


Fig. 11. The cups in the efficent dataset. when the handle is behind the cup, it will be unlikely selected.

At the same time, after our further statistics, we found that the irregular objects samples have the highest ratio in efficient datasets, the approximate rotation-invariant samples are in the middle, and the rotation-invariant samples are the lowest. The above analysis not only verifies the hypothesis that the IID train set has high redundancy, but also classifies the sample features with high information. It is found

that the active learning methods will tend to select more differentiated samples to form a dataset, so that the selected samples have high information.

D. Effect of Adding OOD Samples on Generalization Performance of Models

In order to explore whether the addition of OOD samples will affect the generalization performance of the classification model, we designed the following experiments: 2% of the OOD train set samples of each class were selected to join in the OOD train set for training and the rest of the samples were used for testing, adding a total of 10 rounds, up to 20%. The experimental results are shown in Table II. As with our assumption, a small number of OOD samples can greatly improve the generalization performance. It is worth mentioning that after reaching a certain threshold, more OOD samples will not improve the accuracy. In contrast, using the added 20% samples for training and testing the remaining samples, the accuracy even slightly exceeds the results of the OOD train set + 20% OOD samples. This is because IID samples bring little information gain to the OOD test set classification problem, and may even drag back.

TABLE II. The Test Accuracy of Adding OOD Samples, Tested on ResNet18

train IID	Add OOD	Test
100%	0%	18.808%
100%	2%	46.684%
100%	4%	62.132%
100%	6%	61.507%
100%	8%	71.484%
100%	10%	74.069%
100%	12%	83.008%
100%	14%	83.112%
100%	16%	80.446%
100%	18%	82.188%
100%	20%	84.635%
0%	20%	84.659%

V. Conclusion and Future Works

In this paper, we construct a dataset called CSE, which has various element labels such as rotation angle, object category, distance, etc., which can be used to explore the distribution of each element of the high-informative train set samples, and how to add samples to the OOD test set, which can improve the generalization of the model. We believe that the CSE dataset we constructed can promote the development of active learning interpretability and active learning algorithm design. At the same time, we also believe that analyzing and designing active learning algorithms from the perspective of elements will be a direction of active learning development.

We use the backbone network to obtain the performance of the dataset under the IID and OOD test sets, confirming that the dataset we constructed exhibits IID and OOD distributions. We put forward the conclusion that the active learning model tends to select more differentiated samples. In the IID experiment, after using the active learning algorithm on the IID train set to extract the efficient train set, the rotation angle is divided into three basic types for statistical analysis, which confirms this assertion. We put forward the conclusion that adding OOD samples will greatly improve the generalization performance of the model, and verified this thesis by gradually adding OOD samples for training and testing in the OOD experiment. Among them, the experimental results of IV.B and IV.D also prove that OOD will bring low performance, and even if the model complexity increases, it will not bring noticeable performance improvement. So when creating datasets in various fields, researchers should pay strict attention to how well the training data fits the ground truth or testing data distribution.

However, we also noticed that batch addition of active learning algorithms brings some problems, such as the possibility of similarity between samples added in the same batch. In future work, we will expand the element information (such as pitch angle, background, etc.) of the CSE dataset to establish a more complete dataset for subsequent research.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under grant No.32101612 and No.61871283.

References

[1] A. Diaz-Pinto, N. Ravikumar, R. Attar, A. Suinesiaputra, Y. Zhao, E. Levelt, E. Dall'Armellina, M. Lorenzi, Q. Chen, T. D. Keenan, *et al.*, "Predicting myocardial infarction through retinal scans and minimal personal information," *Nature Machine Intelligence*, pp. 1–7, 2022.

[2] Y. Li, J. Yang, Z. Zhang, J. Wen, P. Kumar, "Healthcare data quality assessment for cybersecurity intelligence," *IEEE Transactions on Industrial Informatics*, 2022.

[3] V. Srivastava, S. Gupta, G. Chaudhary, A. Balodi, M. Khari, V. García-Díaz, "An enhanced texture- based feature extraction approach for classification of biomedical images of ct-scan of lungs," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 7, pp. 18-25, 2021.

[4] J. Nie, Y. Wang, Y. Li, X. Chao, "Sustainable computing in smart agriculture: survey and challenges," *Turkish Journal of Agriculture and Forestry*, vol. 46, no. 4, pp. 550– 566, 2022.

[5] S. Qiu, K. Cheng, T. Zhou, R. Tahir, L. Ting, "An EGG 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.

[6] 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," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol.6, no.7, pp. 101-108, 2021.

[7] Y. Li, J. Yang, "Few-shot cotton pest recognition and terminal realization," *Computers and Electronics in Agriculture*, vol. 169, p. 105240, 2020.

[8] J. Yang, X. Guo, Y. Li, F. Marinello, S. Ercisli, Z. Zhang, "A survey of few-shot learning in smart agriculture: developments, applications, and challenges," *Plant Methods*, vol. 18, no. 1, pp. 1–12, 2022.

[9] Y. Wang, Q. Yao, J. T. Kwok, L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.

[10] Y. Li, J. Yang, "Meta-learning baselines and database for few-shot classification in agriculture," *Computers and Electronics in Agriculture*, vol. 182, p. 106055, 2021.

[11] T. Isomura, T. Toyoizumi, "Dimensionality reduction to maximize prediction generalization capability," *Nature Machine Intelligence*, vol. 3, no. 5, pp. 434–446, 2021.

[12] J. Yang, Y. Zhao, J. Liu, B. Jiang, Q. Meng, W. Lu, X. Gao, "No reference quality assessment for screen content images using stacked autoencoders in pictorial and textual regions," *IEEE transactions on cybernetics*, 2020.

[13] K. Sim, J. Yang, W. Lu, X. Gao, "Mad-dls: mean and deviation of deep and local similarity for image quality assessment," *IEEE Transactions on Multimedia*, vol. 23, pp. 4037–4048, 2020.

[14] J. Yang, A. Li, S. Xiao, W. Lu, X. Gao, "Mtd-net: learning to detect deepfakes images by multi-scale texture difference," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4234–4245, 2021.

[15] M. Bernhardt, D. C. Castro, R. Tanno, A. Schwaighofer, K. C. Tezcan, M. Monteiro, S. Bannur, M. P. Lungren, A. Nori, B. Glocker, *et al.*, "Active label cleaning for improved dataset quality under resource constraints," *Nature Communications*, vol. 13, no. 1, pp. 1–11, 2022.

[16] Y. Li, X. Chao, "Toward sustainability: trade-off between data quality and quantity in crop pest recognition," *Frontiers in plant science*, vol. 12, 2021.
[17] Y. Li, X. Chao, S. Ercisli, "Disturbed-entropy: A simple data quality assessment approach," *ICT Express*, 2022.
[18] S. J. Pan, Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
[19] J. Yang, J. Wen, B. Jiang, H. Wang, "Blockchain- based sharing and tamper-proof framework of big data networking," *IEEE Network*, vol. 34, no. 4, pp. 62–67, 2020.
[20] S. Kornblith, J. Shlens, Q. V. Le, "Do better imagenet models transfer better?," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2661–2671.
[21] G. Kang, L. Jiang, Y. Yang, A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4893–4902.
[22] Z. Hou, B. Yu, Y. Qiao, X. Peng, D. Tao, "Affordance transfer learning for human-object interaction detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 495–504.
[23] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43– 76, 2020.
[24] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[25] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[26] S. Zagoruyko, N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.
[27] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.
[28] Y. Li, X. Chao, "Distance-entropy: An effective indicator for selecting informative data," *Frontiers in Plant Science*, vol. 12, 2022, doi: 10.3389/fpls.2021.818895.
[29] Y. Li, J. Yang, J. Wen, "Entropy-based redundancy analysis and information screening," *Digital Communications and Networks*, 2021, doi: https://doi.org/10.1016/j.dcan.2021.12.001.

Zhuo Zhang

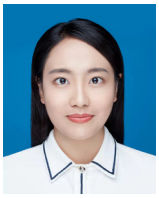
Zhuo Zhang received the M.S. degree in electronic information engineering from Tianjin University, Tianjin, China, in 2021. He is currently pursuing the Ph.D. degree at the school of electrical and information engineering, Tianjin University, Tianjin, China. His research interests include Data Information Assessment and Deep Learning.

Yang Li

Yang Li received the M.S. degree in Electrical Engineering from Dalian University of Technology, China, in 2016. He is currently a Lecturer at Shihezi University, China. His research interests include information processing, pattern recognition, few-shot learning, and data quality assessment. He is an Associate Editor of Plant Methods, Precision Agriculture, and Data Technologies and Applications.

Yicheng Gong

Yicheng Gong received B.S. degree in communication and information engineering from Tianjin University, Tianjin, China, in 2021. He is currently pursuing the M.S. degree at the school of electrical and information engineering, Tianjin University, Tianjin, China. His research interests include Pattern Recognition and Deep Learning.



Yue Yang

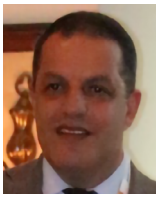
Yue Yang received B.S. degree in electronic information engineering from Tianjin University, Tianjin, China, in 2018. She is currently pursuing the M.S. degree at the school of electrical and information engineering, Tianjin University, Tianjin, China. Her research interests include deep Learning and image information quality evaluation.

Shukun Ma

Shukun Ma received B.S. degree in communication engineering from Tianjin University, Tianjin, China, in 2021. He is currently pursuing the M.S. degree at the school of electrical and information engineering, Tianjin University, Tianjin, China. His research interests include deep Learning and image information quality evaluation.

Xiaolan Guo

Xiaolan Guo received B.S. degree in communication engineering from Tianjin University, Tianjin, China, in 2020. She is currently pursuing the M.S. degree at the school of electrical and information engineering, Tianjin University, Tianjin, China. Her research interests include deep Learning and image information quality evaluation.



Sezai Ercisli

Sezai Ercisli received his PhD degree at Ataturk University, Erzurum, Turkey in 1996. He is currently a Professor at the Department of Horticulture, Agricultural Faculty of Ataturk University in Turkey. His research interests include the modern technology in plant production, such as deep learning and machine learning. He is the Editor-in-Chief of Turkish Journal of Agriculture and Forestry.

COVID-19 Disease Prediction Using Weighted Ensemble Transfer Learning

Pradeep Kumar Roy[1], Ashish Singh[2] *

[1] Department of Computer Science & Engineering, Indian Institute of Information Technology, Surat Gujarat-394190 (India)
[2] School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar-751024, Odisha (India)

Received 28 January 2022 | Accepted 25 January 2023 | Early Access 7 February 2023

Abstract

Health experts use advanced technological equipment to find complex diseases and diagnose them. Medical imaging nowadays is popular for detecting abnormalities in human bodies. This research discusses using the Internet of Medical Things in the COVID-19 crisis perspective. COVID-19 disease created an unforgettable remark on human memory. It is something like never happened before, and people do not expect it in the future. Medical experts are continuously working on getting a solution for this deadly disease. This pandemic warns the healthcare system to find an alternative solution to monitor the infected person remotely. Internet of Medical Things can be helpful in a pandemic scenario. This paper suggested a ensemble transfer learning framework predict COVID-19 infection. The model used the weighted transfer learning concept and predicted the COVID- 19 infected people with an F1-score of 0.997 for the best case on the test dataset.

Keywords

Convolutional Neural Network, COVID-19, Deep Learning, Ensemble Learning, Healthcare,Transfer Learning.

DOI: 10.9781/ijimai.2023.02.006

I. Introduction

With the advancement of computer science technology, healthcare devices are also smart and capable of recording the patient's health information like Blood pressure, Body Temperature, and others with the help of different kinds of sensors. The captured information is further passed to Health experts using connected devices [1], [2], [3]. One Internet of Medical Things (IoMT) scenario is shown in Fig. 1. Different components involved in setting up the IoMT environment are shown, along with a tentative flow of healthcare information between patients and health experts. In the IoMT environment, mainly two components are present- (i) different kinds of sensors and (ii) connective devices. The sensors are used to collect the patient's health information, and networking devices are used to pass the collected information to the concerned medical expert. With IoMT environment, it is possible that multiple patients are being monitored by a single health expert at a time, which is almost impossible in the physical environment [4], [5], [6]. This paper discusses the recent medical issues -coronavirus in detail with a possible framework capable of predicting the COVID-19 with high accuracy.

The World Health Organization (WHO) got the first update on COVID-19 in December 2019 and then declared a public health emergency in January 2020[1]. Coronavirus is a deadly virus that has spread over the world and has been a global health hazard since its inception [7], [8], [9], [10]. This virus first infected animals, then humans. If a human comes into contact with an infected animal, it will become infected. The coronavirus is commutable, which means that if one person becomes sick, all those who come into touch with them may also become infected. The coronavirus can spread through respiratory droplets when someone chats with others, sneezes or coughs.

Many health issues are started in human beings infected with the coronavirus. Such as respiratory failure, liver issues, and others [7], [11], [10]. The virus created both short and long-term health issues. When the person is infected, they are being cured with timely treatment. But, if the infected person has some pre-existing disease, then the chances of being cured are very less.

During the first wave of COVID-19, September 2020 received the highest case at 2,622,328 whereas, in the second wave, it increased to 9,016,561 in May 2021. The USA is the most infected country whereas India is present in the second position. Brazil, UK and Russia stand at the 3rd, 4th and 5th positions.

In India, during the peak of the first wave, i.e., in September 2020, 33,424 deaths were reported, whereas, during the second wave, 131,084 death were reported in a single month of May 2021. These statistics indicate their impact on human beings. Every months thousands of lives are lost due to this virus. Even hospitals are full and unable to occupy the infected persons for treatment. During the second wave of the COVID-19, the shortage of Oxygen in various places was also reported[2,3].

[1] https://www.who.int/emergencies/diseases/novel-coronavirus-2019?

* Corresponding author.
E-mail addresses: pkroynitp@gmail.com (P. K. Roy), ashish.singhfcs@kiit.ac.in (A.Singh).

[2] https://www.dw.com/en/india-covid-oxygen-shortage/a-57425951, [accessed online: 08-10-2021]

[3] https://www.bbc.com/news/world-asia-india-57911638, [accessed online: 08-10-2021]

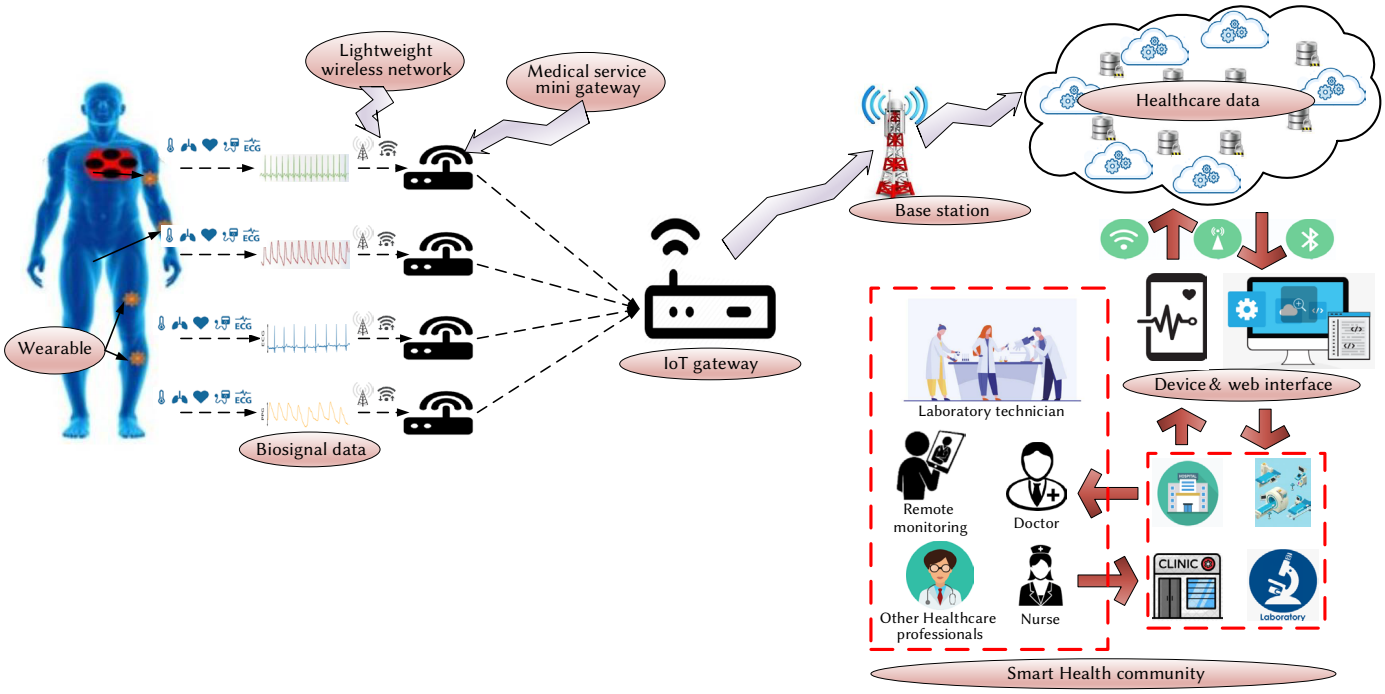


Fig. 1. Framework of Internet of Medical Things Scenario.

Currently, the test name called reverse transcription- polymerase chain reaction (RT-PCR) is used to check whether the person is infected or not from COVID-19. However, RT- PCR test kits take around 4-6 hours to provide the test result [12]. Meanwhile, many people may get infected by each other. The 4-6 hours time to get the result about the infection is an issue to stop it from spreading. The other alternatives that were developed recently include examining CT-scan [9], [13], Chest X-ray [8], [11], [14], reports and other symptoms of suspicious people. Many models reported in the literature have False Positive prediction issues, which means the test result is positive even though people are not infected.

The poor prediction rate and false prediction may have multiple reasons. One reason is less number of actual health reports of the infected person. This research developed a weighted ensemble transfer learning model for COVID-19 detection. The proposed model aims to minimize false-positive prediction cases. The model works in two folds: first, the Chest X-ray images are processed and passed to the pre- trained transfer model. Based on the training performances, the model's weights are decided. Secondly, the outcomes of the three transfer learning models are weighted and ensemble to get the final results. Finally, the developed model can be embedded with an application for real-time COVID-19 predictions. The major contributions of this research include the following:

- A weighted ensemble transfer learning framework is developed in this research for predicting COVID-19 pa- tients.
- The proposed model uses only the Chest X-ray images for training purposes and achieves high accuracy with a minimal false positive prediction rate. The model can be embedded in the portable application for fast COVID-19 prediction.

The rest of the paper is organized as follows: Section II discusses the relevant work related to our objective. In Section III, the model overview is discussed. Section IV discusses a proposed methodology in detail. In Section V, the outcomes of the individual and ensemble learning models are presented, and finally, Section VI concludes this work.

II. Literature Review

This section discusses existing works for the detection of coronavirus disease. Most recent research uses the ben- efits of transfer learning models to design the COVID-19 predictive framework [14], [15], [16], [17]. In [18], Chest X-ray radiographs images are used to build the model for coronavirus detection. They have used multiple pre-trained Convolutional Neural Network (CNN) models during the experiment and achieved a 99.70% classification performance using the ResNet50 model. A COVID-19 detection technique has been proposed in [19] using deep learning and transfer learning schemes. They have used X-rays and CT-scan images that are collected from many sources. Their modified CNN model achieved 94.10% accuracy, whereas, with a pre-trained model, 98% accuracy was achieved. A binary and multi-class classification COVID-19 detection mechanism has been proposed in [20]. The experiment was performed on raw Chest X-ray images. Their model achieved the classification accuracy 98.08% for binary classes and 87.02% for multi-class classification of COVID-19 patients.

Another model proposed using the Chest X-ray images for COVID-19 detection in [16]. The authors used an open source dataset for the model development. To develop detection mechanisms, the model uses VGG-16, OxfordNet with Faster Regions CNN. Their model achieved 97.36% accuracy for the best case. In [17], a CoroNet model for detection of COVID- 19 infection has been proposed. Their model was based on Xception architecture and pre-trained on the ImageNet Chest X-ray dataset. In [21], three deep learning-based s were used for the detection and diagnosis of COVID-19 cases. Deep neural network based CNN s are applied to the X-ray images of lungs to diagnose the disease. The results with the CNN model show maximum accuracy of 93.20%. A deep learning technique based on CNN and LSTM has been proposed in [22] for the diagnosis of COVID-19 disease. In this technique, feature extraction was handled by the CNN and detection of the disease is handled by the LSTM model using extracted features. In [8], fine-tuned deep learning techniques have been proposed to speed up the detection and classification of COVID-19 disease. The research was conducted on two different datasets containing 959 X-ray images. DenseNet121 shows higher accuracy, 97% for the first

dataset, and MobileNetV2 has 81% for the second dataset. In [23], authors proposed a diagnosis model for COVID-19 disease. They have used ResNet-50 and DenseNet-201 pre- trained networks for feature extraction and backpropagation neural network model to classify the results into multiple levels. Their model achieved 98.50% accuracy.

In [24], authors proposed an automated technique for COVID-19 detection by using CT-scan images. Their ET- NET model has been evaluated on publicly available data using deep learning techniques. Two different approach used for detection of COVID-19 infection by [25]. Firstly, they used Artificial Neural Networks (ANN) and then Bidirectional Long Short-Term Memories (Bi-LSTM) model was used to design the proposed hybrid architecture. A modified ensemble deep learning models with the inclusion of extra layers has been proposed in [26]. For binary classification model their model achieves 99.49% accuracy and whereas for multi-class, the accuracy value was 99.24%. Machine learning (ML) and transfer learning s based COVID-19 detection model has been proposed in [27]. They have used 5,480 samples having two classes for their experimental purpose. Another work [28], uses a multi-stage transfer learning technique for diagnosis of COVID-19 using CT-scan images with 86.70% accuracy. In [29], Lung CT-scan images are used for COVID- 19 infection detection using ensemble classifier. The proposed detection model uses a transfer learning approach with eight different pre-trained models. In [10], authors used an ensemble transfer learning models to build a COVID-19 detection model. They used Chest X-ray images to design the COVID-19 detection framework with pre-trained transfer learning models. The researchers have suggested multiple frameworks to address COVID-19 detection issues and have achieved good accuracy. However, one of the limitations of the existing research includes the dataset used for model development. This research uses a comparatively larger dataset to develop the model. Also used a novel weight assignment approach to build the ensemble framework. The model can be embedded with devices having Internet connectivity. This enables remote monitoring facilities in the healthcare domain and limits the crisis of health experts.

III. Model Overview

This section mainly highlights the configuration and work- ing of the Transfer Learning (TL) models and ensemble frameworks. First, the working of TL models are explained in detail, and then the reason behind selecting the few TL model will be discussed. At last, the working of the ensemble learning framework and weight generation process are explained in detail.

A. Transfer Learning

Along with technological developments, training deep learn- ing neural networks in recent years received many advance- ments. The main reason behind using the concept of TL is to utilize the existing complex and successful pre-trained models [30] learned from a huge data corpus, viz., (ImageNet model trained with 1000 categorical dataset) and transfer the learnt knowledge [31] to the simple task like binary image classifi- cation (COVID-19 Positive, COVID-19 Negative) having less number of data samples. The labelled data can achieve the optimal mapping of images, labels, and sentences. However, the issues of generalizability are still observed when the model is used with unseen and different datasets.

Mathematically, if ImageNet TL has input data (I_s), the labels (L_s) have 1000 categories, and their corresponding output, i.e., the trained classifier, will be represented using O_s.

The knowledge of the transfer learning can be represented as in (1).

$$S = \{X_s, L_s, O_s\} \tag{1}$$

Next, the aim is to utilize the source knowledge for building a new model with target input data X_t, labels L_t (COVID-19 Positive, COVID-19 Negative), and the model O_t (see (2).

$$T = \{X_t, L_t, O_t\} \tag{2}$$

The classifier or model built by utilizing the knowledge of TL can be written as $O_t(X_t, L_t|S)$, whereas a model built without using the TL can be written as $O_t(X_t, L_t)$. It can be represented as in (3).

$$O_t = \begin{cases} O_t(X_t, L_t|S) & \text{with TL concept} \\ O_t(X_t, L_t) & \text{without TL concept} \end{cases} \tag{3}$$

The model built by utilizing the TL concept, i.e., ($O_t(X_t, L_t|S)$ is probably more suitable for the said problem than the model developed without using the TL concept. This means assuming the large input sample X has labels L. The error e is assumed to be lesser with the TL concept-based model.

$$error[O_t(X_t, L_t|S), (X), L] < error[O_t(X_t, L_t), (X), L] \tag{4}$$

where *error* (4) is a function used to calculate the error for input values, the pre-trained model generally helps the user save time, efficiency, and resources as tuning the parameter may not be necessary. The pre-trained models help extract the low-level features from the input images, such as shades and tints. The target model needs only to tune the parameter of the last few last layers.

Instead of directly using the transfer learning models and their weights in the proposed methodology, a weight gen- eration function architecture is used to assign weights to respective models. After referencing several research papers, we found that the transfer learners ResNet50V2, DenseNet201 and InceptionV3 have performed exceptionally well in image classification challenges. Hence, these three individual models have been used and explored for the study.

B. Ensemble Learning

The ensemble approach involves combining the predictive power of various learners to improve the overall performance and robustness of the model. The error in the predictive ca- pacity of a model can be decomposed into three errors—bias, variance, and variance of the irreducible error of the model. The model error can be described as Model-Error which is a collective sum of errors obtained from bias, variance, and irreducible error.

The term Bias error is used to describe how much the expected values vary from the actual value on average. A high bias results from under-performance where the model misses important trends during the training phase. Variance indicates the predictive capacity of a model on the same observation. A model tends to overfit with high variance and would perform worse in the validation data set. Various ensemble approaches have been proposed to address this bias-variance trade-off. The optimal scenario is to reach a minimum bias error and a minimum variance error. The reducible error (Bias error and Variance error) is the element that can be improved. We reduce the quantity when the model learns on a training dataset. We attempted to get this quantity as close to zero to reduce the overall error in the model's predictive capacity. The fundamental error is the error that can not be removed. The error is generated because of noise in the observations or outliers in the data set.

The novel contribution to the research is that the dif- ferent transfer learning architectures were assigned weights depending upon their predictive accuracy over a dataset. The ensemble version is the weighted average of the individual transfer learning-based model's performances obtained with the test dataset.

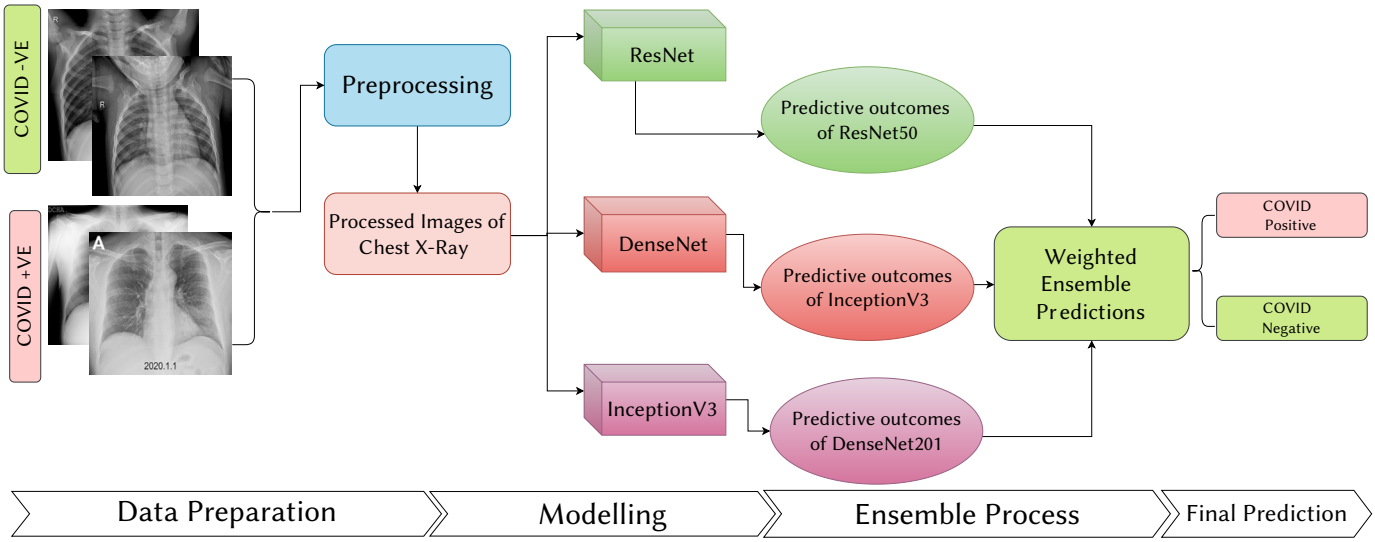


Fig. 2. An overview of the proposed weighed ensemble transfer learning frameworks.

IV. Proposed Methodology

The proposed deep ensemble TL framework works two-fold. Firstly, the TL models were to train the data set, i.e., Chest X- ray (CXR) images and generate the corresponding model files. Secondly, to test the real-time inference of the proposed model, 5-Fold cross-validation was used on the validation set, and the corresponding accuracy and loss function graphs were plotted to check the model's performance over increasing epochs on unseen data. Finally, the model files were used to predict the respective Chest X-ray images and their probabilities of belonging to that class. Fig. 2 shows the overview of the working steps involved in the proposed ensemble framework Fig. 3 shows the detailed steps of the proposed models, like the weight generation steps, the data used for the 5-fold cross- validation technique, the metrics used for the evaluation of the model and others.

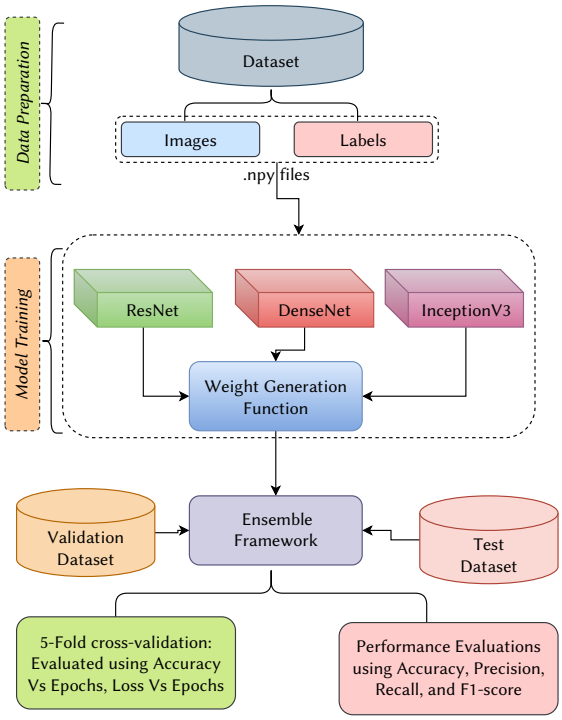


Fig. 3. Detailed steps involved in the proposed weighed ensemble transfer learning framework.

The data set used in this research was collected from https://www.kaggle.com/tawsifurrahman/covid19-radiography-database, which has collated images from various sources. Considering the computational resource restraints, a part of the data set was used for the experiment consisting of 2000 images. The model classified the image into two classes (i) "COVID-19 Positive" and (ii) "COVID-19 Negative". A total of 2000 Chest X-ray images were used for the experiment. The detailed break up of the number of images in the data set has been shown in Table I. For optimal model performance, the training and validation sets have been split into 60%, 20%, and 20% ratios, respectively.

TABLE I. Data Distributions Used for Model Training, Validation and Testing Purpose

Class	Train	Test	Validation	Total
COVID-19 Positive	600	200	200	1000
COVID-19 Negative	600	200	200	1000
Total	1200	400	400	2000

The images correspond to the "COVID-19 Positive" class, and the "COVID-19 Negative" class was converted to ".npy" files and their appropriate labels. The ".npy" file format is NumPy's basic binary file format for storing a single NumPy array on a disk. This way, the shape and the data type information necessary to create the array on a system with different architecture remains intact. This process leads to faster processing of data. All of the images from the source were 299x299 pixels in nature, which had been converted to an appropriate size of 224x224 pixels to suit the respective transfer learning architecture. The code structure appends the data and the images according to their respective classes and labels and stores it in a standard binary format, the ".npy" file. Threading was used for the parallel execution and utilized the multiprocessing capacity to optimal. 1 discusses the complete working steps of data preparation.

A. Model Training

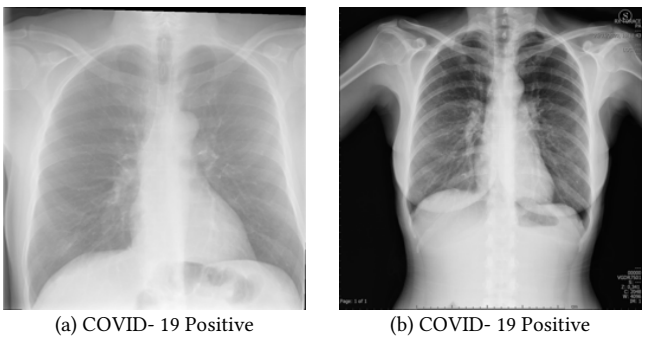
Three pre-trained TL models, (i) ResNet50V2, (ii) DenseNet201 and (iii) InceptionV3, were used during the training phase proposed model. The models were trained with 200 epochs, and the learning rate was fixed to 0.001, with a batch size of 32. The callback function was explicitly used to monitor the model performance by fixing the patience value of 5. This means, the model will stop training for further epochs if they encounter no improvement in the validation accuracy in five consecutive epochs. The model was compiled using

TABLE II. Details of the Parameters Used for Model Development With ResNet50V2, DenseNet201, and InceptionV3

Parameters	ResNet50V2	DenseNet201	InceptionV3
Learning Rate	0.001	0.001	0.001
Batch Size	32	32	32
Number of epochs trained	20	35	19
Loss Function	Sparse Categorical Cross Entropy	Sparse Categorical Cross Entropy	Sparse Categorical Cross Entropy
Training Time	287 seconds	987 seconds	310 seconds
Accuracy on Test Data set	98.75%	99.75%	98.25%
Weights assigned	0.75235685	0.76147539	0.7513245

Algorithm 1. Data preparation for model training

1: Input: Raw Images
2: Output: *.npy* files
3: **BEGIN**
4: create data()
5: data = [ ]
6: **for** `category` **in** `categories` **do**
7: a. path = location of the files
8: b. class num = categories.index(category)
9: c. files = loaded from path
10: d. total = number of files
11: e current = 1
12: **for** `img in files`: **do**
13: 1. **try:**
14: a. img array = read image from path
15: b. img array = resize the image
16: c. data.append([img array,class num])
17: 2. **except** Exception **as** e:
18: a. pass
19: 3. current += 1
20: random.shuffle(data)
21: images = [ ] 22: classes = [ ] 23: current = 1
24: **for** `image, cls` **in** `data`: **do**
25: a. images.append(image)
26: b. classes.append(cls)
27: c. current += 1
28: images = np.array(images).reshape(-1,image size, im- age size,3)
29: images = images/255.0
30: classes = np.array(classes) #Preparing .npy files
31: np.save(save filename images,images)
32: np.save(save filename labels,classes)
33: **END**



(a) COVID- 19 Positive (b) COVID- 19 Positive

Fig. 4. Chest X-ray of COVID-19 Positive patient.

Adam Optimizer, with the exponential decay rate for the first moment being 0.90 and for the second moment being 0.999. The ensemble function calculates the weighted average of predicted models based on their weights, respectively. The individual weights were assigned based on predictive accuracy. A sigmoid function was applied to it to give the probability of occurrence.

The performance of the trained model on an unseen dataset was evaluated using the performance metrics like F1-score, Precision and Recall. The values of these metrics are calcu- lated using the parameter of the confusion metrics, such as true positive (TP), false positive (FP), true negative (TN), and false-negative (FN). Mathematically, these metrics are defined in Eqs. (5)-(8):

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1 - score = \frac{Precision * Recall}{Precision + Recall} \tag{7}$$

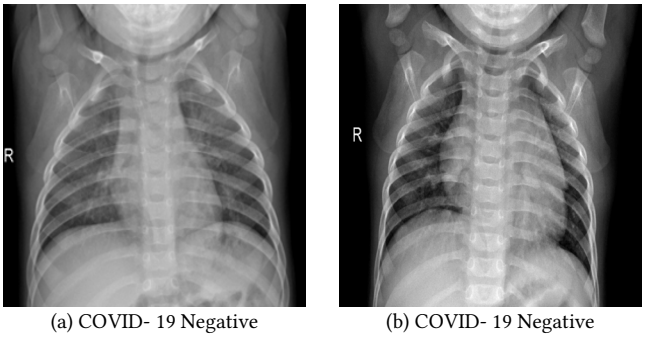
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

V. Results

This section discusses the experimental outcomes of the pro- posed deep ensemble TL models in different hyper-parameter settings. As discussed, a weight generation function was used to generate the weight of the individual TL models based on their prediction accuracy. The weights are generated using the sigmoid function, resulting in the class probabilities in [0-1]. Table III consists of the name of the models, the time taken for training, the accuracy obtained on the test dataset and the weights generated by the weight generation function.

TABLE III. Individual Model's Accuracy and Corresponding Weights Defined by Weight Generation Function

Parameters	ResNet50V2	DenseNet201	InceptionV3
Training Time	279 seconds	899 seconds	260 seconds
Accuracy on Test Data set	98.75%	99.75%	98.25%
Weights assigned	0.75235685	0.76147539	0.7513245

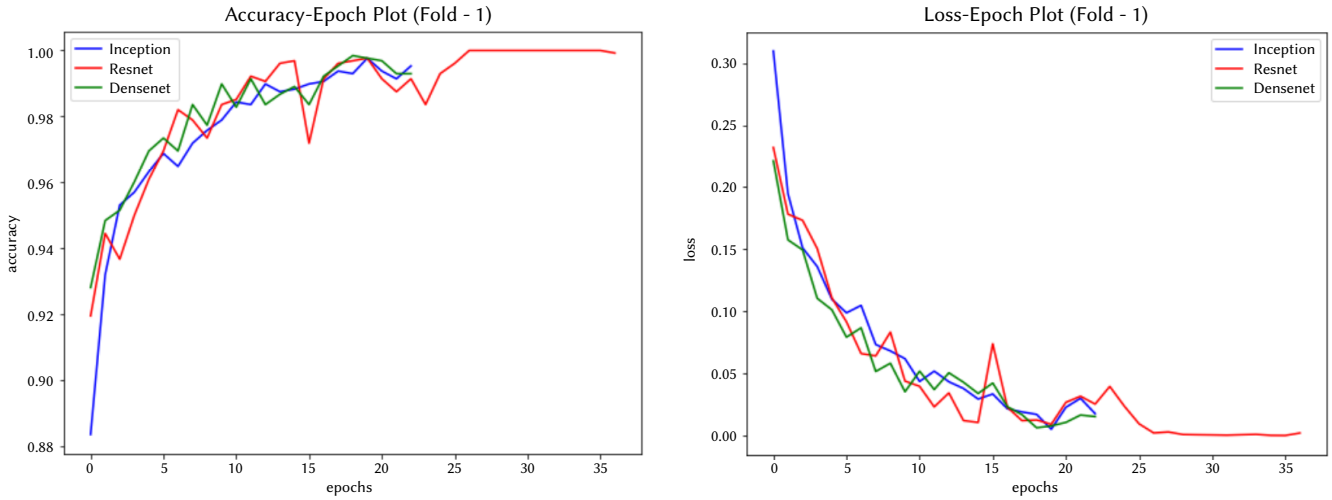


(a) COVID- 19 Negative (b) COVID- 19 Negative

Fig. 5. Chest X-rays of COVID-19 Negative patient.

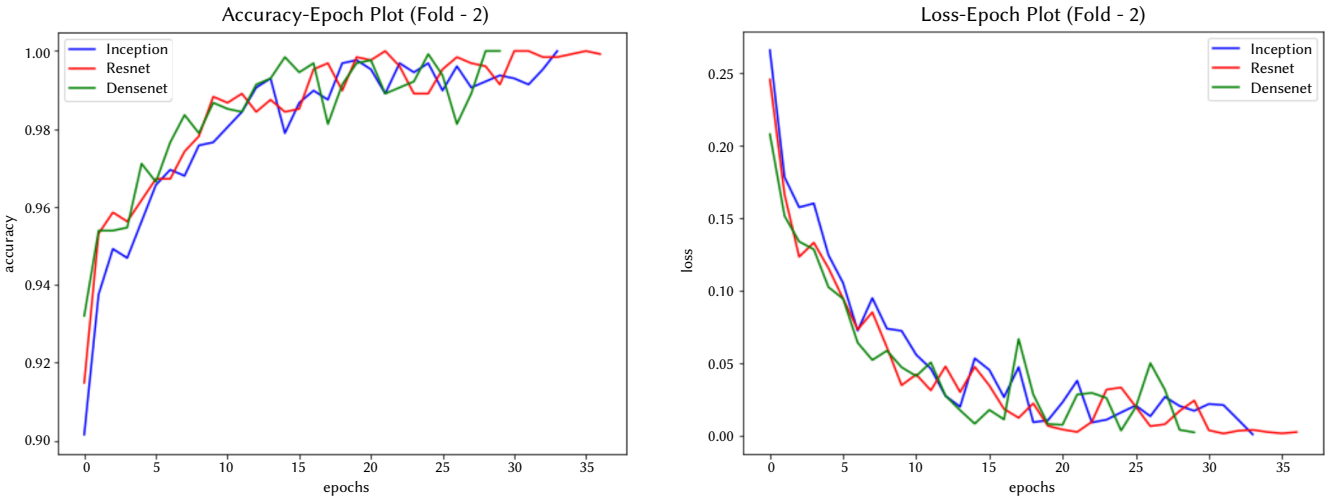
TABLE IV. The Performance Obtained With ResNet50V2, DenseNet201, and InceptionV3 in Different K-Folds

Fold No.	Fold 1			Fold2			Fold 3		
Parameters	**ResNet50V2**	**DenseNet201**	**InceptionV3**	**ResNet50V2**	**DenseNet201**	**InceptionV3**	**ResNet50V2**	**DenseNet201**	**InceptionV3**
No of epochs actually trained	37	23	23	37	30	34	19	29	25
Train Time	301	365	221	301 seconds	435 seconds	279 seconds	157 seconds	454 seconds	208 seconds
Accuracy on validation set	96.25%	93.25%	97.50%	98.00%	97.50%	96.25%	84.00%	98.25%	95.25%
Weights assigned	0.72237036	0.72237036	0.72487028	0.72424661	0.72487028	0.72611498	0.7051357	0.72048628	0.72237036
Ensemble Model's Accuracy	97.25%			97.75%			97.00%		
Fold No.	Fold 4			Fold 5					
Model	**ResNet50V2**	**DenseNet201**	**InceptionV3**	**ResNet50V2**	**DenseNet201**	**InceptionV3**			
No of epochs actually trained	27	28	19	29	31	37			
Train Time	221 seconds	439 seconds	159 seconds	238 seconds	485 seconds	303 seconds			
Accuracy on validation set	97.75%	96.50%	96.25%	97.75%	97.25%	97%			
Weights assigned	0.72174321	0.72237036	0.72299665	0.73044372	0.73044372	0.72921135			
Ensemble Model's Accuracy	98.00%			97.75%					



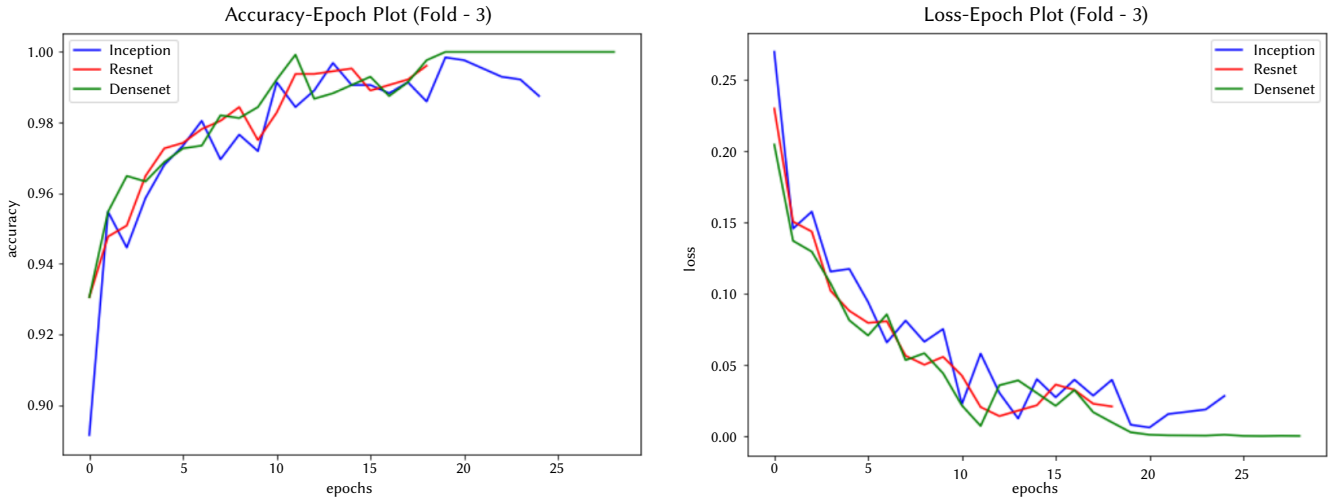
(a) Accuracy vs Epochs (b) Loss vs Epochs

Fig. 6. The plot of accuracy and losses obtained with different epochs for Fold 1 of the cross-validation technique.



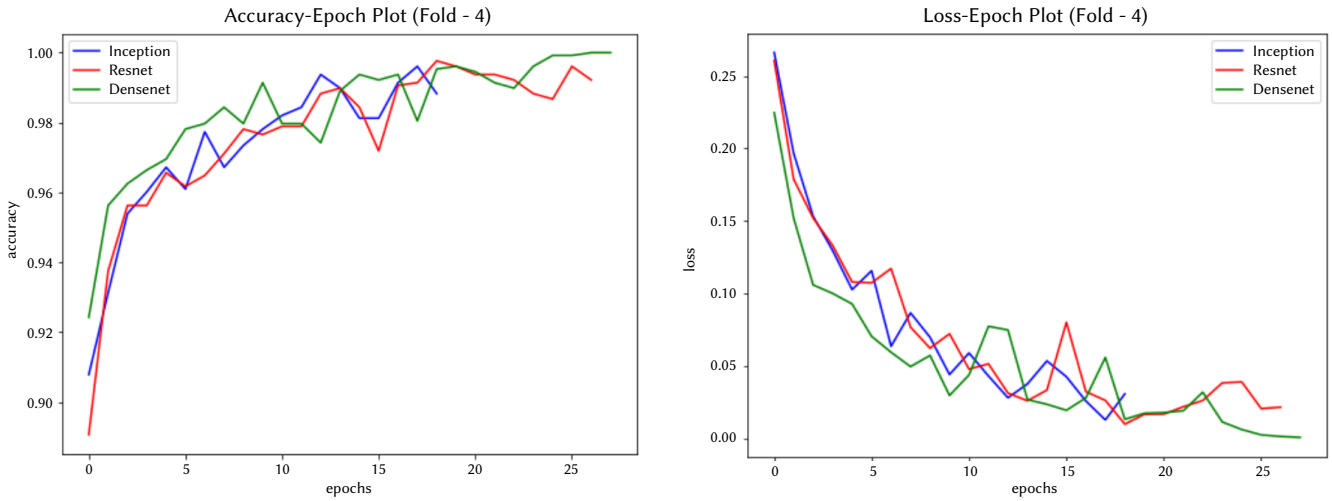
(a) Accuracy vs Epochs (b) Loss vs Epochs

Fig. 7. The plot of accuracy and losses obtained with different epochs for Fold 2 of the cross-validation technique.



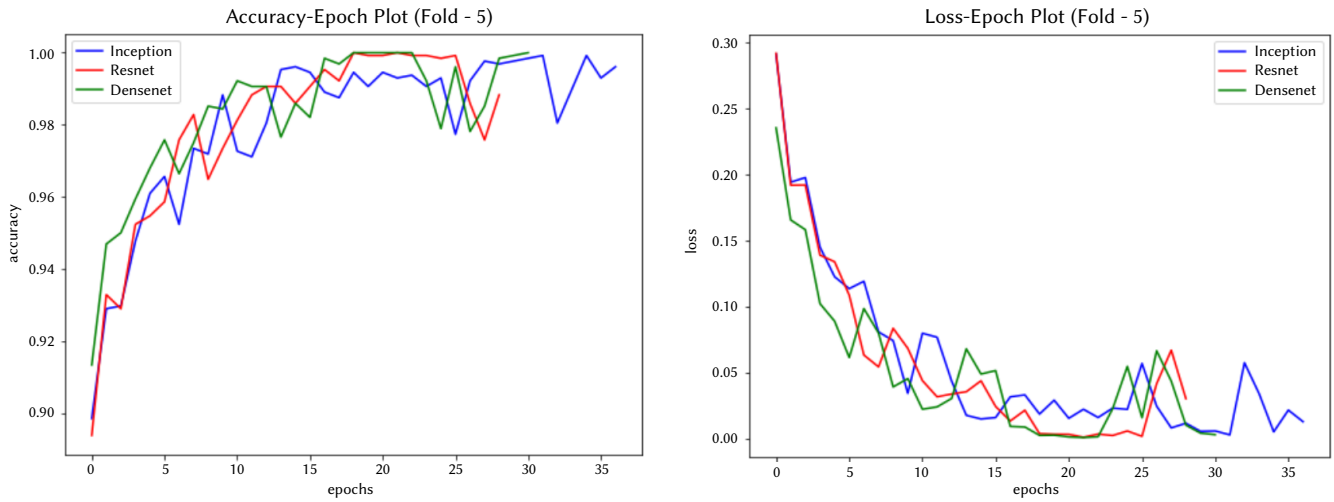
(a) Accuracy vs Epochs (b) Loss vs Epochs

Fig. 8. The plot of accuracy and losses obtained with different epochs for Fold 3 of the cross-validation technique.



(a) Accuracy vs Epochs (b) Loss vs Epochs

Fig. 9. The plot of accuracy and losses obtained with different epochs for Fold 4 of the cross-validation technique.



(a) Accuracy vs Epochs (b) Loss vs Epochs

Fig. 10. The plot of accuracy and losses obtained with different epochs for Fold 5 of the cross-validation technique.

TABLE V. Performance of Proposed Ensemble Model on Manually Weighted and Weighted With a Function

Models	DenseNet	ResNet	InceptionV3	TP	TN	FP	FN	Precision	Recall	F1
Manual Weight Assignment	0.10	0.60	0.30	200	196	0	4	0.990	0.980	0.990
	0.50	0.20	0.30	200	195	0	5	0.995	0.990	0.995
	0.10	0.80	0.10	200	195	0	5	0.987	0.976	0.987
	0.10	0.10	0.80	200	194	0	6	0.985	0.970	0.985
Weighted by function	0.75235685	0.76147539	0.7513245	200	197	0	3	0.997	0.995	**0.997**

TABLE VI. Performance Comparison With Existing Research

Models	Number of samples	Types of data	Technique	Precision	Recall	F1-score
Islam et al. [22]	1525	CXR	CNN, LSTM	–	0.993	0.989
Khan et al. [17]	284	CXR	DCNN	0.93	0.982	–
Ozturk et al. [20]	127	CXR	DCNN	0.98	0.951	0.965
Shibly et al. [16]	283	CXR	RCNN	0.9929	0.976	0.984
Hassantabar et al. [21]	200	CT	CNN and DNN	–	0.960	–
Maghdid et al. [19]	431	CT and CXR	CNN and Transfer Learning	–	0.960	–
Proposed	**2000**	**CXR**	**Ensemble Transfer Learning**	**0.997**	**0.995**	**0.997**

A. Performance With K-Fold Setting

To check the model's performance on unseen data using the K-fold setting. The experiment was repeated with the same set of TL models. The value of K was fixed to 5. In K-Fold cross- validation, the data is split into K folds. K-1 folds act as a train- ing set for each iteration, whereas the remaining fold is a test set. This results in less bias and better performance because each observation in the original data set is optimally used.

The performance of the K-fold cross-validation technique was checked over an increasing number of epochs. The hyper- parameters values remain the same as the previous setting as discussed in section IV-A. In the proposed model, each transfer learner was individually trained for all the 5-Folds, and a corresponding ensemble accuracy was calculated for each fold. Table IV shows the experimental outcomes of K-fold settings with the value of the used parameter. The accuracy value obtained on different K-folds, i.e., Fold 1, Fold 2, Fold 3, Fold 4, and Fold 5, are 97.50%, 97.75%, 97.00%, 98.00%, and 97.75%, respectively. The accuracy value for the different K-fold settings lies between 97.00% to 98.00%. For each fold, *Accuracy vs Epoch*, and *Loss vs Epoch* graphs are plotted to check how the model performs on the validation data (unseen). The plots are shown in Fig. 6 to Fig. 10.

B. Performance Comparison With Manual Weight Assignments

As shown in Table V, the performance of the proposed weighted ensemble TL framework achieved satisfactory performance. To verify the weights generated by the function, we have manually assigned the weights to the individual TL mod- els as shown in Table V and checked their performances. In the first case, the weight of DenseNet, ResNet, and InceptionV3 was 0.10, 0.60, 0.30. The model yielded precision, recall, and F1-score value of 0.990, 0.980, and 0.990, respectively. In the second case, the weight of DenseNet, ResNet, and InceptionV3 was 0.50, 0.20, 0.30. The model yielded precision, recall, and F1-score values of 0.995, 0.990, and 0.995. In the third case, the weight of DenseNet, ResNet, and InceptionV3 was 0.10, 0.80, 0.10. The model yielded precision, recall, and F1-score values of 0.987, 0.976, and 0.987. In the fourth case, the weight of DenseNet, ResNet, and InceptionV3 was 0.10, 0.10, 0.80. The model yielded precision, recall, and F1-score values of 0.985, 0.970, and 0.99.

Finally, the model's performance with the manual weight assignment technique was compared with the performance ob- tained using automated weight assignment techniques, where the precision, recall, and F1-score were 0.997, 0.995, and 0.997, respectively. The proposed automated weight assign- ment technique helps achieve better performance. The con- fusion matrix's parameter values were as follows: the true positive (TP) is 197, FP is 0, TN is 197, and FN is 3, indicating that the model misclassifies only three samples among the total test sample. The confusion matrix obtained using the best model setting is shown in Fig. 11.

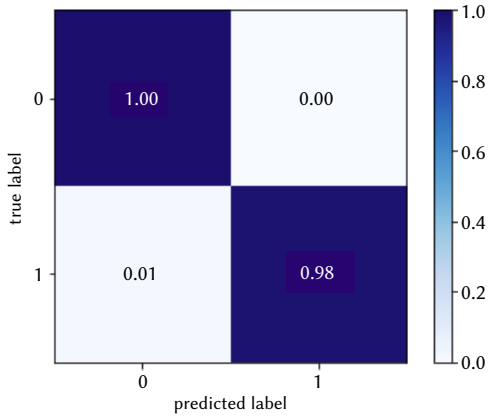


Fig. 11. Confusion matrix obtained using the best model.

C. Performance Comparison With State-Of-The-Art

The experimental outcomes of the proposed weighted en- semble TL model were compared with the existing deep learning-based COVID-19 predictive models in Table VI. It can be seen that the proposed weighted ensemble TL models outperformed existing research. As compared to the listed research (Islam et al. [22], Khan et al. [17], Ozturk et al. [20], Shibly et al. [16], Hassantabar et al. [21], Maghdid et al. [19]), this research consider more number of data CXR samples to train and validate the model. Among the existing researchers, Islam et al. [22] achieved the best prediction performance, where the recall value was 0.993 and F1-score was 0.989. However, the proposed model achieved the F1-score of 0.997, outperforming the existing research.

VI. Conclusion

The healthcare industry is equipped with the latest tech- nologies to provide the best treatment for patients suffering from critical diseases. Recent technology like the Internet of Things (IoT) can also be used for various purposes in the medical field, like remote monitoring of patient health. This study suggested an Internet of

Medical Things-assisted frame- work to fulfil medical experts' needs during the COVID-19 pandemic. The proposed framework uses the weighted average of predictive accuracy of individual transfer learning models, namely ResNet50V2, DenseNet201 and InceptionNetV3. The ensemble learning framework uses the individual strength of the transfer learners to detect COVID-19 from the Chest X-ray images. The model performs well on the validation data set, which can be observed from the results of the 5-Fold cross- validation. The IoMT based model helps predict and monitor COVID-19 patients remotely with the embedded application. The proposed model's performance can be improved by using the Regularization techniques such as Data Augmenta- tion and Generative Adversarial Networks. Despite its heavy computational requirements and complex structure, this framework is practical enough to provide optimal results on the validation data set. The dataset used in this research can be extended using some preprocessing techniques. The model works on the posterior-anterior (PA) view of X-Rays. Hence it can not differentiate anterior-posterior (AP), lateral views etc. The problem can be further extended to predict the mild and severe cases of COVID-19. This would reduce the load on the existing healthcare infrastructure. Also, there is a need for efficient radiologists to identify and confirm the results of the proposed model.

Declaration of Competing Interest

There is no Conflict of Interest.

References

[1] P. Verma and S. K. Sood, "Fog assisted-iot enabled patient health monitoring in smart homes," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1789–1796, 2018.

[2] M. Kang, E. Park, B. H. Cho, and K.-S. Lee, "Recent patient health monitoring platforms incorporating internet of things-enabled smart devices," *International neurourology journal*, vol. 22, no. Suppl 2, p. S76, 2018.

[3] O. Taiwo and A. E. Ezugwu, "Smart healthcare support for remote patient monitoring during covid-19 quarantine," *Informatics in medicine unlocked*, vol. 20, p. 100428, 2020.

[4] D. V. Dimitrov, "Medical internet of things and big data in healthcare," *Healthcare informatics research*, vol. 22, no. 3, pp. 156–163, 2016.

[5] W. Sun, Z. Cai, Y. Li, F. Liu, S. Fang, and G. Wang, "Security and privacy in the medical internet of things: a review," *Security and Communication Networks*, vol. 2018, 2018.

[6] F. Hu, D. Xie, and S. Shen, "On the application of the internet of things in the field of medical and health care," in *2013 IEEE international conference on green computing and communications and IEEE Internet of Things and IEEE cyber, physical and social computing*. IEEE, 2013, pp. 2053–2058.

[7] S. Chahar and P. K. Roy, "Covid-19: A comprehensive review of learning models," *Archives of Computational Methods in Engineering*, vol. 29, p. 1915–1940, 2021.

[8] S. Aggarwal, S. Gupta, A. Alhudhaif, D. Koundal, R. Gupta, and K. Polat, "Automated covid-19 detection in chest x-ray images using fine-tuned deep learning architectures," *Expert Systems*, p. e12749.

[9] S. Shambhu, D. Koundal, P. Das, and C. Sharma, "Binary classification of covid-19 ct images using cnn: Covid diagnosis using ct," *International Journal of E-Health and Medical Communications (IJEHMC)*, vol. 13, no. 2, pp. 1–13, 2021.

[10] P. K. Roy and A. Kumar, "Early prediction of covid-19 using ensemble of transfer learning," *Computers and Electrical Engineering*, vol. 101, p. 108018, 2022.

[11] S. Tabik, A. Gómez-Ríos, J. L. Martín-Rodríguez, I. Sevillano-García, M. Rey-Area, D. Charte, E. Guirado, J. L. Suárez, J. Luengo, M. A. Valero-González, P. García-Villanova, E. Olmedo-Sánchez, and F. Herrera, "Covidgr dataset and covid-sdnet methodology for predicting covid-19 based on chest x-ray images," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 12, pp. 3595–3605, 2020.

[12] A. Bernheim, X. Mei, M. Huang, Y. Yang, Z. A. Fayad, N. Zhang, K. Diao, B. Lin, X. Zhu, K. Li *et al.*, "Chest ct findings in coronavirus disease-19 (covid-19): relationship to duration of infection," *Radiology*, p. 200463, 2020.

[13] A. Hiremath, K. Bera, L. Yuan, P. Vaidya, M. Alilou, J. Furin, K. Armitage, R. Gilkeson, M. Ji, P. Fu, A. Gupta, C. Lu, and A. Madabhushi, "Integrated clinical and ct based artificial intelligence nomogram for predicting severity and need for ventilator support in covid-19 patients: A multi-site study," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 11, pp. 4110–4118, 2021.

[14] V. Dwivedy, H. D. Shukla, and P. K. Roy, "Lmnet: Lightweight multi-scale convolutional neural network architecture for covid-19 detection in iomt environment," *Computers and Electrical Engineering*, vol. 103, p. 108325, 2022.

[15] A. Kumar, P. K. Roy, and J. P. Singh, "Bidirectional encoder representations from transformers for the covid-19 vaccine stance classification," in *Working Notes of FIRE-13th Forum for Information Retrieval Eval- uation, FIRE-WN 2021*, 2021, pp. 1216–1220.

[16] K. H. Shibly, S. K. Dey, M. T.-U. Islam, and M. M. Rahman, "Covid faster r–cnn: A novel framework to diagnose novel coronavirus disease (covid-19) in x-ray images," *Informatics in Medicine Unlocked*, vol. 20, p. 100405, 2020.

[17] A. I. Khan, J. L. Shah, and M. M. Bhat, "Coronet: A deep neural network for detection and diagnosis of covid-19 from chest x-ray images," *Computer Methods and Programs in Biomedicine*, vol. 196, p. 105581, 2020.

[18] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks," *Pattern Analysis and Applications*, pp. 1–14, 2021.

[19] H. S. Maghdid, A. T. Asaad, K. Z. Ghafoor, A. S. Sadiq, S. Mirjalili, and M. K. Khan, "Diagnosing covid-19 pneumonia from x-ray and ct images using deep learning and transfer learning s," in *Multimodal Image Exploitation and Learning 2021*, vol. 11734. International Society for Optics and Photonics, 2021, p. 117340E.

[20] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. R. Acharya, "Automated detection of covid-19 cases using deep neural networks with x-ray images," *Computers in biology and medicine*, vol. 121, p. 103792, 2020.

[21] S. Hassantabar, M. Ahmadi, and A. Sharifi, "Diagnosis and detection of infected tissue of covid-19 patients based on lung x-ray image using convolutional neural network approaches," *Chaos, Solitons & Fractals*, vol. 140, p. 110170, 2020.

[22] M. Z. Islam, M. M. Islam, and A. Asraf, "A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images," *Informatics in medicine unlocked*, vol. 20, p. 100412, 2020.

[23] A. Aswathy, A. Hareendran, and V. C. SS, "Covid-19 diagnosis and severity detection from ct-images using transfer learning and back propagation neural network," *Journal of Infection and Public Health*, vol. 14, no. 10, pp. 1435–1445, 2021.

[24] R. Kundu, P. K. Singh, M. Ferrara, A. Ahmadian, and R. Sarkar, "Et-net: an ensemble of transfer learning models for prediction of covid-19 infection through chest ct-scan images," *Multimedia Tools and Applications*, vol. 81, no. 1, pp. 31–50, 2022.

[25] M. F. Aslan, M. F. Unlersen, K. Sabanci, and A. Durdu, "Cnn- based transfer learning–bilstm network: A novel approach for covid-19 infection detection," *Applied Soft Computing*, vol. 98, p. 106912, 2021.

[26] F. Altaf, S. Islam, and N. K. Janjua, "A novel augmented deep transfer learning for classification of covid-19 and other thoracic diseases from x-rays," *Neural Computing and Applications*, vol. 33, no. 20, pp. 14 037–14 048, 2021.

[27] S. M. Rezaeijo, M. Ghorvei, R. Abedi-Firouzjah, H. Mojtahedi, and H. E. Zarch, "Detecting covid-19 in chest images based on deep transfer learning and machine learning s," *Egyptian Journal of Radiology and Nuclear Medicine*, vol. 52, no. 1, pp. 1–12, 2021.

[28] J. F. H. Santa Cruz, "An ensemble approach for multi-stage transfer learning models for covid-19 detection from chest ct scans," *Intelligence-Based Medicine*, vol. 5, p. 100027, 2021.

[29] N. S. Shaik and T. K. Cherukuri, "Transfer learning based novel ensemble classifier for covid-19 detection from chest ct-scans," *Computers in Biology and Medicine*, vol. 141, p. 105127, 2022.

[30] R. Glatt, F. L. Da Silva, R. A. da Costa Bianchi, and A. H. R. Costa, "Decaf: deep case-based policy inference for knowledge transfer in reinforcement learning," *Expert Systems with Applications*, vol. 156, p. 113420, 2020.

[31] Z. Benbahria, I. Sebari, H. Hajji, and M. F. Smiej, "Intelligent mapping of irrigated areas from landsat 8 images using transfer learning," *International Journal of Engineering and Geosciences*, vol. 6, no. 1, pp. 40–50, 2021.

Dr Pradeep Kumar Roy

Dr Pradeep Kumar Roy received his M. Tech and PhD degrees in Computer Science and Engi- neering from the National Institute of Technology Patna in 2015 and 2018, respectively. He received a Certificate of Excellence for securing a top rank in the M. Tech course. He has been featured in the Top 2% of Scientists in the World list prepared by Stanford University, USA, in 2022. He is currently working as an Assistant Professor at the Department of Computer Science and Engineering, Indian Insti- tute of Information Technology (IIIT) Surat, Gujarat, India. He also worked at Vellore Institute of Technology, Vellore, Tamil Nadu, India. His specialization straddles question answering, text mining and information retrieval, social network, and wireless sensor networks. He has published articles in different journals, including IEEE Transactions on Artificial Intelligence, IEEE Transactions on Network Science and Engineering, Neural Processing Letters, Computers and Electrical Engineering, IJIM, Neural Computing and Applications, Future Generation Computer Systems, Journal of Information Sciences, and others. He has also published conference articles at various international conferences, including ACL, FIRE, IEEE, Springer, and others.

Dr Ashish Singh

Dr Ashish Singh is currently working as an As- sistant Professor, School of Computer Engineering, Kalinga Institute of Industrial Technology, Deemed to be University, Bhubaneswar, Odisha-751024. He completed his BE and M.Tech in Computer Science and Engineering in 2013 and 2015, respectively. The Ph. D. degree has been received in Computer Science & Engineering from the National Institute of Technology Patna (Bihar), under Visvesvaraya Ph.D. Scheme for Electronics & IT Ministry of Electronics & Information Technology (MeitY) Government of India in 2020. His research areas are cloud security, trust management, healthcare security, Internet of Things, access control, Edge computing, and network security. He has published articles in different Journals, including Journal of Network and Computer Applications, ICT Express, Journal of Ambient Intelligence and Humanized Computing, Multimedia Tools and Applications, Computers and Electrical Engineering, Cluster Computing, IEEE Transactions on Industrial Informatics, IEEE Transactions on Network Science and Engineering, and others. He has also published many conference proceedings at prestigious international conferences. In a special section, he manages the editorial ship as a lead guest editor in Computers and Electrical Engineering and IEEE Transactions on Industrial Informatics Journal.

Human Activity Recognition From Sensorised Patient´s Data in Healthcare: A Streaming Deep Learning-Based Approach

Sandro Hurtado[1]*, José García-Nieto[1], Anton Popov[2], Ismael Navas-Delgado[1]

[1] Institute for Software Technologies and Software Engineering (ITIS), Biomedical Research Institute of Málaga (IBIMA), Department of Computer Languages and Computing Sciences, University of Málaga ETSI Informática, Campus de Teatinos, Málaga, 29071 (Spain)
[2] Department of Electronic Engineering, Igor Sikorsky Kyiv Polytechnic Institute, Kyiv (Ukraine)

Received 4 June 2021 | Accepted 25 March 2022 | Early Access 6 May 2022

Abstract

Physical inactivity is one of the main risk factors for mortality, and its relationship with the main chronic diseases has experienced intensive medical research. A well-known method for assessing people's activity is the use of accelerometers implanted in wearables and mobile phones. However, a series of main critical issues arise in the healthcare context related to the limited amount of available labelled data to build a classification model. Moreover, the discrimination ability of activities is often challenging to capture since the variety of movement patterns in a particular group of patients (e.g. obesity or geriatric patients) is limited over time. Consequently, the proposed work presents a novel approach for Human Activity Recognition (HAR) in healthcare to avoid this problem. This proposal is based on semi-supervised classification with Encoder-Decoder Convolutional Neural Networks (CNNs) using a combination strategy of public labelled and private unlabelled raw sensor data. In this sense, the model will be able to take advantage of the large amount of unlabelled data available by extracting relevant characteristics in these data, which will increase the knowledge in the innermost layers. Hence, the trained model can generalize well when used in real-world use cases. Additionally, real-time patient monitoring is provided by Apache Spark streaming processing with sliding windows. For testing purposes, a real-world case study is conducted with a group of overweight patients in the healthcare system of Andalusia (Spain), classifying close to 30 TBs of accelerometer sensor-based data. The proposed HAR streaming deep-learning approach properly classifies movement patterns in real-time conditions, crucial for long-term daily patient monitoring.

Keywords

Deep Learning, Healthcare, Human Activity Recognition, Semisupervised Learning, Spark Streaming Processing.

DOI: 10.9781/ijimai.2022.05.004

I. Introduction

Physical inactivity is one of the main risk factors for chronic diseases such as cardiovascular, cancer and diabetes [1], [2]. Knowing the habits and types of activity carried out by people and their relationship with these diseases is a key task to design treatment strategies and prevention recommendations. Numerous advances in Human Activity Recognition (HAR) has been crucial to deepen in high-level knowledge about people's daily life [3]. One of the main objectives of HAR is to provide long-term monitoring of people's daily activities to allow medical doctors to get additional information of their patients to design care plans that may prevent or help against chronic diseases.

HAR has gained much attention in healthcare due to its wide range of applications, such as monitoring of geriatric patients specially focused on fall detection [4]–[6], as well as many other studies related to chronic diseases such as Parkinson, obesity, cardiovascular and neurodegenerative diseases [7]–[10]. These research activities have shown that HAR can effectively improve the quality of health care for some groups of people suffering from some pathologies or chronic diseases.

HAR mainly focus on two types of methods: video-based and sensor-based. Video-based methods provide a dense feature space to allow fine-grained analysis in HAR. However, it is exposed with a high complex background of images, since an environment with very strict conditions, such as well-positioned cameras and individuals, is required for data collection process with a high cost at the level of computing resources, energy consumption and price. Therefore, video-based methods remain limited in epidemiological studies where the evaluation of daily physical activity requires a reliable, accurate, and low-cost methodology. Sensor-based methods are widely used in scientific physical activity studies since they provide better adaptability in variable environments, high recognition accuracy and low power consumption. Furthermore, in [3] the use of accelerometers is exposed as the most used sensor in the literature since most wearable devices are equipped with them and have easy access. Additionally, the use of accelerometer is considered a reasonably competent sensor for

* Corresponding author.
E-mail address: sandrohr@uma.es

recognising of many types of activities since most of them are simple body movements. This work is motivated by an ongoing collaboration project in the context of a real-world healthcare system (in Andalusia, Spain). We focus on a sensor-based approach, with the main propose of discriminating basic posture change movements or activities of a group of patients with obesity and cardiovascular problems. The goal of the project is to provide tools to practitioners to follow the daily routine of their patients and thus prevent sedentary lifestyle. In this sense, many related studies in the literature have reported high classification accuracy [11]–[14]. However, most of them have been tested in academic datasets on young, healthy subjects, that can hardly resemble the conditions of a real patient's environment. Besides, most of these experiments have been carried out under controlled environments, where activity conditions are restricted.

However, as observed in actual healthcare scenarios, a series of critical issues arise related to the limited amount of available labelled data to build a classification model regarding to the total volume and velocity of sensorised data. In addition, the discrimination ability of features is often difficult to capture for different classes, since the variety of movement patterns in certain group of patients, e.g. obesity and/or geriatric patients, is bounded and maintained over time. Another issue is the usual class imbalance of data registered in this kind of sensor data streams. Due to samples representing specific constant postures, such as sleeping, sitting, active, inactive, etc., are perceptually abundant, compared other ones (running, up-stairs, etc.). Therefore, these challenges demand the design and development of hybrid data-driven approaches, where semi-supervised models can act at the core of data processing workflows, usually involving modern Big Data technologies.

In this study, a streaming classification model for Human Activity Recognition in healthcare systems, is proposed for patient monitoring in real-time. This proposal is based on a combination strategy of public labelled/private unlabelled raw data integration, semi-supervised classification with Convolutional Neural Networks (CNNs) and Spark streaming processing.

Guided by practical requirements, accelerometer sensor-based data have been considered in this work since low power consumption and use of resources are mandatory through long-term daily patient monitoring in uncontrolled environments. In this sense, as sensorised samples are mostly unlabelled, a data fusion task is conducted with commonly used datasets in the literature (WISDM [15], PAMAP2 [16], HUGADB [17] and USC-HAD [18]). These datasets have been previously labelled according to systematic procedures and share common attributes. This way, labelled and unlabelled samples are integrated for feeding the semi-supervised models to classify new incoming flows of data, through Spark streaming processing engine, by following a sliding window strategy.

In this approach, semi-supervised models are generated with Encoder-Decoder CNNs [19], which allows data augmentation by considering unlabelled samples and statistical features, hence embracing the global properties of the accelerometer time series. For testing purposes, a real-world case study is conducted with a group of more than 300 overweight patients in the healthcare system of Andalusia (Spain), classifying close to 30 TBs of accelerometer sensor-based data.

The main contributions of this study are summarised as follows:

- A streaming semi-supervised HAR strategy is proposed for monitoring overweight patients in the context of a real-world healthcare system, involving a data fusion task of accelerometer sensorised data from labelled/unlabelled samples.
- Thorough experimentation is conducted for model selection and validation, where a semi-supervised CNN-Encoder-Decoder is evaluated with varying amounts of unlabelled data.
- The resulting analysis workflow is deployed on a cluster of Spark nodes, so the continuous classification of 30 TBs sensor data is predicted for a group of patients. The proposed HAR streaming deep-learning approach properly classifies movement patterns in real-time conditions, which is crucial for long-term daily patient monitoring.

The remainder of this paper is structured as follows. Section II presents a review of related studies in the current state of the art. In Section III, methodology and approach are described. The experimental procedure is explained and results are analysed in Section IV. Finally, Section V contains concluding remarks and future work.

II. Related Work

The discovery of patterns of human activity has led to several studies on how to analyze the data collected through activity bracelets, smartwatches and smartphones [20]. Many classification methods have been used in previous studies, especially conventional approaches using Machine Learning algorithms [21] such as Extra Trees, AdaBoost, Random Forest (RF), Naive Bayes, k-nearest Neighbours (kNN) or Support Vector Machines (SVM). To name some representative studies of them, in [22] SVM was used to carry out the classification problem of HAR, collecting inertial sensor data through a smartphone mounted in the waist of the individuals. C4.5 Decision Tree and Naive Bayes classifiers were used to recognize 20 daily activities in [23]. In [24] kNN was declared the best classifier in comparison with C4.5 (J48) Decision Tree, Multilayer Perceptron Neural Network, Naive Bayes, logistic regression, and ensembles based on boosting and bagging. However, they still showed classification failures in similar activities.

Even when conventional approaches have obtained promising results with high-level classification accuracies in different controlled environments, these methods rely on feature-based classification guided by human domain knowledge, which supposes a heavily effort in the pre-processing data stage. Besides, the discrimination of very similar activities for these methods is still a difficult task. Deep Learning (DL) algorithms seem to be a good solution to overcome these problems since they conduct layer-by-layer structural modelling for specific feature extraction and allow the classification process after the segment pre-processing of raw data. One of the first approaches can be found in [25], where HAR classification is carried out with CNNs by extracting features without any domain-specific knowledge about raw-data. Also in [11], Convnets is proposed to perform efficient and effective HAR using smartphone sensors by exploiting the inherent characteristics of activities and 1D time-series signals, at the same time providing a way to automatically and adaptively extract robust features from raw data. Various state-of-the-art classification techniques under different scenarios are compared in [12], showing how deep neural networks perform with the best accuracy when the training data volume is drastically reduced.

Many other HAR studies have been implemented with deep learning methods, such as convolutional and recurrent approaches [9], [13], [14], [26]. In this sense, a thorough survey is reported in [3] where new challenges and trends are identified for this area. In concrete, two of these main challenges are related to the online/streaming processing or sensorised data, and the requirement of dealing with unlabelled data. These are, in fact, the direct consequence of working in real-world environments, requiring the management of high volumes of continuously sensorised data. Recent proposals [19], [27] are based on suitable semi-supervised frameworks to cope with these issues, although they are still limited when tackling with scalable data processing.

Moreover, in order to alleviate some of the drawbacks encountered in the literature, we have made an exhaustive study of general features

in the existing methods, as exposed in [3], [20], [28]–[31]. We have distinguished four main challenges pertaining to human activity recognition. These features are presented below:

- *Design issues*:
  1. *Cost*: Cost is a key factor for any technique. If accuracy of a solution is good but cost is too high, then it is of no practical use. Accelerometers are inexpensive, require relatively low power, and are embedded in most of today's cellular phones [32].
  2. *Obtrusiveness*: To be successful in practice, HAR systems should not require the user to wear many sensors nor interact too often with the application. There are systems which require the user to wear four or more accelerometers or carry a heavy rucksack with recording devices. These configurations may be uncomfortable, invasive, expensive, and hence not suitable for activity recognition.
  3. *Energy consumption*: extending the battery life is a desirable feature, especially for medical applications that are compelled to deliver critical information (Long term monitoring).
  4. *Sampling rate (frequency)*: low sampling frequencies tend to lose information in specific movements.
- *Data collection protocol drawbacks:*
  5. *Real world environments (No controlled environment)*: The procedure followed by the individuals while collecting data is critical in any HAR. In [33] demonstrated 95.6% of accuracy for ambulation activities in a controlled data collection experiment, but in a natural environment (i.e., outside of the laboratory), the accuracy dropped to 66%!
  6. *Large volume of data*: A comprehensive study should consider a large number of individuals.
  7. *Long term patient monitoring*: most studies do not offer patient monitoring over time, which is essential to improve the problem of HAR.
  8. *Data collection Flexibility*: people perform activities in a different manner which means that an acceptable number of subjects is needed for the study so that the trained model is flexible enough to work with other subjects.
- *Model selection drawbacks:*
  9. *Semi-supervised learning*: Typically, HAR systems rely on large amount of labelled training data. However, annotating data can be challenging in some situations, especially when the granularity of the activities is great or the user is unwilling to help with the gathering process. Using semi-supervised learning, these unlabelled data can still be used to train a recognition model.
  10. *Deep Learning*: Deep learning algorithms attempt to learn high-level features from data in an incremental manner. Nevertheless, in classical machine learning, domain experts must extract features from raw sensor data in order to make the patterns more visible for the learning algorithm.
- *Model evaluation drawbacks:*
  11. *Model generalisation*: People certainly perform activities in a different manner due to particular physical characteristics. We have proposed to evaluate activity recognition systems based on the subjects rather than of the segmented windows. This prevents over-fitting on the subjects and helps to achieve better generalisation results.
  12. *Latency*: Latency is a critical factor. If a solution is accurate but takes long time to provide the results, it is not practical.
  13. *Real time classification/real-time decision making*: This is important for human activity recognition because getting the results in real time is a compulsion in many situations.

Table I shows a comparison between our approach and a set of related works found in the literature of HAR in this section, according to the list criteria exposed above. As can be observed, desirable features related to real-world environments as real-time processing of the sensorised data, dealing with unlabelled data and managing of high volumes of continuously sensorised data are covered by our approach, which represent an advantage with regards to these compared works.

The proposed approach is conceived to cope with these limitations by combining semi-supervised Encoder-Decoder CNN dynamic models with Spark streaming processing in the context of real-world healthcare environments.

TABLE I. Comparison of Related Works Found in the Literature in Human Activity Recognition. The Comparison Has Been Made According to Four Main Challenges Encountered in the State of the Art Pertaining to Human Activity Recognition. Additionally, Our Streaming *Semi-Supervised Deep-Learning Approach* (*SSSDA*) Is Presented in This Table as *SSSDA*. It Is Worth to Note That Our Approach Represents an Advantage With Regards to these Compared Works in Terms of Real-Time Classification in Real-World Environments.

Features/HAR refs	[22]	[23]	[24]	[25]	[11]	[9]	[13]	[14]	[26]	[19]	[27]	*SSSDA*
1. *Cost*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2. *Obtrusiveness*	✓	×	✓	×	✓	✓	✓	✓	✓	×	✓	✓
3. *Energy*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4. *Sampling-rate*	×	×	-	✓	×	×	×	×	✓	✓	≈✓	✓
5. *Real-environment*	×	≈✓	×	✓	-	≈✓	×	✓	×	×	✓	✓
6. *Large data-volume*	×	×	×	✓	×	×	×	-	×	×	×	✓
7. *Long-monitoring*	×	×	×	≈✓	×	×	×	×	×	×	×	✓
8. *Data-flexibility*	×	×	×	✓	×	×	×	≈✓	×	-	×	✓
9. *Semi-supervised*	×	×	×	×	×	≈✓	×	×	×	✓	✓	✓
10. *Deep-Learning*	×	×	×	✓	✓	✓	×	✓	✓	✓	✓	✓
11. *Model-generalisation*	×	✓	×	×	✓	✓	×	×	×	×	×	✓
12. *Latency*	-	✓	-	-	×	≈✓	×	×	×	-	×	✓
13. *Real-time classify*	-	✓	×	≈×	×	≈✓	×	×	×	×	×	✓

III. Methodology and Approach

This section is devoted to present the data acquisition strategy and data pre-processing tasks conducted for data consolidation. The semi-supervised deep learning model used is also described. After this, the overall approach is detailed to illustrate how all the elements are integrated.

A. Data Acquisition Strategy

In this work, we have followed a combined strategy for data acquisition that consists in merging private patient's sensor data and academic datasets. The former source comprises data streams of unlabelled attributes (patients' movements) that have to be classified. The latter one considers a series of labelled datasets from related studies of human activity recognition time-series in the literature. The main purpose of this strategy is to generate an enriched dataset that, after a feature engineering process for data fusion, is suitable for feeding semi-supervised models, avoiding bias and overfitting problems as much as possible.

Sensor data are generated using GENEActiv[1] wearable devices, which incorporate a MEMS triaxial accelerometer placed on the non-dominant wrist of the study subjects.

Each measurement of this bracelet contains three real values on each of the sensor axes ('x-y-z') with a sampling rate at 100Hz, range of +/- 8g and resolution of 12 bits. In this way, after a weekly observation period, a total amount of 30 TBs of raw movement data was collected from 300 patients' daily activities. This final time series dataset is a set of observations $X = (x_t^1, x_t^2 \ldots x_t^L)$ where each one is recorded at a specific time T and L as a length of time-step.

Nevertheless, as commented before, sensorised data still lacks class labelled features, which are required for model training. Therefore, a series of widely used datasets in the literature have been considered in the proposed approach, each one of them contributing with labelled samples for different, sometimes overlapping, activities. These datasets are: WISDM (Actitracker) [34], PAMAP2 [35], USC-HAD [36] and HuGaDB [37]. These datasets were previously labelled according to systematic procedures and sharing common attributes. The time-series recorded in these datasets have been collected from heterogeneous devices (smartphones and bracelets) located in different parts of the body, considering a different number of individuals and with a different sampling frequency (e.g. WISDM at 20Hz, HUGADB at 50Hz, USC-HAD and PAMAP2 at 100Hz) in the study. Moreover, they have been modelled to consider different sets of daily activities, which are recorded through different time intervals.

Therefore, a thorough pre-processing phase has been carried out to homogenise all these data sources, including those commonly detected activities among all the individuals in observation. In concrete, these shared activities are: running, walking, sitting, standing, up stairs and down stairs, which are used as labelled categories for the semi-supervised models worked in this proposal.

B. Data Pre-Processing

Besides, raw data have been normalised through Z-score Normalisation. Feature standardisation makes the values of each feature in the raw data have zero-mean and unit-variance. This normalisation is formulated in (1), where x is the original feature vector, x' is the normalised value, $\tilde{x} = average(x)$ is the mean of that feature vector, and σ is its standard deviation.

$$x' = \frac{x - \tilde{x}}{\sigma} \quad (1)$$

Also, linear interpolation have been conducted to tackle with missing values and to fill gaps in raw data time series. This method searches for a straight line that passes through the end points x_A and x_B, as formulated in (2), where xi are observed data, X_i are the interpolated value(s) of missing data and α is the interpolation factor that varies from 0 to 1.

$$X_i = (1 - \alpha)x_B + \alpha x_A \quad (2)$$

However, the most relevant task in this regard has been re-sampling data. In particular, down-sampling and up-sampling are performed on data, since when dealing with "waves" in time-series, it is observed that low sampling frequencies tend to lose information in specific movements, where a high frequency is required to identify them correctly. For this reason, we must determine the wave frequency according to the type of recognition faced. Fig. 1 shows an example where raw data of a patient's activities ("walking" and "cycling") are collected by an accelerometer sensor on a wrist. After re-sampling, data are transformed for each activity at frequencies of 100Hz (top), 50Hz (middle) and 20Hz (bottom). The effect of this re-sampling is illustrated and it is possible to identify some losses in the data information as long as the frequency is decreasing. It can be observed in Fig. 1 a), where different waves peaks "disappear" provoking inconsistent data representations at different sampling frequencies. Therefore, a high re-sampling (100Hz) is performed to keep informative level in samples, while making data homogeneous for all the sources.

Another quite common, yet important, issue registered in HAR datasets is the class imbalance. Even more in real-world sensor data from the particular case of obesity patients, where the balance between classes is not guaranteed and biased to sedentary activities. For example, the "sitting" activity is more frequent in the case of overweight patients than the "running" activity, producing an important class imbalance that could lead learning models to behave with a bias towards majority classes. As a consequence, algorithms will fail in the classification of the underrepresented minority classes, which provokes a severe decreasing in the overall accuracy of the results [38].

In order to cope with class imbalance, several approaches have been used such as oversampling and under-sampling methods at the data level and many other solutions at the algorithmic level trying to trade-off the class imbalance in modelling time . In the context of HAR, Synthetic Minority Oversampling Technique (SMOTE) is a common over-sampling method used to generate new synthetic data of the minority classes. It has shown a great deal of success in several applications where SMOTE helps to enhance the classification accuracy for imbalanced datasets. For example, in data balancing was used through SMOTE oversampling approach, leading the worked model to reach high accuracy results.

By default, SMOTE re-samples all classes excepting the majority class, that is, the minority classes are increased to reach the total number of the majority class. However, the study in [39] suggested combining SMOTE with random under-sampling of the majority class, since a high over-sampling could provoke model over-fitting. For this propose, our methodology addresses class imbalance at training stage by balancing classes in two separate steps: firstly, SMOTE oversampling technique is used to over-sample those minority classes to have 50% of the number of examples of the majority class. Then, under-sampling using random elimination is performed on the majority classes, to have 20% more than the minority class. Then a difference of 20% between classes of samples is obtained, which helps the model to avoid problematic class imbalance, preventing the generation of synthetic data in a high percentage.

[1] https://www.activinsights.com/products/geneactiv/

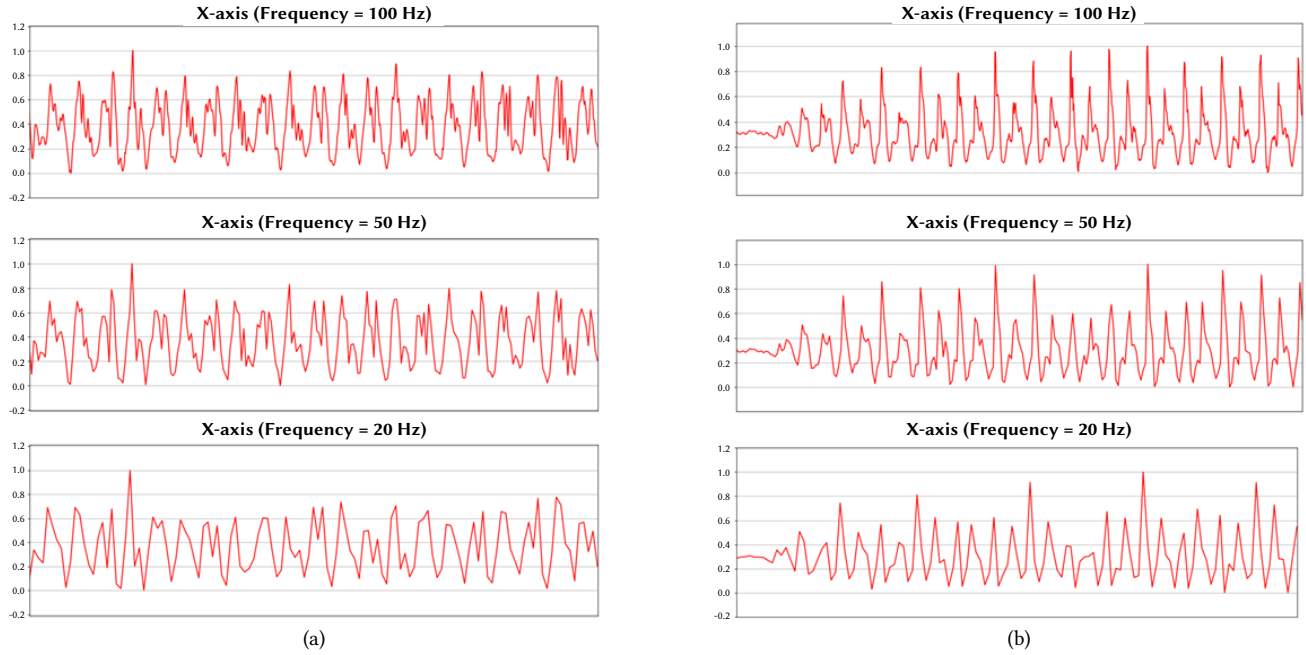


Fig. 1. Raw data from accelerometer sensor of different activities: Walking (a) and cycling (b) at 100 Hz (top) and resampled data at 50 Hz (middle) and 20 Hz (bottom). It can be noticed that as the sampling rate decreases, aspects at high frequency are removed from the wave.

C. Semi-Supervised Modelling

One of the main challenges arising in this study is the possibility of taking advantage of dealing with labelled and unlabelled data. In this sense, the use of semi-supervised learning techniques constitutes a suitable option to perform predictive analysis, since they allow to train models with, labelled and unlabelled samples, which mainly improve generalisation and avoid over fitting [19].

In particular, the use of CNN based approaches has been shown to perform successfully for HAR, since they provide hidden representations of data and to identify patterns in activity time-series [25], [27]. Therefore, considering a dataset with N pairs $(x_1, t_1), (x_2, t_2), ..., (x_N, t_N)$, being x_i a sliding window input with length T and t_i the label representing a given activity, we adopt a similar semi-supervised strategy to a CNN Encoder-Decoder [27] in our approach. In this, labelled samples $\{(x_i, t_i) | \ 1 \le i \le N\}$ are used together with unlabelled ones $\{x_i \,|\, N+1 \le i \le N+M\}$ in training, to fit the model with both data sources (sensorised and academic).

In general, the encoder network maps a given input signal $x \in X \subset \mathbb{R}^{d_0}$ to a feature space $z \in Z \subset \mathbb{R}^{d_k}$, whereas the decoder takes this feature map as an input, process it and produce an output $y \in Y \subset \mathbb{R}^{d_L}$

The rationale behind the CNN Encoder-Decoder for semi-supervised classification is to include noise into all the layers of the network, so it works to minimise the distance between the clean input and the reconstructed decoder one. In this way, the learning procedure can be summarised in the following steps:

1. Labelled and unlabelled data are processed by the clean encoder to compute hidden variables in the middle layers z_i^k;
2. Both labelled and unlabelled data are corrupted with Gaussian noise and transformed to an abstract representation $\tilde{z}_i^k$, by the noisy encoder;
3. Labelled data $(\tilde{x}_i, 1 \le i \le N)$ are used to perform the prediction task on a softmax based on cross entropy cost. The predicted classes are denoted with $\tilde{y}_i$;
4. The decoder works to reconstruct unlabelled samples $(\tilde{x}_i, N+1 \le i \le N+M)$ which are denoted with $\hat{x}_i$, so they should be as close as possible to the corresponding input (x_i). To measure this similarity, square error is computed.

The cost function is formulated in (3) as an aggregation of the supervised cross entropy of the noisy output $\tilde{y}_i$ predicting the class activity t_i for the input x_i (first term in this equation), whereas the unsupervised cost (second term in this equation) is the denoising square error between clean input x_i and their noisy reconstruction output $\hat{x}_i$.

$$Cost = -\frac{1}{N}\sum_{i=1}^{N} \log P(\tilde{y}_i = t_i | x_i) + \frac{\lambda}{N}\sum_{i=N+1}^{N+M} || \hat{x}_i - x_i ||_2^2 \tag{3}$$

Therefore, the semi-supervised CNN Encoder-decoder allows unlabelled samples from sensor streaming sources to take part in the learning model in training time, so it will avoid bias to certain classes and promote generality.

D. Overall Approach

A general overview of the proposed approach is illustrated in Fig. 2, where all the elements are organised, from data acquisition to model evaluation and human activity prediction. It partially follows the so-called activity recognition chain (ARC), extensively studied in [44] as a general-purpose framework for processing time-series sensorised data, training and evaluating HAR workflows. These steps are thoroughly described next:

1. *Data acquisition.* As commented before, we have followed a combined strategy of self data collection from sensors together with public datasets, with the aim of feeding a semi-supervised model with unlabelled and labelled samples, respectively. Nevertheless, public datasets have been generated with different devices and human conditions, sometimes far from the habits observed in our patients (with obesity), so a preliminary exploration phase has been conducted to select that public dataset containing distributions more similar to our self-collected (private) data. In this regard, Fig. 3 shows the boxplot distributions of the three accelerometer axis (x,y,z) for each of the four considered public datasets (WISDM, PAMAP2, USC-HAD and HuGaDB), taking into account the 6 activities which have in common these datasets (walking, running, sitting, standing, downstairs, upstairs), as well as for our private data. After this process, the WISDM dataset is selected to provide our model with labelled samples, since it contains in overall the

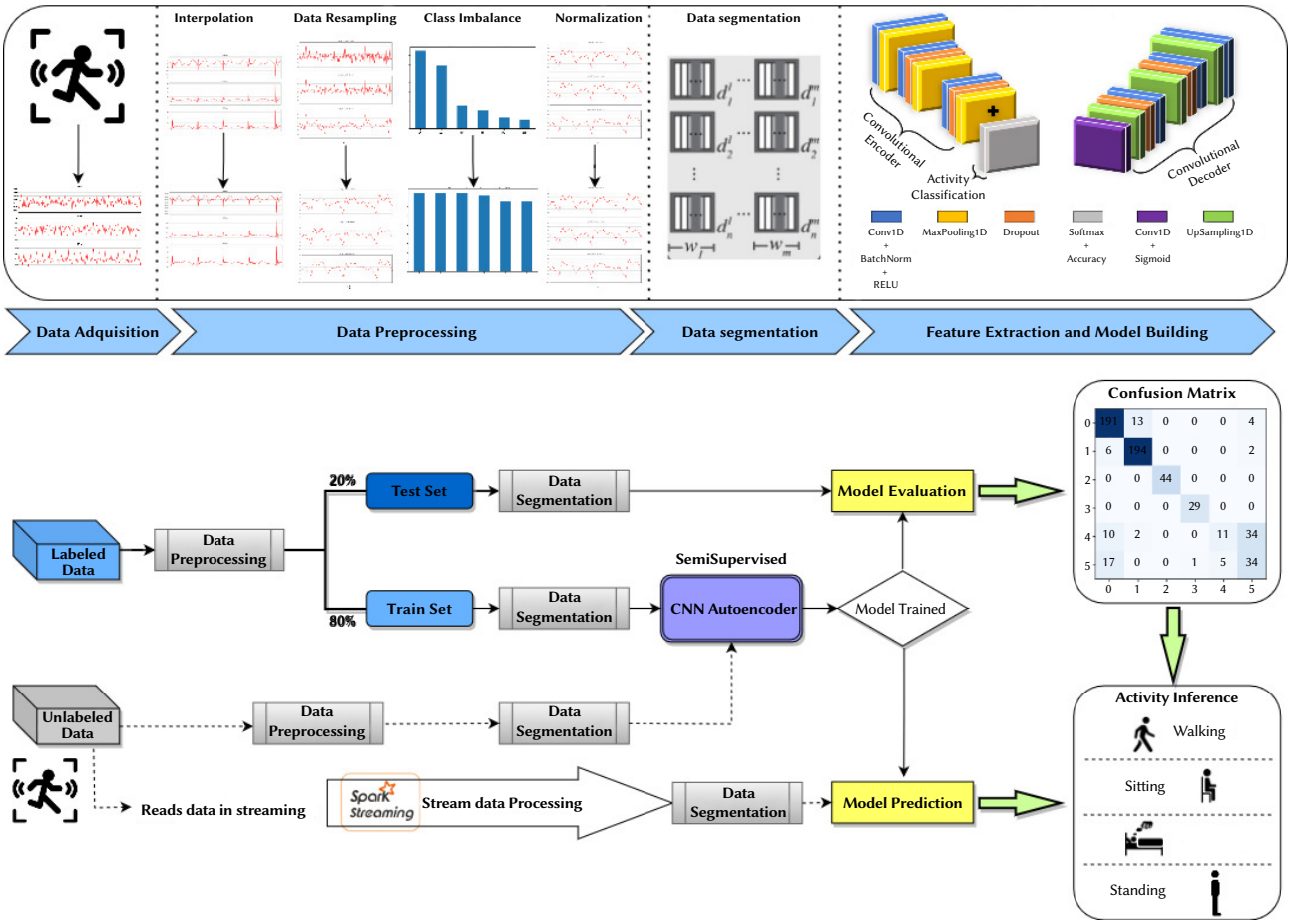


Fig. 2. General overview of the proposed approach that is presented as a HAR workflow. This workflow is composed of several steps: (1) **Data acquisition**: the data is acquire combining unlabelled data sensors (private dataset) and from public datasets. (2) **Data pre-processing**: these data is pre-process, which involves interpolation for missing data imputation, re- sampling, class imbalance processing and normalisation. Also labelled dataset is then split into two subsets with 80% of selected samples for training and 20% of remaining ones for testing. (3) **Data segmentation**: a temporal sliding window with size of 400, corresponding to roughly 4 seconds of physical activity data, and overlap of 100 (1 second) is performed to labelled and unlabelled data. (4) **Feature extraction and model training**: a CNN Encoder-Decoder model is trained with labelled and unlabelled, capturing the most relevant characteristics of the training data in order to provide activity inference of the 30TB of unlabelled data. (5) **Model evaluation**: the model is evaluated with the test sets where confusion matrix and deviated metrics are obtained (Precision, Recall, F1-score) (6) **Streaming processing and activity recognition**: once the model is evaluated and provide us promising results an Spark Streaming classification process is carried out.

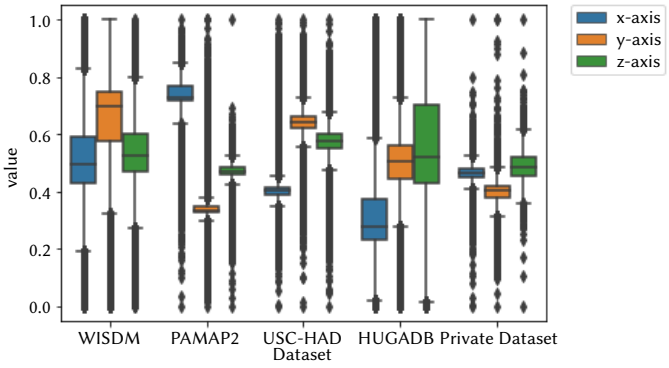


Fig. 3. Boxplot distributions of the three accelerometer axis corresponding to WISDM, PAMAP2, USC-HAD and HUGADB, taking into account the 6 activities which have in common these datasets (walking, running, sitting, standing, downstairs, upstairs). Also our private dataset was included in the bloxplot distribution.

closest axis distributions to the sensorised data of our patients. Therefore, we avoid the model to underfit with excessive data variation. When the instances are augmented using the WISDM dataset the model became more stable with smaller standard deviation. On the contrary, using all datasets together to train the model add additional variation and it deteriorates the model too much. In concrete, WISDM (Actitracker) dataset considers 6 activities registered in a controlled environment: jogging, walking, ascending stairs, descending stairs, sitting and standing. A number of 36 individuals have taken part in these measures.

2. *Data pre-processing*. A second step of data processing is performed (as explained before) on labelled and unlabelled data, which involves interpolation for missing data imputation, re-sampling, class imbalance processing and normalisation. It is worth to note, we re-sampling WISDM dataset from 20Hz to 100Hz (same frequency of our private dataset) in order to keep data information as commented before in Fig. 1. The labelled dataset is then split into two subsets with 80% of selected samples for training and 20% of remaining ones for testing.
3. *Segmentation*. At this step, data samples are still structured in the time domain, since all the axis points are collected at a certain time instant from sensors. Therefore, a segmentation stage is required to transform these input data into the frequency domain, more suitable for training deep learning models as signal processing prediction tasks. To do so, for each axis attribute in the dataset, a temporal sliding window with size of 400, corresponding to

roughly 4 seconds of physical activity data, and overlap of 100 (1 second), is performed. This overlapping among windows guarantees high numerosity of training and testing samples to train the model. To match the input shape of the CNN-Encoder-Decoder, it is necessary to reshape the sample obtained in the previous step. Therefore, each window comes in the form of a matrix of values, of shape N x 400 x 3, where N is the number of samples resulting of the segmentation, 400 is the time window and 3 is the number of features to train the model (x-axis, y-axis, z-axis). In this segmentation, sliding windows are checked to contain samples from just one human activity.

4. *Feature extraction and model training.* This step entails the semi-supervised learning task, which merges the labelled segments in the training set with those unlabelled from sensors. The CNN Encoder-Decoder involves up-sampling for maxpooling decoding, as well as convolutional operation for deconvolution [27]. As argued in [27], using this semi-supervised CNN Encoder-Decoder, it is possible to learn the network and features simultaneously from the data.
5. *Model evaluation.* Once the model is built, an evaluation step is carried out with regards to the test set, where confusion matrix and deviated metrics are obtained (Precision, Recall, F1-score, etc). It is worth noting that this test set is completely obtained from the public dataset (in this case WISDM), although the model has been trained with both, public and private data, so final predictions are expected to show certain model generalisation with moderate accuracies. The final goal is to get a prediction model suitable for a very dynamic data flow environment, but not for a specific dataset in a certain time period.
6. *Streaming processing and activity recognition.* Finally, a streaming processing task is deployed through an Apache Spark environment, in which new sensorised data are pre-processed to be predicted according to the model previously built. An internal segmentation step is carried out with streaming data by using a similar sliding window size as used in model training phase. This is then a continuous process of human activity label assignation of new samples regarding patient's movements, which can be now monitored by practitioners.

The whole process is repeated with a certain frequency to rebuild models with updated data. Therefore, the framework to monitor patient's movements will consider new individuals in a transparent way to the learning model, since new sensor data will be in the same Spark streaming source.

IV. Experimental Results and Analysis

In this section, we investigate the effects of training a semi-supervised CNN Encoder-Decoder using labelled data from one public dataset (WISDM) and unlabelled data from our private dataset.

The goal is to be able to classify the 30 TB of unlabelled data. The Convolutional Encoder will compress the input signal x into a space of latent variables ($h = f(x)$) , then learning how to reconstruct the data back from the reduced encoded representation. Meanwhile, the Convolutional Decoder works to reconstruct the input signal based on the information previously collected ($r = g(h)$), as observed in Fig. 4. Therefore, the latent variable space h will capture the most relevant characteristics of the training data.

In this regard, the algorithm learns how to reconstruct the input by using the Adam optimiser [45] and using the mean square error as a loss function. Therefore, the model will be able to extract more significant characteristics from the unlabelled data that will help us to make predictions.

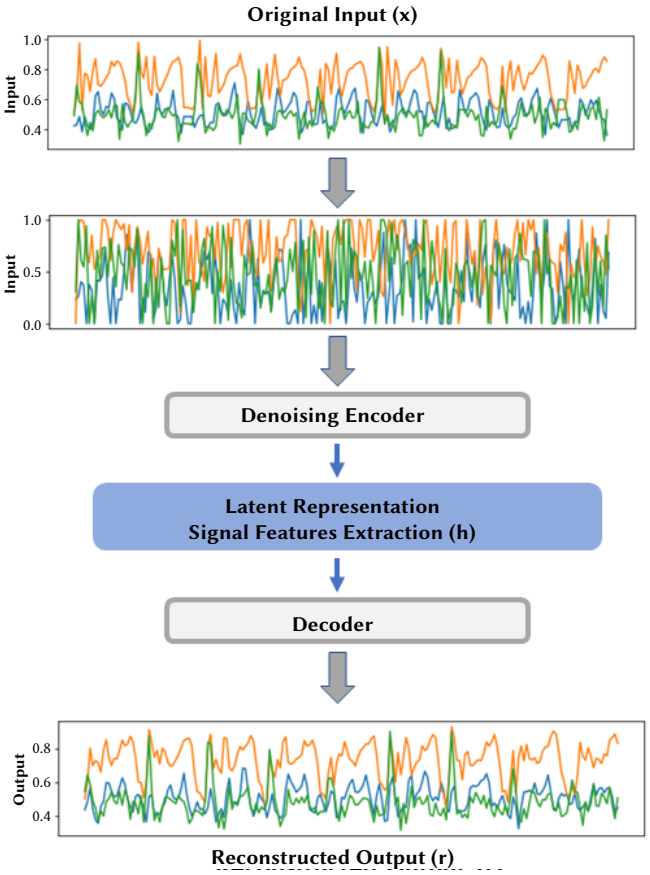


Fig. 4. General structure of the CNN Encoder-Decoder, contains a clean convolutional Encoder, noisy convolutional encoder, and a convolutional decoder. Labelled and unlabelled data are processed by clean convolutional encoder and then corrupted with Gaussian noise. Then the convolutional decoder works to reconstruct the clean inputpxq from high-level representation $r = g(h)$.

A. Model Selection

The full structure of our CNN-Encoder-Decoder model is shown in Fig. 2.

Encoder: The encoder network consists of three down-sampling blocks. Each down sampling block is composed of 1D convolutional layers with kernel size of 3, followed by a max-pooling layer. Additionally, for each block a batch normalisation is added to reduce internal co-variate shift [46], accelerating the training process of the model, and a dropout layer was added to improve generalisation performance and avoid over fitting. It then follows an structure [Conv1D + BatchNorm + MaxPooling1D + Dropout]

Decoder: Each encoder layer has a corresponding decoder layer. Thus, the decoder network consists of three up-sampling blocks composed of 1D convolutional layers with a kernel size of 3, followed by an up-sampling layer. As for the encoder, for each up-sampling block, batch normalisation and dropout layers were added, with a structure [Conv1D + BatchNorm + UpSampling1D + Dropout].

Bayesian optimisation has been used for efficient hyper-parameter tuning [47]. The hyper-parameters were tuned by performing 10-fold Stratified Shuffle Split cross-validation on the training set using Bayesian optimisation, obtaining a filter size of 64 for each of the 1D convolutional layers, which is activated by the Restricted Linear Unit (ReLU) function. Moreover, each of the max-pooling and up-sampling layers contains a pooling size of 2 and the dropout was set to 0.1 for each one. The Bayesian optimisation was executed with a batch size of 50, 500 and 1000, obtaining the best results with 50.

In order to assess the performance of our classification methodology system, we split the available dataset into 80% train data and 20% test data. This was done based on the subjects rather than of the segmented windows. In this regard, train data contain from subjects 1 to 32 of WISDM dataset and test data include the rest of the subjects (32 to 36). Thus, for each experiment four subjects out of 36 are always kept isolated to evaluate the model. This prevents over-fitting on the subjects and helps to achieve better generalisation results.

To comprehensively evaluate the model, we used several evaluation metrics to evaluate the classification results: accuracy, precision, recall, F1-score, loss function, receiver operating characteristic (ROC) and normalised discounted cumulative gain (NDCG), as shown in Table II. It should be noted, we opted to estimate the mean F1-score (Fm-score), that is the mean F1-score across all the classes. It's shown in (4) and (5), where TP is the number of true positives in prediction, FP are the false positives and FN are the number of false negatives.

$$precision = \frac{TP}{TP + FP} \quad recall = \frac{TP}{TP + FN} \tag{4}$$

$$Fm - score = \frac{2 * precision * recall}{precision + recall} \tag{5}$$

The CNN Encoder-Decoder has been implemented in TensorFlow using Keras. The experiments to evaluate the model have been executed on a machine with 16 CPUs (Intel(R) Xeon ® Gold 6130 CPU 2.10GHz). After each epoch of training, we evaluate the performance of the model on the validation set. Each model is trained for at least 50 epochs. Training stop condition is configured if there is no increase in validation performance for 10 subsequent epochs. We select the epoch that showed the best validation-set performance and apply the corresponding model to the test set.

B. Sensitivity to Unlabelled Sample Size

In this section, we study the performance of our semi-supervised CNN Encoder-Decoder model trained with varying amounts of unlabelled data. The amount of the unlabelled data will be proportional to a percentage of samples of the labelled data used for training. Therefore, we evaluate the metrics of our model trained using unlabelled data of 10%, 20%, 30%, 50%, 80%, 100%, 150% proportion of labelled data used for training, as shown in Table II. The number of unlabelled samples varies from 97,814 (10% of train labelled data) to 1,467,222 (150% of train labelled data).

Fig. 5 shows how the Fm-score evolves when varying the number of unlabelled examples in the experimental results. Fm-score generally decreases when there are more unlabelled samples as expected. This is explained by the fact that unlabelled data comes from a different dataset then including variation. However, it can be observed in Fig. 5 that for percentages of unlabelled data less than 100%, we obtain a high Fm-score in the result.

TABLE II. Metrics Obtained With Varying Number of Unlabelled Examples in Training Set. The Amount of Unlabelled Data Is Taken As a percentage of the Training Set of the Labelled Data (WISDM Dataset). The Number of Unlabelled Samples Varies From 97,814 (10% of Train Data) to 1,467,222 (150% of Train Data)

Metrics: Public data (labelled) + Private data (Unlabelled)						
%	acc	loss	recall	Fm-score	roc	ndcg
0	0.981	0.069	0.981	0.981	0.998	0.998
10	0.976	0.075	0.977	0.967	0.995	0.997
20	0.971	0.076	0.949	0.949	0.992	0.993
30	0.951	0.148	0.940	0.938	0.991	0.990
50	0.947	0.151	0.925	0.926	0.990	0.988
80	0.905	0.292	0.905	0.903	0.987	0.985
100	0.875	0.319	0.872	0.871	0.983	0.984
150	0.685	0.601	0.685	0.655	0.941	0.981

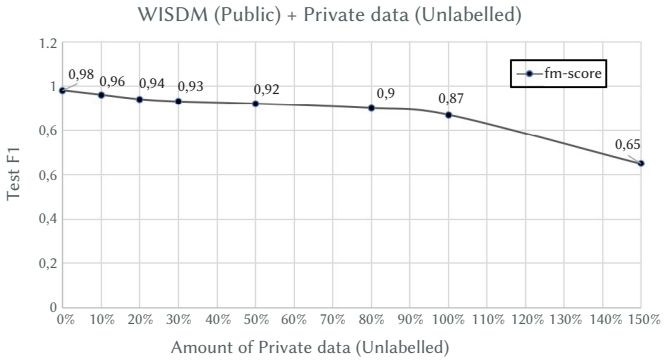
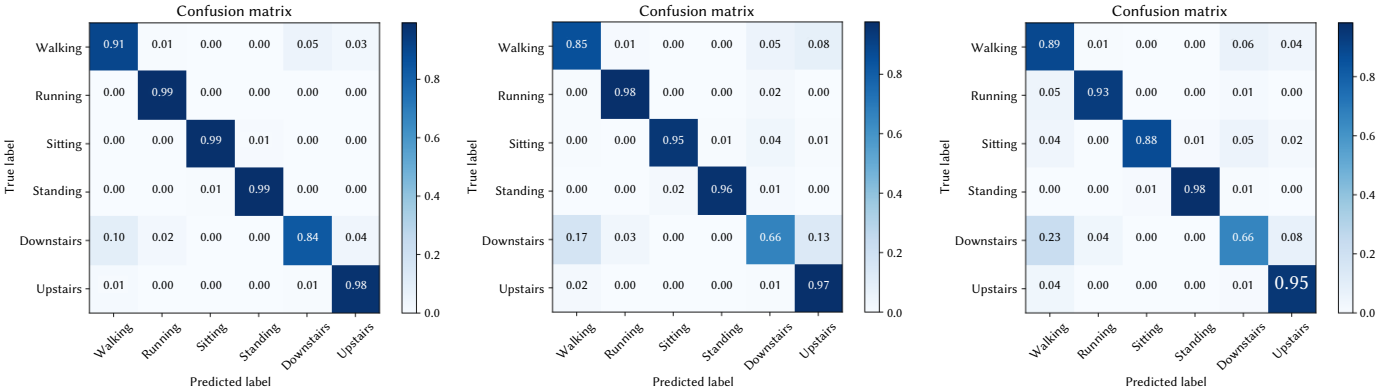


Fig. 5. Fm-scores obtained with varying number of unlabelled examples in training set.

Thus, our approach can potentially learn the network and features simultaneously from the data using unlabelled data in our CNN Encoder-Decoder model. Therefore, it is possible to use this model as core predictor. To do so, we have chosen the amount of 80% of unlabelled data to classify the 30 TB from sensors, since at this point, the model is still getting good results (Fm-score = 0.90).

More in depth, Fig. 6 shows the resulting confusion matrices when varying the amount of unlabelled data with 10%, 50% and 80% in the model training. It can be observed that the model achieves promising



(a) 10% of unlabelled data (b) 50% of unlabelled data (c) 80% of unlabelled data

Fig. 6. Illustration of confusion matrices showing the sensitivity of the networks for each individual class when varying 10%, 50% and 80% of unlabelled data when training the semi-supervised CNN-Encoder-Decoder.

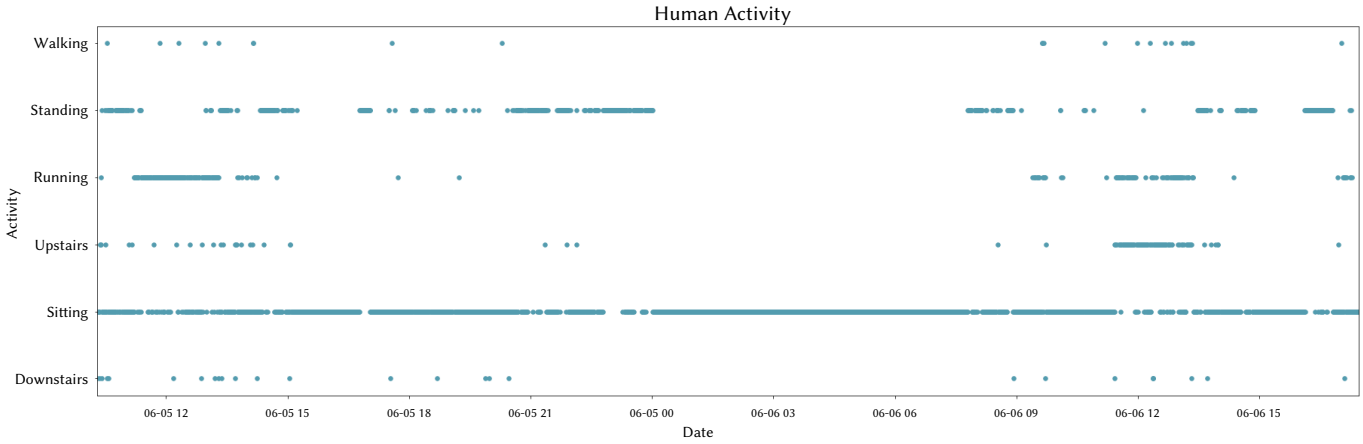


Fig. 7. Snapshot of the Human Activity Recognition for a randomly anonymous patient. It is shown how during the night sitting (resting) is the main activity, later around 8:30, the patient starts to be more active and does short movements. Then, at 12:00 the patient seems to start some moderate activity and finally, after 00:00 resting is the main activity.

predictions for activities walking, running, sitting, standing and upstairs even when increasing the number of unlabelled samples. In contrast, the model start to show limited predictions in detecting downstairs, since, if we see the patterns between walking and downstairs, they are characterised with very close signal shapes in movements, as mentioned in [15]. This is general an acceptable precision, since even for 80% unlabelled data it still gets good predictions for all classes.

As we know, it is hard to assess performance in unlabelled data, but we still need to know if it passes *"the eye test"*. For this propose, we classify a randomly chosen sample of unlabelled data in order to demonstrate that the distributions of the predictions are reasonable. It is shown in Fig. 7 (format date is month-day hour) how the main activity is resting (sitting and standing) as we expected. It's normal since this unlabelled data correspond to one of the 300 overweight patients in the healthcare system of Andalusia. In the same way, during the night (from 00:00 to 08:30 approximately) the patient is totally resting (sitting) . Later, the patient is standing and starts to be more active. Then around 12:00, the patient seems to starts to do moderate physical activity (running and upstairs). It can be seen that on both days at 12:00 (06-05 12:00 and 06-06 12:00) the patient carries out physical activity. This could be explained by the fact that patients follow the doctors' instructions doing daily exercise to avoid sedentary life. Afterwards, the patient does some short movements and finally, after 00:00 resting is the main activity.

It should be note, the classification has been carried out according to the labels that we have from the WISDM dataset, however our private dataset provide us a long-term monitoring of patient's daily activities where we can find more activities and transitions between activities. Even so, the results obtained in Fig. 7 seem quite reasonable to us for this first approach in which we try to address the problem of HAR in a real world case without previously labelled activities in our dataset.

C. Additional Experiments

Additional experiments have been implemented to demonstrate the feasibility of the proposed semi-supervised methodology. A first experiment was carried out to see whether the model was able to pass *"the eye test"* without taking into account the semi-supervised approach. In consequence, the model was trained only with raw data from WISDM dataset. After that, a classification task was performed from a randomly chosen sample from our 30TB private unlabelled dataset. As expected, the model didn't pass *"the eye test"* without using unlabelled private data in the training phase (Fig. 11).

Moreover, the proposed methodology has been synthetically evaluated by using another public dataset as a simulation of the unsupervised portion. In this sense, HUGADB dataset has been considered as *"unlabelled dataset"* and WISDM as labelled dataset. HUGADB dataset was classified with and without considering our proposed semi-supervised methodology. Finally, the model was evaluated if it can predict the activities in HUGADB dataset. In this experiment, we concluded that using the semi-supervised approach give us better predictions as observed in Table IV in Appendix. The same experiment was carried out with PAMAP2 as *"unlabelled dataset"*. See Appendix for more details in the experiments.

D. Computational Performance

To carry out the streaming classification process, a deployment of the complete approach has been conducted on a virtualisation environment operating on an on-premise high-performance cluster computing platform. This infrastructure is located at the Ada Byron Research Center of the University of Malaga (Spain). It comprises several units of virtualisation that allows to visualise the performance of the cluster. Concretely, this platform has 10 virtual machines, each one with 16 cores (CPU 16 x 2.10 GHz), 128 GB RAM and 1 TB of virtual storage (adding up to 176 cores, 1408 GBs of memory and 10 TB HD storage). These virtual machines have been used with the role of Worker node (Apache Spark) to make the activity predictions. The Master node, which runs the Keras CNN Encoder-Decoder, is hosted in a different machine with 16 cores at 2.10 GHz, 128 GB RAM and 5,000 TB of virtual storage space. All these nodes use Linux 4.15.0-118-generic 64-bit distribution. The whole cluster uses Spark 3.0.1.

Additionally, an NFS distributed file system has been configured to be able to access the sensorised data from all the machines. The Master node will physically store the data (server), while the Worker nodes will behave as clients to access the data remotely. In this way, it is possible to perform the activity prediction in parallel from the different machines connected to the same network to access remote files as if they were local ones.

For the parallelisation of Spark streaming processes the classification of activities accessing a directory at the NFS distributed system. The data is passed in streaming from the repository. Each of the CSV files that are included in the directory will behave as a Spark streaming batch that will go through a segmentation process by time windows (400 rows corresponding to 4 seconds of monitoring activity) as observed in Fig. 2. Finally, the CNN Encoder-Decoder model trained will predict the activity of each batch in streaming. The results are saved in text files using the same name as the original CSV files (See Code Snippet 1).

TABLE III. Experimental Results Spark Streaming Computational Performance

	Running Time (seconds)				Speedup			Efficiency		
Batch Size	T_1	T_{40}	T_{80}	T_{160}	S_{40}	S_{80}	S_{160}	E_{40}	E_{80}	E_{160}
64 MB	28.10	6.29	7.15	7.08	4.46	3.93	3.96	11.16%	4.91%	2.47%
128 MB	69.17	4.71	4.03	4.22	14.68	17.16	16.39	36.71%	21.45%	10.24%
256 MB	124.65	5.74	10.44	10.94	21.72	11.92	11.39	54.29%	14.92%	7.12%
512 MB	244.28	5.85	34.34	34.05	41.76	7.11	7.17	104.39%	8.89%	4.48%
1 GB	462.75	8.18	124.56	115.21	56.57	3.72	4.02	141.48%	4.64%	2.51%

Code Snippet 1: Spark streaming segmentation and classification by batch

```
//Read csv in Streaming with Spark from directory
df = spark.readStream(directory)
//Load the CNN-Encoder-Decoder model
model = keras.load(model)

classify(batch, batch_id, model):
     // we set time window to 400 (4 seconds of activity)

     time_window = 400
     // raw data segmentation by time Window
     batch.map(lambda x,y: [raw_data],time_window)
     // group by time_window
     batch.reducebyKey(lambda x,y: x+y)
     // activity prediction of raw data
     batch.map(lambda r: model.predict(r))
     // save the result
     batch.saveAsTextFile(batch_id+ .txt)

// Streaming classification for each batch
df.foreachBatch(classify(batch, batch_id, model))
```

The performance of the proposed streaming solution has been evaluated through a series of experiments to measure the performance in terms of *Speedup (SN)* and the *Efficiency (EN)*. Thus we analyse the computational effort and the data management process. The standard formula of the *Speedup* calculates the ratio of *T1* over *TN*, where *T1* is the running time of the analysed algorithm in 1 processor and *TN* is the running time of the parallelised algorithm on *N* processing units (processors or cores), while the *Efficiency (EN)* is calculated as shown in (6).

$$SN = \frac{T1}{TN} \; EN = \frac{SN}{TN} * 100 \tag{6}$$

Table III shows the running time in seconds used by the Spark streaming classification approach running on 40, 80 and 160 cores with different batch sizes of raw data. This way, we have centred on file sizes of 64 MB, 128 MB, 256 MB, 512 MB and 1 GB, since they are the average size of CSV files that are in the 30 TB of data. In this sense, we measure the computational influence of using different number of cores with different batch size. This table also contains the corresponding Speedup and Efficiency values to the resulting times. As mentioned, the running time is reduced in relation to the increase in the number of cores used in the parallel model. The highest reduction in time is obtained when our approach is configured with 40 cores in parallel, for which the running time is reduced from 28.10 s to 6.29 s in the case of the smallest batch size (64 MB), and from 462.75 s to 8.18 s with the biggest batch size (1 GB) used in the experiments. Also, in terms of efficiency, the highest percentage, 141.48%, is reached with 40 cores with a batch size of 1 GB reaching the best efficiency. In contrast, it decreases as the number of resources gets larger. This behaviour was somewhat expected as the particular cluster configuration involves computing overheads due to virtualisation and network communications, so a trade-off setting is reached with less nodes, but stabilising from 80 nodes in advance. Considering the results, it is worth mentioning that both cluster configurations (80 and 160 cores) yield similar speedup and efficiency values, which indicates that the bottleneck is due to the parallel infrastructure, so increasing the number of cores do not compensate the synchronisation and communication costs.

Therefore, according to the results the best configuration to obtain the maximum performance in the streaming classification process with Spark, are observed when using the cluster resources with 40 cores and a batch size of 1 GB (Fig. 8). In this regard, we can consider our Spark streaming classification methodology as a real-time classification since we can classify 1 GB in 8.18s, that is approximately 12,000,000 of samples rows, what is equivalent to almost one week of daily patient activities monitoring (30 TBs in 2 days and 8 hours).

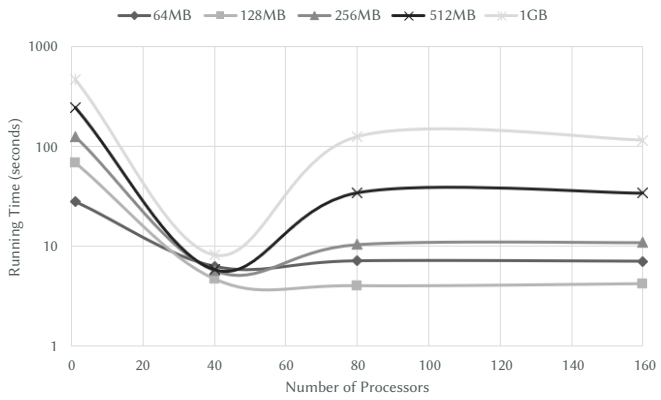


Fig. 8. Running time in seconds (logarithmic scale) of the Spark Streaming process classification executed on 40, 80 and 160 cores in the cluster computing platform.

In terms of computational effort, we have plotted the *Load one* measure of the entire cluster while running experiments with 40 and 160 cores with a batch size of 1 GB in Fig. 9 and Fig. 10 respectively, to check the overall CPU load. In particular, the *Load one* computes the number of threads at kernel level that are running and being queued while waiting for CPU resources, averaged over the last minute. We could interpret this number in relation with the number of hardware threads available on the machine and the time it takes to drain the run queue. Fig. 9 captures a short time (close to minute 8:00) in which the master node (Spark driver) delivers tasks to the worker nodes and they start to undertake data processing jobs when we run the experiment with 40 cores and 1 GB of batch size. The *Load one* measure in Fig. 10 shows an increasing activity in minute 9:20 approximately, even more than in the previous experiment when increasing the number of cores to 160.

V. Conclusions

This article presents a novel approach for Human Activity Recognition in healthcare systems for obesity patient monitoring. It comprises a combination of public (labelled) and private (unlabelled) raw data integration, semi-supervised classification with CNN Encoder-Decoder and Spark streaming processing with sliding

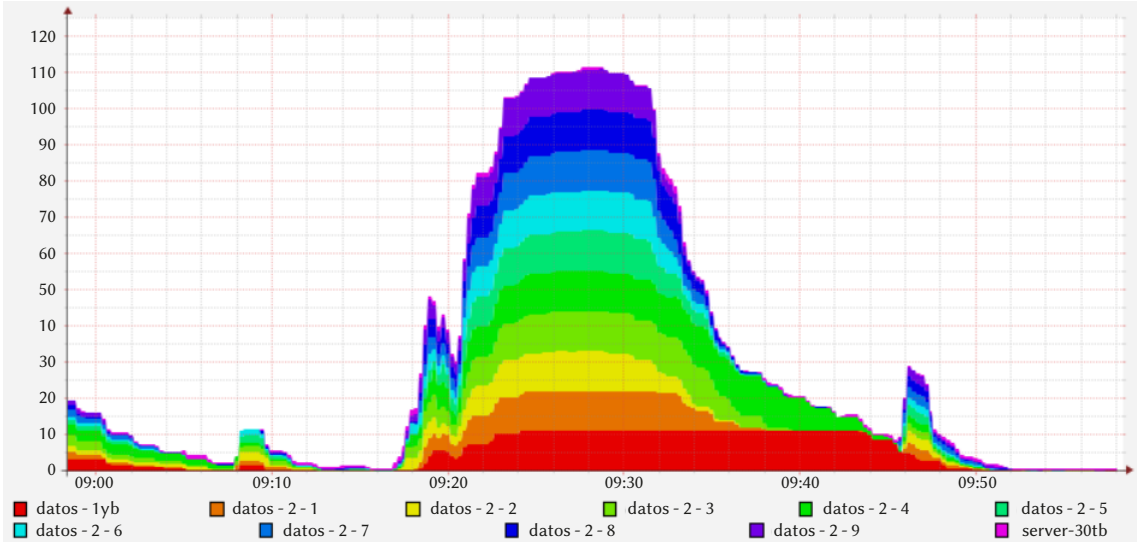


Fig. 9. Load_one. Number of threads per node (40 cores configuration). Plot captured from Ganglia cluster monitoring system for the master node (server-30tb) and 10 worker nodes.

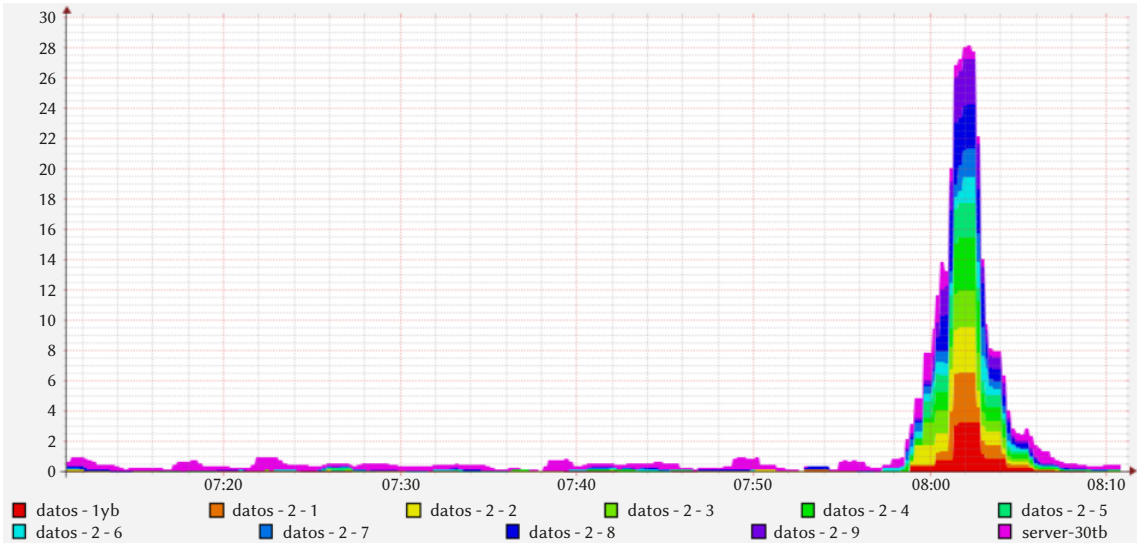


Fig. 10. Load_one. Number of threads per node (160 cores configuration). Plot captured from Ganglia cluster monitoring system for the master node (server-30tb) and 10 worker nodes.

window, to allow continuous activity recognition. The proposal has been validated in the context of a real-world case study with a group of 300 overweight patients in the healthcare system of Andalusia (Spain), classifying close to 30 TBs of accelerometer sensor-based data in real-time conditions, which is crucial for long-term daily patient monitoring.

The experimental results demonstrate that our proposed method can achieve significant Fm-scores training the model even with 100% of unlabelled data (proportion of data labelled used for train), since from this point the results decrease below to 0.8 of Fm-score. Finally, we choose the amount of 80% of unlabelled data, since at this percentage, the model reach a trade-off result (Fm-score = 0.90) between Fm-score and amount of unlabelled data added to the model. Moreover, in order to demonstrate the performance of our model we observe that the distributions of the predictions in unlabelled data are reasonable, as shown in Fig. 7.

In addition, an Spark streaming process for the activity classification was implemented in a cluster computing platform to be able to classify the raw data sensor in real-time. For this propose, we found out the best configuration to minimise the running computation time of the streaming classification, using the cluster with 40 cores and predicting with streaming batch size of 1 GB, being able to classify one week of daily patient monitoring in approximately 8 seconds.

The proposed approach represents a step forward to meet the challenges identified in a recent survey [3], which mainly consist in the generation of real-time activity recognition platforms and the development of more accurate unsupervised modelling for this problem. As argued by authors of this survey, the performance of deep learning still relies on labelled samples to a large extent, which added to the fact that acquiring sufficient activity labels is expensive and time-consuming, makes unsupervised activity recognition an urgent task. Our semi-supervised deep learning on Spark streaming processing is a solution in this direction.

Future lines of research include the generation of advanced visualisations and alarms system to support practitioners in healthcare in patient monitoring. From the perspective of prediction models, the development and use of new ensemble semi-supervised methods will enhance the precision in this kind of environments, where unlabelled data continuously flow in streams and should be properly processed as fast as they are captured.

Appendix

In the following we present the complete list of *Additional Experiments* presented in section IV subsection C. These experiments have been carried out to study the impact of specific design decisions in the context of the downstream task.

A. First Experiment

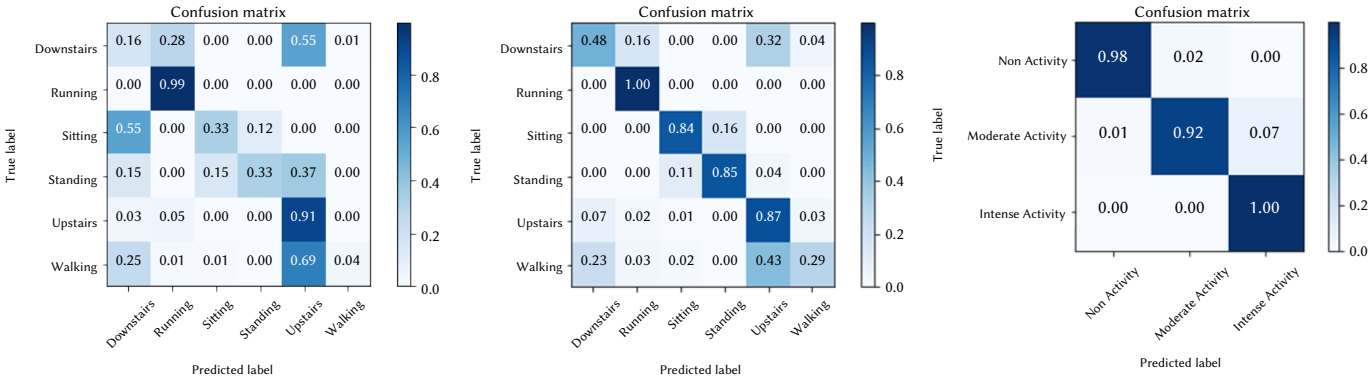
In this first experiment, we wanted to see whether the model was able to pass *"the eye test"* without taking into account the semi-supervised approach. For this propose, the model was trained only with labelled data from WISDM dataset without considering our private unlabelled data in the training phase. Afterwards, the prediction of a randomly chosen sample (five days prediction) from our 30TB private unlabelled data set was performed, as shown in Fig. 11. Can be observed that the model predicts running and walking downstairs as the main activities of the patient even during the nights and rarely predicts the activities of standing and sitting, despite the fact that these are the most prevalent behaviours among obese patients. Overall, it may be said the model is not able to make reasonable predictions if the unsupervised task is not used in the training regime.

B. Second Experiment

In a second experiment, the proposed semi-supervised methodology has been synthetically evaluated by using another public dataset as a simulation of the unsupervised portion. In this sense, HUGADB dataset has been considered as *"unlabelled dataset"* since it contains in overall the closest axis distributions to the sensorised data of WISDM dataset and the lowest standard deviation in the data as shown in Fig. 3. Hence, we study the performance of our semi-supervised CNN Encoder-Decoder model trained with a combination of WISDM as public annotated data WISDM and 70% of HUGADB dataset as a simulation of the unsupervised portion to classify the activities in HUGADB, as observed in Fig. 12. First, the model has been trained only with labelled data from WISDM without considering unlabelled data in the training phase. Afterwards, the model has been validated in the remaining 30% of HUGADB dataset, as shown in Fig. 12a. Subsequently, to demonstrate the feasibility of our semi-supervised approach the model has been trained again but this time 70% of HUGADB has been taken into account as a simulation of the unsupervised portion in the training phase. As previously, the model has been validated in the remaining 30% of HUGADB dataset, as shown in Fig. 12b. It can be appreciated that our semi-supervised approach improves the predictions results from 0.414 to 0.704 in terms of Fm-score, as shown in Table IV.

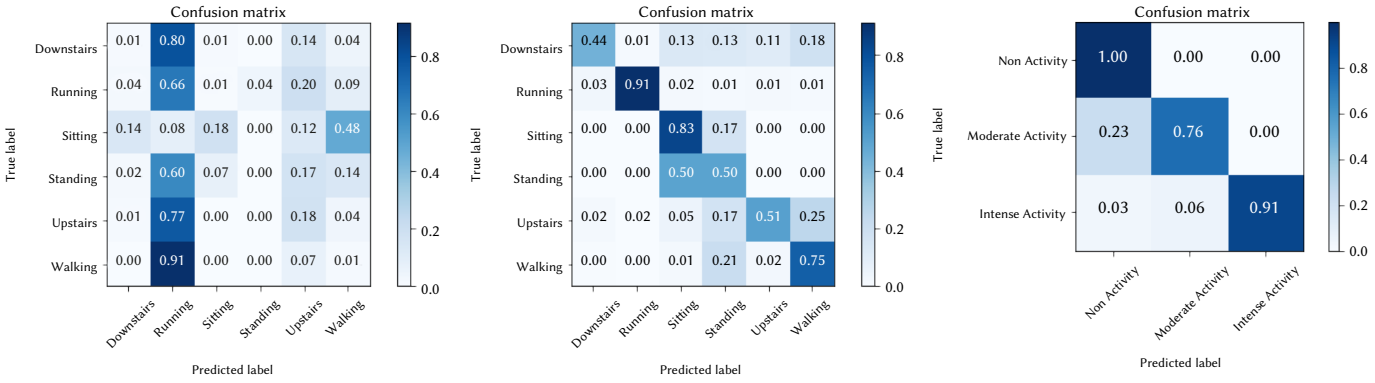
This second experiment has been repeated with another public dataset as a simulation of the unsupervised portion to verify the quality of the semi-supervised approach. In this case PAMAP2 has been selected since it contains different axis distributions to the sensorised data of WISDM dataset and the highest standard deviation

Fig. 11. Activity classification of a randomly chosen sample (five days prediction) from our 30TB private unlabelled data set. For these predictions, the model has been trained only with labelled data from WISDM dataset without considering our semi-supervised strategy with private unlabelled data in the training phase.



(a) 0% of unlabelled data (b) 70% of unlabelled data (c) 70% of unlabelled data

Fig. 12. Illustration of confusion matrices showing the sensitivity of the networks for each individual when varying the percentage of unlabelled data in the training regime from 0% to 70% (HUGADB as a simulation of the unsupervised portion). In addition, the dimensionality of HAR classification problem has been reduced into three basic classes in Fig.(c): Non Activity (sitting and standing), Moderate Activity (walking, walking downstairs and walking upstairs) and Intense Activity (running).



(a) 0% of unlabelled data (b) 70% of unlabelled data (c) 70% of unlabelled data

Fig. 13. Illustration of confusion matrices showing the sensitivity of the networks for each individual when varying the percentage of unlabelled data in the training regime from 0% to 70% (PAMAP2 as a simulation of the unsupervised portion). In addition, the dimensionality of HAR classification problem has been reduced into three basic classes in Figure (c): Non Activity (sitting and standing), Moderate Activity (walking, walking downstairs and walking upstairs) and Intense Activity (running).

in the data as shown in Fig. 3. It's shown in Table IV, how the semi-supervised methodology increases the predictions results from 0.129 to 0.667 in terms of Fm-score. Also, in Fig. 13 the semi-supervised strategy increases the accuracy in all the classes.

TABLE IV. Metrics Evaluation With Varying Number of Unlabelled Examples in Training Set. HUGADB and PAMAP2 Datasets Have Been Taken as a simulation of the Unsupervised Portion to Synthetically Evaluate the Proposed Semi-supervised Methodology

Metrics: Public data (labelled) + Public data (Unlabelled)				
Labelled/Unlabelled	%	acc	recall	Fm-score
WISDM/HUGADB	0%	0.461	0.461	0.414
WISDM/HUGADB	70%	**0.722**	**0.722**	**0.704**
WISDM/PAMAP2	0%	0.173	0.173	0.129
WISDM/PAMAP2	70%	**0.667**	**0.667**	**0.667**

In spite of improving the quality of results with our semi-supervised approach, the model starts to show limited predictions in detecting some activities. For example, for the model it's difficult to predict downstairs and walking, since, if we see the patterns between walking and downstairs, they are characterised with very close signal shapes in movements, as commented before in the paper. Furthermore, static activities can be recognised easily than periodic activities (running, walking, etc.). However, highly similar postures (sitting and standing) create great complexities in case of separation due to notable overlapping in feature space as observed in Fig. 13b. In general, the dimensionality of HAR classification problem can be reduced by classifying into three basic types: Non Activity (sitting and standing), Moderate Activity (walking, walking downstairs and walking upstairs) and Intense Activity (running) as shown in Fig. 12c and Fig. 13c. It's worth to note that we can obtain promising results that will allow us to provide patient activity information to doctors which is essential to prevent obesity. In conclusion, it can be said that the semi-supervised approach achieve improvements in the results, when trying to predict activities from a dataset that the model has never seen before. With the semi-supervised strategy the model can extract important features from the unlabelled data that help us to make better predictions.

Acknowledgements

This work has been partially funded by the Spanish Ministry of Science and Innovation via Grant TIN2017-86049-R (AEI/FEDER, UE) and Andalusian PAIDI program with grant P18-RT-2799.

References

[1] K. González, J. Fuentes, J. L. Márquez, "Physical inactivity, sedentary behavior and chronic diseases," *Korean journal of family medicine*, vol. 38, no. 3, p. 111, 2017.

[2] W. L. Haskell, S. N. Blair, J. O. Hill, "Physical activity: health outcomes and importance for public health policy," *Preventive medicine*, vol. 49, no. 4, pp. 280–282, 2009.

[3] J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[4] P. Bet, P. C. Castro, M. A. Ponti, "Fall detection and fall risk assessment in older person using wearable sensors: a systematic review," *International journal of medical informatics*, 2019.

[5] A. Bourke, J. O'brien, G. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait & posture*, vol. 26, no. 2, pp. 194–199, 2007.

[6] F. Bagala, C. Becker, A. Cappello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra, J. Klenk, "Evaluation of accelerometer-based fall detection algorithms on real- world falls," *PloS one*, vol. 7, no. 5, 2012.

[7] F. M. Palechor, A. De la Hoz Manotas, P. A. Colpas, J. S. Ojeda, R. M. Ortega, M. P. Melo, "Cardiovascular disease analysis using supervised and unsupervised data mining techniques.," *JSW*, vol. 12, no. 2, pp. 81– 90, 2017.

[8] D. Arifoglu, A. Bouchachia, "Activity recognition and abnormal behaviour detection with recurrent neural networks," *Procedia Computer Science*, vol. 110, pp. 86– 93, 2017.

[9] G. Kalouris, E. I. Zacharaki, V. Megalooikonomou, "Improving cnn-based activity recognition by data augmentation and transfer learning," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1, 2019, pp. 1387–1394, IEEE.

[10] A. Papagiannaki, E. I. Zacharaki, K. Deltouzos, R. Orselli, A. Freminet, S. Cela, E. Aristodemou, M. Polycarpou, M. Kotsani, A. Benetos, *et al.*, "Meeting challenges of activity recognition for ageing population in real life settings," in *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2018, pp. 1–6, IEEE.

[11] C. A. Ronao, S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert systems with applications*, vol. 59, pp. 235–244, 2016.

[12] Y. Saez, A. Baldominos, P. Isasi, "A comparison study of classifier algorithms for cross-person physical activity recognition," *Sensors*, vol. 17, no. 1, p. 66, 2017.

[13] T. Lv, X. Wang, L. Jin, Y. Xiao, M. Song, "Margin- based deep learning networks for human activity recognition," *Sensors*, vol. 20, no. 7, p. 1871, 2020.

[14] F. Cruciani, A. Vafeiadis, C. Nugent, I. Cleland, P. McCullagh, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen, R. Hamzaoui, "Feature learning for human activity recognition using convolutional neural networks," *CCF Transactions on Pervasive Computing and Interaction*, vol. 2, no. 1, pp. 18–32, 2020.

[15] J. R. Kwapisz, G. M. Weiss, S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74– 82, 2011.

[16] A. Reiss, D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th International Symposium on Wearable Computers*, 2012, pp. 108–109, IEEE.

[17] R. Chereshnev, A. Kertész-Farkas, "Hugadb: Human gait database for activity recognition from wearable inertial sensor networks," in *International Conference on Analysis of Images, Social Networks and Texts*, 2017, pp. 131–141, Springer.

[18] M. Zhang, A. A. Sawchuk, "Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 1036– 1043.

[19] D. Balabka, "Semi-supervised learning for human activity recognition using adversarial autoencoders," in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, UbiComp/ISWC '19 Adjunct, New York, NY, USA, 2019, p. 685–688, Association for Computing Machinery.

[20] O. D. Lara, M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[21] A. Subasi, K. Khateeb, T. Brahimi, A. Sarirete, "Human activity recognition using machine learning methods in a smart healthcare environment," in *Innovation in Health Informatics*, M. D. Lytras, A. Sarirete Eds., Next Gen Tech Driven Personalized MedSmart Healthcare, Academic Press, 2020, pp. 123 – 144, doi: https://doi.org/10.1016/B978-0-12-819043-2.00005-8.

[22] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes- Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International workshop on ambient assisted living*, 2012, pp. 216–223, Springer.

[23] L. Bao, S. S. Intille, "Activity recognition from user- annotated acceleration data," in *International conference on pervasive computing*, 2004, pp. 1–17, Springer.

[24] W. Wu, S. Dasgupta, E. E. Ramirez, C. Peterson, G. J. Norman, "Classification accuracies of physical activities using smartphone motion sensors," *Journal of medical Internet research*, vol. 14, no. 5, p. e130, 2012.

[25] Y. Chen, Y. Xue, "A deep learning approach to human activity recognition based on single accelerometer," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, 2015, pp. 1488–1492, IEEE.

[26] N. Y. Hammerla, S. Halloran, T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *arXiv preprint arXiv:1604.08880*, 2016.

[27] M. Zeng, T. Yu, X. Wang, L. T. Nguyen, O. J. Mengshoel, I. Lane, "Semi-supervised convolutional neural networks for human activity recognition," in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 522–529.

[28] A. D. Antar, M. Ahmed, M. A. R. Ahad, "Challenges in sensor-based human activity recognition and a comparative analysis of benchmark datasets: a review," in *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, 2019, pp. 134–139, IEEE.

[29] S. Slim, A. Atia, M. Elfattah, M. Mostafa, "Survey on human activity recognition based on acceleration data," *International Journal of Advanced Computer Science and Applications*, vol. 10, pp. 84–98, 2019.

[30] Z. Hussain, M. Sheng, W. E. Zhang, "Different approaches for human activity recognition: A survey," *arXiv preprint arXiv:1906.05074*, 2019.

[31] A. Gupta, K. Gupta, K. Gupta, K. Gupta, "A survey on human activity recognition and classification," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020, pp. 0915–0919, IEEE.

[32] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, pp. 1–27, 2010.

[33] F. Foerster, M. Smeja, J. Fahrenberg, "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring," *Computers in human behavior*, vol. 15, no. 5, pp. 571–583, 1999.

[34] J. R. Kwapisz, G. M. Weiss, S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, p. 74–82, Mar. 2011, doi: 10.1145/1964897.1964918.

[35] A. Reiss, M. Weber, D. Stricker, "Exploring and extending the boundaries of physical activity recognition," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*, 2011, pp. 46–50.

[36] M. Zhang, A. A. Sawchuk, "Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors," in *ACM International Conference on Ubiquitous Computing (Ubicomp) Workshop on Situation, Activity and Goal Awareness (SAGAware)*, Pittsburgh, Pennsylvania, USA, September 2012.

[37] R. Chereshnev, A. Kertész-Farkas, "Hugadb: Human gait database for activity recognition from wearable inertial sensor networks," in *Analysis of Images, Social Networks and Texts*, Cham, 2018, pp. 131–141, Springer International Publishing.

[38] H. He, E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[39] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "Smote: synthetic minority over- sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[40] K. T. Nguyen, F. Portet, C. Garbay, "Dealing with imbalanced data sets for human activity recognition using mobile phone sensors," 2018.

[41] S. Ertekin, J. Huang, L. Bottou, L. Giles, "Learning on the border: active learning in imbalanced data classification," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, pp. 127–136.

[42] D. A. Cieslak, T. R. Hoens, N. V. Chawla, W. P. Kegelmeyer, "Hellinger distance decision trees are robust and skew-insensitive," *Data Mining and Knowledge Discovery*, vol. 24, no. 1, pp. 136–158, 2012.

[43] L. G. Fahad, S. F. Tahir, M. Rajarajan, "Activity recognition in smart homes using clustering based classification," in *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 1348–1353, IEEE.

[44] A. Bulling, U. Blanke, B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," vol. 46, no. 3, 2014, doi: 10.1145/2499621.

[45] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, 2018, pp. 1–2, IEEE.

[46] S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

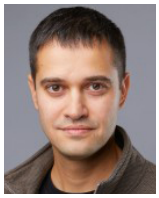
[47] J. Snoek, H. Larochelle, R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.

Sandro Hurtado

Sandro Hurtado PHD student in Bioinformatics applications, with a Degree in Health Engineering with a mention in Biomedicine (2017) and a Master's Degree in Software Engineering and Artificial Intelligence (2019), from the University of Málaga. His main lines of research are the development of ontologies in the domain of Gene Regulation Networks and Software applications for the collection, consolidation and analysis of clinical data, thus providing medical and biological information to researchers and doctors in this field.

Prof. José García-Nieto

He received his Ph.D. degree in computer science with honors from the University of Málaga (Spain) in 2013, and his degree in engineering with honors in 2006, also from the University of Málaga. His current research interests include optimisation and machine learning algorithms, Big Data processing, Web Semantics and their application to real-world problems in interdisciplinary domains of Precision Agriculture, Bioinformatics and Smart Cities. His research activity has resulted in scientific publications consisting of 40 journal articles, 4 book chapters and more than 40 papers in referred international and national conferences.



Prof. Anton Popov

Anton Popov is the Artificial Intelligence/Deep Learning technical lead at Ciklum with 15+ years of experience in the development and implementation of bio-signal analysis and classification algorithms. He is affiliated with Igor Sikorsky Kyiv Polytechnic Institute as an Associate Professor. Since 2002, Anton has been a member of the IEEE Engineering in Medicine and Biology Society, has published 100+ papers.

Prof. Ismael Navas-Delgado

Computer Engineer (2002), Doctor by the University of Málaga (2009) and Master in Cell Biology and Molecular Biology (2008). His research is developed within the KHAOS Research group participating in multiple research projects (15), being the second principal investigator of the projects TIN2014-58304-R and TIN2017-86049-R, and technology transfer (2). His research activity focuses on the integration of data through the use of semantic technologies and their application to Life Sciences.

An Efficient Bet-GCN Approach for Link Prediction

Rahul Saxena[1,2], Spandan Pankaj Patil[3], Atul Kumar Verma[1], Mahipal Jadeja[1], Pranshu Vyas[1], Vikrant Bhateja[4], Jerry Chun-Wei Lin[5] *

[1] Department of Computer Science and Engineering, Malaviya National Institute of Technology Jaipur, Jaipur (India)
[2] Department of Information Technology, Manipal University Jaipur, Jaipur (India)
[3] Department of Electrical Engineering, Malaviya National Institute of Technology Jaipur, Jaipur (India)
[4] Department of Electronics Engineering, Faculty of Engineering and Technology, Veer Bahadur Singh Purvanchal University, Shahganj Road, Jaunpur-222003, Uttar Pradesh (India)
[5] Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen (Norway)

Received 5 July 2022 | Accepted 12 October 2022 | Early Access 1 February 2023

Abstract

The task of determining whether or not a link will exist between two entities, given the current position of the network, is called link prediction. The study of predicting and analyzing links between entities in a network is emerging as one of the most interesting research areas to explore. In the field of social network analysis, finding mutual friends, predicting the friendship status between two network individuals in the near future, etc., contributes significantly to a better understanding of the underlying network dynamics. The concept has many applications in biological networks, such as finding possible connections (possible interactions) between genes and predicting protein-protein interactions. Apart from these, the concept has applications in many other areas of network science. Exploration based on Graph Neural Networks (GNNs) to accomplish such tasks is another focus that is attracting a lot of attention these days. These approaches leverage the strength of the structural information of the network along with the properties of the nodes to make efficient predictions and classifications. In this work, we propose a network centrality based approach combined with Graph Convolution Networks (GCNs) to predict the connections between network nodes. We propose an idea to select training nodes for the model based on high edge betweenness centrality, which improves the prediction accuracy of the model. The study was conducted using three benchmark networks: CORA, Citeseer, and PubMed. The prediction accuracies for these networks are: 95.08%, 95.07%, and 95.3%. The performance of the model is comprehensive and comparable to the other prior art methods and studies. Moreover, the performance of the model is evaluated with 90.13% for WikiCS and 87.7% for Amazon Product network to show the generalizability of the model. The paper discusses in detail the reason for the improved predictive ability of the model both theoretically and experimentally. Our results are generalizable and our model has the potential to provide good results for link prediction tasks in any domain.

Keywords

Graph Convolution Network (GCN), Graph Theory, Link Prediction, Network Centrality, Social Networks.

DOI: 10.9781/ijimai.2023.02.001

I. Introduction

Social Networks have been the primary source of information exchange between people for more than a decade now. The flow of information in this era depends heavily on the interactions of people with their peers and friends, such as liking a post, following a page, buying products, etc. Both the social networking websites and the e-commerce website are influenced by this fact. Miao et al. [1] discusses the impact of online customer reviews on product returns. The study found that the influence is even greater for sellers with good quality or branded products. Ullal et al. [2] also concluded in their study that customer reviews can significantly influence the selling and buying behavior of e-commerce companies. There are many such studies that prove how important the connections a person has are. A person's opinion and thinking are strongly influenced by the views and activities of their social environment. This ideology, in turn, is used by companies to identify the potential customers/buyers in the near future. This is done by analyzing the network of existing customers and identifying people who have the same preferences, characteristics, etc. This correlation in the characteristics of the two people forms the basis for a friendship relationship between them. This concept of link analysis and prediction is not only useful in product recommendation, but also in various areas of network science. Link prediction in network science is an important research area to understand the growth and evolution of the network. The idea of link prediction [3], [4] and analysis is of great importance in community detection, influence analysis, anomaly

* Corresponding author.
E-mail addresses: 2019rcp9153@mnit.ac.in (R. Saxena), 2018uee1353@mnit.ac.in (S. P. Patil), 2019rcp9050@mnit.ac.in (A. K. Verma), mahipaljadeja.cse@mnit.ac.in (M. Jadeja), 2018ucp1444@mnit.ac.in (P. Vyas), bhateja.vikrant@ieee.org (V. Bhateja), jerrylin@ieee.org (J. C.-W. Lin).

detection, recommendation, etc. [5] where the available information plays an important role in identifying the linking patterns. Further, link prediction has a substantial role in the study of protein-protein interaction patterns and prediction of the linkage between the unconnected protein molecules [6]. Similarly, Marcus et al. [7] have used the link prediction to study the time-evolving criminal network. Likewise, there are many applications and related areas where link prediction has played a significant role. Although researchers have proposed various link prediction models and methods, still there is a lot of scope for improvement. With the advancements in deep learning for graphs, the task of link prediction has gained increased attention. This is because deep learning techniques for graphs provide highly accurate predictions over the limited training data.

In this paper, we present the task of link prediction using a Graph Convolutional Network (GCN). The key to this idea lies in the selection of the training pool based on network centrality. This idea is explored in detail in section 4 of the paper. As a result, the link prediction task has higher accuracy given a limited training dataset, since the aggregation of the neighborhood improves due to the selection of edges based on their importance. Therefore, the contributions of the manuscript can be highlighted as follows:

- We proposed an efficient GCN-based link prediction technique where the links of the training set are selected based on edge betweenness centrality.
- The utilized justification of edge betweenness centrality is based on the selection of the training set for GCN.
- Detailed comparison of the results obtained with the current state of the art methods for link prediction.

The flow of the paper is organized as follows: Section I gives a brief introduction to link prediction, its applicability, and the contribution of the manuscript. Section II gives an overview of the state of the art in link prediction methods. Also, Graph Convolutional Networks (GCN) and their applicability to the task of link prediction are discussed in this section. Section III discusses the proposed method, its correctness and modification of the conventional GCN-based link prediction. The section also addresses the importance of network centrality to the link prediction task. Section IV highlights the experimental setup, description of the considered datasets and explanation of the proposed model. Section V discusses the results obtained with the proposed model. In addition, the results are compared with other state-of-the-art implementations over the datasets. Finally, Section VI summarizes the results of the study and highlights some future directions to be further explored.

II. Literature Survey

This section gives a brief literature review of the state of the art, highlighting link prediction and Graph Convolutional Networks. It also discusses the latest graph deep learning based architectures and frameworks to tackle the task of *link prediction*. The section focuses on the need and scope of deep learning techniques for link prediction.

A. Link Prediction

The task of link prediction can be defined as predicting whether or not two nodes will form a link in the future.

So, given a graph, if two nodes are not connected at time t, what is the probability that they will be connected at time $(t+1)$? Taking this idea further, there may be many unconnected nodes in the graph at a given time. So the task is to correctly predict the possible connections between nodes at a given time in the network.

To formulate this more formally, consider a graph $G(V, E)$ defined as follows:

V: Set of vertices or nodes in the graph such that

$$V = \{v_1, v_2, \ldots v_n\} \; \forall \; n \geq 1$$

E: Set of edges or links as $E = \{e_1, e_2, \ldots e_m\} \; \forall \; m \geq 1$

This is the graphical structure at time t_0. At some time $t_1 > t_0$ the graphical structure evolves as $G(V, E')$ suchthat $E' = \{e_1, e_2, \ldots e_k\} \; \forall \; m \geq 1$ and $k \geq m$. Our goal is to predict the edge set E'' for the graph G with the same number of nodes and an increased number of edges as a result of linkages between the disconnected nodes of the graph based on the information at time t_0 of the graph. This edge set should approximate the actual edge set E'.

Fig. 1 shows a graphical network in which the dashed edges represent the possible connections between the unconnected nodes at a given time in the near future. An interesting fact about the creation of connections is that each group of nodes tries to complete its *Triadic closure* [8]. According to Granovetter's theory of *Strength of Weak Ties* [9], if there is a connection between nodes A-B and A-C, then there is a strong tendency for linkage between B-C. The statement is about the triadic closure property for graphical networks. As an extension to this, there are many node groups in the network in which a pair of nodes attempts to close triads. The links between such pairs of nodes have a high probability of appearing in the future. This is one of the main ideas behind link prediction. Another idea for predicting a link between pairs of nodes is based on the different degree of expansion of the network inside and outside the group. According to Bi et al. [10], the network expansion inside the community is high. The nodes outside the community have fewer linkages, or very few nodes are connected. Apart from these, there are several other concepts for building networks such as stochastic block model [11], stochastic block model with Bayesian context, and stochastic block model with spectral clustering [12], which is the basis for *link establishment* between nodes. Another class of concepts are measures of proximity of nodes such as common neighbors, Jaccard coefficient [13], Adamic/Adar [14], Preferential Attachment Model [15] etc., based on which link establishment between nodes can be expected. These are the conventional approaches to link prediction that have evolved over time. Various improvements to these general ideas have been developed to achieve better and more efficient results. However, the increasing size of networks, aggregation of features in the form of node attributes and information, dynamic evolution of the network, and many other factors pose challenges to the computational ease and predictive ability of the methods. Machine learning/deep learning based approaches to the problem of link prediction are therefore attracting increasing attention. Combining these general ideas with artificial intelligence (AI) and machine learning (ML) based approaches has proven to be successful. The results obtained are very accurate. The remainder of the discussion in this section therefore focuses on the current state of the art in deep learning-based approaches to link prediction over the graph.

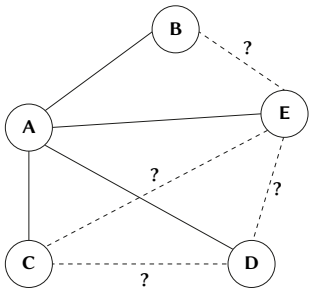


Fig. 1. Network as Graph with possible edges or links between the nodes.

B. Graph Convolutional Networks (GCN)

Graph Convolutional Networks (GCNs) have emerged in recent years as powerful machine learning methods for graph processing [16]. The basic idea behind the operation of convolutional networks is neighborhood aggregation, where the features of each node play a crucial role in decision making. Unlike an image, the structure of the graph is irregular and cannot be mapped to a fixed grid (see Fig. 2). Therefore, the structure of the graph also plays an important role. For this reason, the conventional Convolutional Neural Network (CNN) based approach cannot be used for graph structures. A GCN uses both the network structure and the features of the neighboring node to evaluate the folded value over the considered node. This additional information about the context of the node in the form of the network structure plays an important role in the prediction and classification tasks.

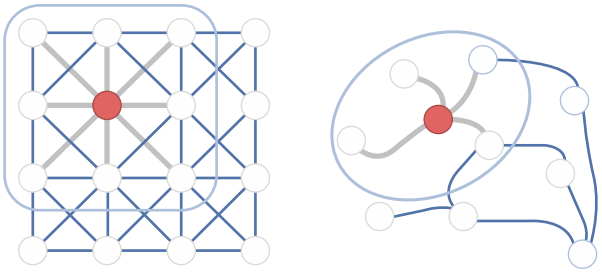


Fig. 2. Image structure v.s. graph structure [17].

The task of modeling a Graph Convolutional Network (GCN) for a graph is solved by two mathematical approaches: *Spectral Graph Theory* and *Spatial Graph Theory*. Spectral Graph Theory requires a Fourier transform based computation of translation in the frequency domain to create a graph Laplacian [17]. Since this requires a detailed mathematical explanation, we will only explain the main steps here. At a high level, the spectral graph convolution in the Fourier domain is defined by applying the filter $g\theta$ to the input signal x:

$$g\theta * x \tag{1}$$

$g\theta$: A diagonal matrix diag(θ) parameterized by $\theta \in R^n$

Since the operator based on spectral graph convolution is a position invariant of the nodes of the graph, the *graph Laplacian* matrix L for a graph G of dimensions $N \times N$ is given as:

$$L = I_N - D^{-1/2}.A.D^{-1/2} = U\Lambda U^T \tag{2}$$

Here A stands for *Adjacency Matrix*, I for *Identity Matrix*, and D for *Diagonal Matrix*. Their product gives the aggregate sum and $D^{-1/2}$ normalizes this product to suppress the effect of high degree nodes. Moreover, L can be factorized using U, which contains eigenvectors of L and Λ with the corresponding eigenvalues. Since L is a positive semi-definite matrix and U is the Fourier basis, the Fourier transform over x can be defined as follows:

$$F(x) = U^T x \tag{3}$$

Hence the inverse is presented as:

$$F^{-1}(\hat{x}) = U^T \hat{x} \tag{4}$$

If F is the Fourier domain space, the graph convolution operator can be defined as an elementwise product:

$$x * Gg = F^{-1}(F(x) \odot F(g)) = U(U^T x \odot U^T g) \tag{5}$$

Comparing equation (5) with equation (1), the final convolution equation of the graph can be given as follows:

$$x * Gg_\theta = Ug_\theta U^T x \tag{6}$$

such that:

$$g_\theta = diag(U^T x), g \in R \tag{7}$$

With g_θ filled with the learning parameters $\theta_{i,j}^k$, the output on layer k can be defined as follows:

$$H_{:,j}^k = \sigma\left(\sum_{i=1}^{f_{k-1}} U\theta_{i,j}^k U^T H_{:,i}^{k-1}\right) (j = 1,2,\dots k) \tag{8}$$

Here, f_{k1-1} and f_k are the number of input and output channels in layer k, respectively, $H_{:,j}^k$ is the output channel in layer k.

However, this spectral convolution has certain limitations. First, computing the eigenvalues of the graph matrix is a computationally intensive task. Second, for very large graphs, the aggregation of neighborhoods for large values of k becomes computationally intensive and degrades the aggregation results. To solve these problems, only a neighborhood of a few hops should be considered in the localization of the filtering process. Therefore, spatial graph convolution methods have gained increasing attention. Thus, by adding formal parameters to the equation (2) and approximating the depth of the network to two, an embedding based on a 2-layer GCN model can be defined as follows:

$$Z = f(A,X) = softmax(K.ReLU(K.X.W^{(0)}).W^{(1)}) \tag{9}$$

Here K is defined as $D^{-1/2}AD^{-1/2}$. The ultimate task is to learn the weights for the model, where $C \times H$ are the trainable weights for $W^{(0)}$. Similarly, HXF are trainable weights for $W^{(1)}$. Here, C refers to the dimensions of the feature vectors, 'F' refers to the dimensions of the resulting vectors, and 'H' is the number of hidden layers. The expression in equation (9) can be further extended depending on the hidden layers in the network. The depth of the network is based on the intuition of the contribution of the k path length of the neighborhood. However, in general, graph networks do not have much impact on neighborhood interactions beyond 2 – 3 path lengths [18]. Therefore, the results of GCN networks at 2 – 3 level are remarkable and impressive; otherwise, the model suffers from the overfitting condition. The final layer of this *spatial* GCN model is guided by a *softmax* function to make predictions. The cross-entropy loss function is considered for training the model:

$$L = -\sum_{y \in Y_l}\sum_{f=1}^{F} Y_{lf}\, lnZ_{lf} \tag{10}$$

Here Y_l is the set of values with their respective labels. The hyperparameters of the model are set to optimize this loss metric, including the learning rate, epochs, layer sizes, etc. A detailed discussion of these parameters can be found in section V of the paper. Further improvements to the model, such as changing the aggregation function, using weighting preferences to cluster the neighborhood, etc., provide a path to advanced versions of GCN such as Graph Attention Model, GraphSage, etc. In the following subsection, we discuss the state of the art regarding the role of GCN/GNN in efficient link prediction execution.

C. Graph Neural Network Based Approaches to Link Prediction

Since the last decade, the world has been experiencing a boom in the research area of graphical neural networks. GNN is a special kind of neural networks characterized by the structures of graphs. Semi-supervised link prediction using label propagation was first introduced by Kashima et al. [19]. This model of link prediction is applicable to multirelational domains and uses auxiliary information such as node similarity. A new fast and scalable algorithm for semi-supervised link prediction was proposed by Raymond et al. [20] for both static and dynamic graphs.

Menon et al. [21] proposed a model that predicts links through Matrix factorization. This model gains knowledge of latent features from the topological structure of the graph. Moreover, the author considered the problem of class imbalance during optimization with stochastic gradient descent and scales. Gao et al. [22] addressed the problem of predicting temporal link prediction. This model integrates the information of graph proximity, global network structure, and node content. The prediction approach called SLiPT (self-training based link prediction using a temporal network) shows better prediction accuracy and was proposed by Zeng et al. [23]. Berton et al. [24] dealt with graph construction in supervised and semi-supervised classification.

To improve performance, Kipf et al. [25] proposed VGAE (Variational Graph Auto-Encoder). This approach uses latent variables and gives better results in predicting links in citation networks. Another approach by Yang et al. [26] defines a new proximity matrix and formulates BANE (Binarized Attributed Network Embedding). In contrast to these methods, Tran et al. [27] focused on a simple but effective architecture. This architecture, named MTGAE (Multi-Task Graph Auto-Encoder), works for unsupervised link prediction and semi-supervised node classification. In the same year, Hisano et al. [28] worked on performance improvement using a simple discrete-time graph embedding approach for link prediction for both temporal cross-sectional network structures. Pan et al. [29] defines (ARGE and ARVGE) adversarial graph embedding framework and demonstrates the efficiency of the algorithm through experiments.

To reduce the information loss, Di et al. [30] recently presented an approach to expand the normal neighborhood when aggregating GNNs. This approach is suitable for graph link prediction, supervised and semi-supervised graph classification, and graph edge classification. Recently, Zhang et al. [31] have advanced research in link prediction using the SegNMF method. This method claims to provide better accuracy in temporal link prediction than the previously developed method.

All the state of art methods discussed above take into account the spatial embeddings of the node into account where the nodes are selected randomly for training the model. Further, the test data taken for predicting the accuracy of the model for link prediction task is very small (5 - 10%). Further few recent state of art models proposed for link prediction task in [27], [32], [33] are designed for solving problems of specific domain only. The complexity of these models tend to increase with the increase in the size of the network. So, the models do not guarantee to generalize well for networks of different nature, size and domain. Thus, the applicability of GNNs for this task on various problems in different domains can still be improved and extended. In summary, following gaps are identified and these gaps motivate us to propose the solution:

- There are no/limited approaches for predicting links between nodes in a graph with limited information available for training the network [34], [35].
- There is no centrality-based approach that can improve the prediction capability of GCN model to identify connections between nodes.
- There is a need for a generalized model which is dependent upon the structural aspects of the underlying network and independent of the application [27], [32],[33].

III. Bet-GCN Approach to Link Prediction

This section discusses how *edge betweenness centrality measure* in combination with *Graph Convolutional Network* (*GCN*) enhances the task of predicting links between unconnected nodes of the network. The content of this section has been divided into the following subsections:

- Basics of edge betweenness centrality.
- Link prediction as a binary classification problem.
- Justification of edge betweenness based training set selection.

A. Basics of Edge Betweenness Centrality

The concept of network was proposed by Roethlisberger et al. [36]. This concept defines the importance of a node based on various attributes such as the degree of a node, closeness with the nodes in its neighborhood, the number of nodes for which it is central, etc., i.e., it identifies the potential of the underlying node in terms of guiding and channeling the flow of information in the network. Based on this, there can be several *centrality measures*, e.g., degree centrality, closeness centrality, PageRank and hits centrality and **betweenness** centrality, etc. In the paper by Saxena and Jadeja [37], all these centrality measures are discussed in detail. Moreover, we investigate the suitability of the centrality measures to find out important nodes depending on the problem or task. In this section, we restrict ourselves to the betweenness centrality measure. The interconnectedness centrality measure is a centrality measure based on the shortest path. Thus, the importance of a node is recognized based on the maximum number of shortest paths in which it participates. This path-based measure, proposed by Freeman et al. [38] has two conjectures: i) *node betweenness* ii) *edge betweenness*. However, one is the implication of the other. The notion of edge betweenness centrality suggests that an edge is involved in the maximum number of shortest paths. Looking at the Fig. 3, the edge **AB** has the highest betweenness centrality compared to other edges in the network. This is because the edge AB is part of most shortest paths between any pair of vertices of the given graph. Consider two sets: set $X = \{A, F, G, H\}$ and set $Y = \{B, C, E\}$. All shortest paths from any vertex of set X to any vertex of set Y use edge AB.

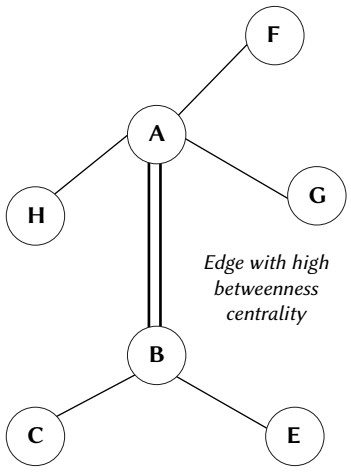


Fig. 3. Graphical network with edge AB as high betweenness centrality edge.

Formally, to identify the *betweenness centrality* of node x, we have:

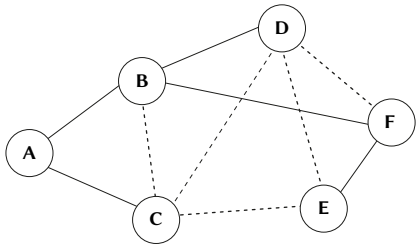
$$c_{bet(x)} = \sum_{y,z \neq x, \sigma_{yz} \neq 0} \frac{\sigma_{yz}(x)}{\sigma_{yz}} \tag{11}$$

Here σ_{yz} is the total number of shortest paths leading from y to z, and $\sigma_{yz}(x)$ refers to the number of these paths that pass through x. Thus, the more shortest paths emanating from node x, the more central node x is. Edges that have one of these nodes as an endpoint have high edge *betweenness centrality*. Edges with high betweenness centrality are especially important in a large network. Endpoint nodes of an edge with high betweenness centrality are more reachable in

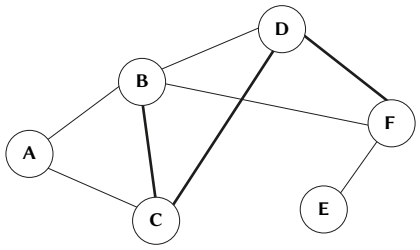
the network with shorter path lengths. Thus, this property allows us to take advantage to increase node coverage. We use this concept to improve the performance of the GCN. An in-depth analysis and execution of this concept is presented in Section IV of the paper. In the following subsection, we discuss the approach to link prediction in a given network as a binary classification problem.

B. Link Prediction as a Binary Classification Problem

The task of link prediction is to determine whether or not a pair of nodes will have a link between them in the future. Consider a graph $G(V, E)$ at a given time t with V as a set of nodes and E as a set of edges, as shown in Fig. 4a. The graph shows various possible links between pairs of nodes that can occur at time $t + \delta t$ (represented by dotted lines). At time $t + \delta t$, as shown in Fig. 4b, some expected connections appear (shown by bold edges in the graph), while some of them do not. Graph Convolution Networks (GCN) captures the properties of the nodes in addition to the topological and structural information of the network. This helps in finding close correlations and probable neighbors of a node based on their behavioral similarities in the network. However, to do this, we must first model the problem in a structure of < *feature, target* > pairs to apply a graph-based machine learning model.



(a) Graphical network at time t



(b) Graphical network at time $t + \delta t$

Fig. 4. Evolution of Graph G from time t to time $t + \delta t$.

Each node has a feature set (vector) associated with it. It consists of a collection of information about the node, its properties, etc., that define and identify that node in the network. An edge has two nodes as its endpoint, so the final feature set in this case is a combination of the feature vectors of the two nodes that form the edge. If we consider the edge as (u, v), where u and v are the nodes under consideration, the final feature vector is as follows:

$$feature\ vector = feature\ vector(u) \cup feature\ vector(v)$$

Further, depending upon whether the two nodes u and v are connected or not, target (label) can be defined as:

target = 1 , if (u, v) is connected

= 0 , otherwise

Thus, by separating the connected and disconnected nodes with the labels 1 and 0, respectively, we can create a *pair* (see Table I). It is now possible to process the data with a machine learning model to make predictions. For the given graph G, we can select a pool of edges for training the GCN model. Here, each edge is accompanied by its label. Also, the GCN model uses the node feature information to train the model. The edges of the test dataset can be randomly selected to test the accuracy of the model for the binary classification problem, i.e., predict 1 for each connected pair and 0 for each unconnected pair.

TABLE I. Graph Edges With Labels

Connected Edge	Label	Unconnected Edge	Label
A-B	1	B-C	0
A-C	1	C-E	0
B-D	1	C-D	0
B-F	1	D-F	0
E-F	1	E-D	0

For real networks, the model is created by randomly hiding some edges from the network. The remaining network is then used to train the GCN. The hidden edges are then used to test the adequacy of the model. This simulative technique is as good as analyzing the temporal transition of the graph because: i) we do not have timestamp snapshots of the real networks at persistent intervals and ii) the network changes its structure gradually. Thus, the network is not significantly perturbed. For these two reasons, we consider only a single real graph as input. In the following subsection, we discuss and analyze how *edge betweenness centrality* based training set selection improves the efficiency of the GCN model for link prediction.

C. Justification of Edge Betweenness Centrality Based Training Set Selection

So far, we have discussed the *edge betweenness centrality* measure and the strategy for solving the link prediction. In this subsection, we will analyze the basis of our proposal:

Training set selection based on edge betweenness centrality improves GCN training efficiency. For this purpose, let us consider a graph $G(V, E)$ for which holds:

V: Set of vertices or nodes defined as $\{v_1, v_2, ..., v_n\} \in V$

E: Set of edges or links defined as $\{e_1, e_2, ..., e_k\} \in E$ such that $n, k > 1$

Let X be the feature matrix defined as:

$$X = \{\{x_{11}, x_{12}, \dots, x_{1m}\}, \dots, \{x_{n1}, x_{n2} \dots x_{nm}\}\} \quad (12)$$

In general, we have $n > m$ (size of training data (number of nodes) > length of a feature vector) to avoid the condition of overfitting during the training process.

Now a set of edges is chosen from the set E to generate a test set t_1 containing t_1. The t_1 is a subset of E containing all connected pairs of nodes. For all these edges (or node pairs), the class label set l_1 is defined as 1. Now, a few random unconnected node pairs are selected from the set $Complement(E)$ or $\overline{E}$ to generate another test set t_2. The corresponding label set for the node pairs of the set t_2 is defined as l_2 with label value 0. Combining the test sets t_1 and t_2, the final test set t can be defined as:

$$t = t_1 \cup t_2 \quad (13)$$

Corresponding to it, the label set L for this test set t can be defined as:

$$L = l_1 \cup l_2 \quad (14)$$

Deleting edges from the graph G creates a graph G', where the edge set of G' is defined as $E' = E - t$. From this residual graph, the training set is constructed in the same way as the test set. Based on this training set, the predicted set of labels for the edges selected from the test set t is obtained as L'. Thus, the objective of the problem can be formulated as follows:

(1) To obtain predicted label set L', we use the GCN model for the edges in test set t, which approximates the label set L i. e., Min. $(L' - L)$ $\forall$ edges in t.

(2) With respect to identification of such a subset, the following observations are made:
- The subset of edges (or node pairs) selected based on the betweenness centrality measure improves the training efficiency of the model.
- The probability of random selection of such an edge set to produce a predicted label set L' is nearly zero.

(3) Let us try to infer the validity of the first statement.
- E'' contains subset of edges chosen randomly. Let this subset be named as E_1.
- E'' contains top d edges based on the betweenness centrality score. Let this subset be named as E_2.

Further, it is assumed that *Cardinality*(E_1) and *Cardinality*(E_2) are the same. Let us consider the first edge from each subset. Let a be the edge chosen from E_1 and b be the edge chosen E_2. Let σ_b represent the edge betweenness centrality of node b and σ_a refer to the edge betweenness centrality of node a. Thus, it is obvious that:

$$\sigma_b > \sigma_a \tag{15}$$

Further, we also assumed that the edge sets E_1 and E_2 are disjoint, i.e., no edges are common to the two sets. Then, extending the above expression for $1 \le i \le l$:

$$\sum_{i=1}^{l} \sigma_{(b_i)} > \sum_{i=1}^{l} \sigma_{(a_i)} \tag{16}$$

Equation (16) holds for a fixed path length p, for the paths covered by the edges in E_1 and E_2. As can be seen from the description of GCN in Section II, it is well known that the neighborhood contribution beyond path length 2 or 3 is not beneficial because of the vanishing gradient problem over the graph Laplacian. So the value of p is $\in \{1, 2\}$. As per Section III, edges with high betweenness centrality allow for greater network coverage with shorter path length. This means that the node coverage (number of reachable nodes) from the nodes of the set E_2 (say ϕ) will be larger than the number of reachable nodes from the nodes of the set E_1 (say α), i.e.,:

$$\phi > \alpha \tag{17}$$

For a GCN model, training efficiency (η) depends on feature availability (f.a.), i.e., the more features available to the model for learning, the better the training of the model. Feature availability increases when the number of nodes reachable from a fixed set of nodes is high, since each node is associated with a feature vector X. Thus, feature availability again depends on node coverage or node reachability (κ). Based on all these discussions, a relationship can be established that looks like the following:

$$\eta \propto f.a. \propto \kappa \tag{18}$$

Considering equations (17) and (18) synchronously, the set E_2 will cover more neighborhood nodes, which means greater feature availability since each node is associated with a feature vector X. This increases the training efficiency of the model compared to selecting the training set based on E_2. To test this observation empirically, let us consider a small example according to Fig. 5. Consider $E_1 = \{(A, F), (B, C)\}$ as the edge set selected for training. For a fixed path length 2, the node coverage of the set is E_1:

$$\alpha = \{A, B, C, D, E, F, G, H\} \tag{19}$$

Now consider another edge set $E_2 = \{(A, B), (E, D)\}$ where the two edges with high betweenness centrality value are selected for training. For the same path length 2, t,he node coverage is the same for this training set:

$$\phi = \{A, B, C, D, E, F, G, H, I, J, K, L, M\} \tag{20}$$

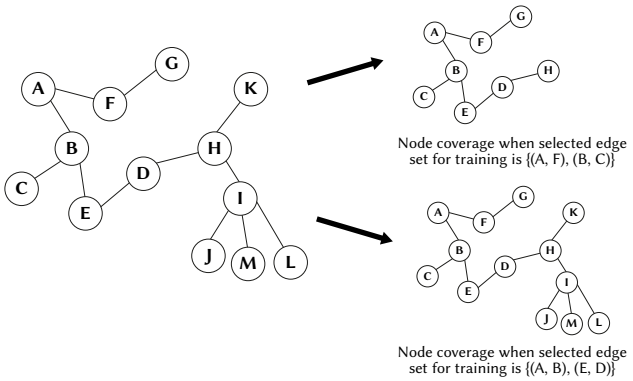


Fig. 5. Node coverage of graph $G(V,E)$ based on training edge selection.

Since E_2 has a larger number of nodes in its neighborhood, the availability of features will also be larger. And finally, it can be confirmed that the training efficiency of the model improves. Thus, it has been successfully analyzed that the selection of the training set based on the *edge betweenness centrality* improves the learning of the GCN-based training model for link prediction. On this basis, we can say that a mapping L' can be obtained which is approximately equal to L.

In the proposed method, the training is edge based, not node based. Hence, the criteria of edge set selection based on the betweenness centrality of edges makes sense. On the other hand, edge selection based on nodes having high degrees is not feasible. The reason for this is a high degree node has many edges associated with it. Each edge associated with the node will have equal weightage. Hence, all the edges incident on the high degree vertex will be selected for training. In such a situation, the model may miss out a significant portion of the network required for training since only edges which are incident to the high degree vertices will be selected. Clearly, this selection fails to capture the crucial structural properties of the of the network. Also, this degree-based selection will not allow the training set to capture diverse feature vectors which is essential for efficient training of the model.

On the other hand, consider the betweenness centrality-based approach for edge selection as discussed in *subsection* B of section III. This high betweenness edge centrality based selection will lead to generation of computation graphs with more number of nodes (in average) during training. Since, feature set aggregation is directly proportional to number of nodes in the underlying computational graph, a better training of the GCN model is guaranteed using proposed approach. This in turn enhances the prediction capability of the model.

Next, we need to ensure that the probability of randomly selecting the edge set E_2 is close to zero. Let us consider the total number of edges in the network as k, such that $k > 1$. The number of ways to choose a subset of length w (subset of w edges) is given as ${}^{k}C_{w}$. Our goal is to find the probability of choosing the subset E_2 from these ${}^{k}C_{w}$ subsets. Thus, let us consider an event Q as: *choosing the subset E_2 of the set E*, where E is the set of all edges of the graph such that $|E| = k$. The probability of this event will be:

$$P(Q) = 1/{}^{k}C_{w} \tag{21}$$

Let us assume that 45% of the edges are used for training. Thus, we have $w = (9/20)k$. Putting this value of w into the equation (21), we get,

$$P(Q) = 1/{}^{k}C_{9k/20} \tag{22}$$

In general, the number of edges for real network graphs is on the order of more than 10^4. Plugging the value of k as 10^4 into the expression, we get,

$$P(Q) \approx 0 \tag{23}$$

Finally, we can also successfully show that the probability of randomly choosing the edge set E_2 is close to zero. Thus, the section successfully verifies the two arguments: i) selecting the training set based on edge centrality improves the performance of the model. ii) the probability of randomly selecting an ordered set based on the centrality score is close to zero. In the next section, we detail the proposed method and its design along with the description of the dataset.

IV. Dataset and Model Description

In this section we discuss mainly about the datasets, the proposed model formulation, and aspects related to its implementation.

A. Dataset Description

To assess the performance of the proposed model, three famous state of the art datasets have been chosen: *CORA*, Citeseer and *PubMed*. The datasets have been summarized in Table II.

TABLE II. Dataset Description

Dataset	Nodes	Edges	Classes	Features	Type
Cora [25]	2,708	5,429	7	1,433	Citation Network
Citeseer [25]	3,312	4,732	6	3,703	Citation Network
PubMed [25]	19,717	44,338	3	500	Citation Network
Amazon [39]	13,752	491,722	10	767	Amazon Product Network
WikiCS [40]	11,701	216,123	10	300	Wikipedia Network

The first three datasets considered are essentially citation networks where *node* stands for *papers* and *edge* stands for the *citation links*. The CORA citation network consists of 2708 scientific publications classified into one of the following seven classes: neural networks, rule learning, reinforcement learning, probabilistic methods, theory, genetic algorithms, and case-based. For each node, there is a feature word vector of length 1433. Thus, the size of the feature matrix is 2708×1433. The Citeseer dataset consists of 3312 scientific papers classified into six classes: Agents, AI, DB, IR, ML, and HCI. The feature matrix has order 3312×3703. The PubMed citation network consists of 19,717 scientific publications with the following classification classes: *1, 2, 3* i.e., diabetes type-1, 2 and 3. The feature vector for each node consists of a TF/IDF vector with 500 unique words. The accuracy of the proposed model with GCN-based training was tested using these three benchmark datasets. The consistency of the results obtained with these networks highlights the effectiveness of the proposed solution. To prove the applicability of the proposed solution to other types of networks, two other graphical networks are considered. *Amazon Computer* [39] is a segment of the Amazon co-purchase graph, which is a network collected by crawling the Amazon website and contains product metadata and rating information about various products. The nodes in the graph represent items, while the edges indicate that two or more goods are usually purchased together. The goal is to assign items to the appropriate product categories by using product ratings as node attributes. *WikiCS* [40] is a novel dataset derived from Wikipedia to benchmark Graph Neural Networks. The dataset contains 11701 nodes corresponding to computer science articles, with edges based on hyperlinks, and 10 classes representing different branches of the field.

It is common for real-life applications with graphs to have limited training data because labels will often be sparse, despite having vast quantities of data. This is true for all the datasets considered in this manuscript. Hence, they are limited training datasets. In the context of link prediction, labels are edge labels (0 for not edge and 1 for an edge). And for training, a very small fraction of labels are known. For example, for Cora, labels of only 5429 edges are known (label 1) out of 3665278 possible edges. Labels of the remaining 3659849 edges are unknown. Hence, the Cora dataset is a limited training dataset. The same is true for other datasets too as shown in Table III.

TABLE III. Dataset With Actual V/s Possible Eddges in the Graphical Networks

Dataset	Nodes	Total possible edges	Total edges in actual graph
Cora	2708	3665278	5429
Citeseer	3312	5483016	4732
PubMed	19717	194370186	44338
Amazon	13752	94551876	491722
WikiCS	11701	68450850	216123

Following this, the next subsection explains the implementation design and operation of the proposed model.

B. Proposed Framework and Experimental Setup

To construct a Graph Convolution Network based training model architecture, *Stellar Graph library* [41] was used. In addition, the graph library *NetworkX* [42] is used to capture the structural information of the network. The input data set for the GCN model consists of an edge list and a *feature matrix* along with *labels*.

The relationship that exists between the data points (nodes) of the graph is represented by the *links* between them, defined by the *edge list*. To prepare the **test dataset**, an *Edge Splitter* () function from the *Stellar Graph library* was used. This function randomly takes some pairs of nodes from the original graph *G*. For each connected pair, the associated label is 1. Also, some unrelated pairs are randomly selected and these pairs are assigned the label 0. Thus, we obtain the final test set tuple t for which the label set L is defined with labels 0 and 1 for each unconnected and connected pair of nodes, respectively.

Let us now consider the training dataset. After removing the edges in the test set t from the graph G, the training dataset is selected from the residual graph G'. The training dataset contains the top k edges with high values of betweenness centrality computed using the *NetworkX* graph library (nx.edge_betweenness_centrality()). This part of the training dataset is denoted as $tr1$ with the corresponding label set as $trl1$ with all label values as 1. Furthermore, few edges are sampled using the *edgesplitter*() function to include some unconnected pairs. Let this part of the training dataset as $tr2$ with the label set $trl2$. Thus we have the final training dataset defined as:

$$tr = tr1 \cup tr2 \tag{24}$$

with training label set defined as:

$$trl = trl1 \cup trl2 \tag{25}$$

Fig. 6 explains the steps to generate a training dataset (55%) and a test dataset (upto 45%). The input graph dataset consists of edge list information along with node feature vectors. Note that each node has a feature vector associated with it. To create the test dataset, edge splitter function randomly pools the edges, marked as label '1', and an equal number of node pairs amongst which no direct edge exists, marked as label '0'. A similar procedure is adopted by the function to create the train dataset. However, in addition to the edges selected, top 'k' betweenness centrality metric-based edges are also appended in the training dataset. Finally, the train and test datasets are supplied to the GCN model. Since the connectivity of the graph must be ensured,

it is not possible to extract a very high percentage of edges from the graph for the creation of the test dataset. Therefore, the test dataset here is a combination of validation test dataset.

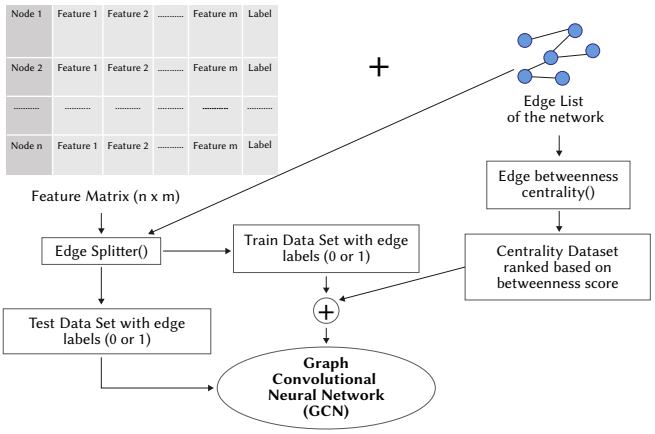


Fig. 6. Proposed Framework: Feeding train and test set to GCN.

Using the node feature matrix defined for each node for the nodes involved according to the training set selection, the model is fed with the input. The function *FullBatchnode Generator*() defines the neural network (NN) for the graphical network. The defined neural network has three layers: the input layer, the hidden layer, and the output layer. The number of hidden layers is best determined from the experimental simulations. However, in the case of GCNs, the number of hidden layers corresponds to the diameter of the graph. This refers to the number of neighbors that are a path length k away from the node under consideration. The value of k is generally kept very low because the vanishing gradient problem affects the performance of the model. Other hyperparameters of the model such as *kernelsize*, *learningrate*, *epochs*, *activationfunction*, etc. are chosen to minimize the error. The hidden layers have a Rectified Linear Unit (ReLU) activation function with a hidden layer size on the order of $4,096 \times 4,096$. However, the size of the *kernel* varies depending on the size of the network. Other parameters of the network such as *learning rate* is set to 0.0001 with *Adam Optimizer* and *Cross Entropy* as loss functions. The output layer of the model uses a *Softmax* function to predict the presence of an edge between a pair of edges over the test dataset. Fig. 7 explains the GCN-based training and classification process.

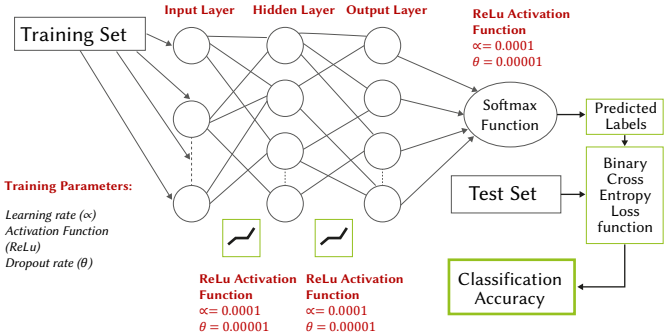


Fig. 7. GCN Based Training of the model.

Fig. 8 sums up the entire process in a block diagram. The algorithmic steps in training of Bet-GCN model are as shown in algorithm *Bet-GCN*. The input to the model is an input graph dataset $G(V, E)$ where V represents set of vertices and E represent set of edges. Each node in the graph has feature vector associated with it. Let A be the adjacency matrix for the graph and X be the feature matrix (as mentioned in equation 12). The algorithm will yield a trained model m which can predict whether an edge exists (edge label 0) or not (edge label 1) between two given pair of nodes (binary classification problem). In **step 1**, *edge splitter* function randomly pools a set of edges from graph G to prepare training dataset (say Tr). The training set consist of edges which exist in the graph labelled as 1 and edges which do not exist in the graph labelled as 0. In **step 2**, from the remaining graph (say G'), *edge splitter* function constructs the test dataset (say Te) in a similar manner. **Step 3** and **Step 4** identifies the top k edges in order of *edge betweenness centrality*. The top k edges identified in *step 4* are added in **step 5** to Tr to generate the final training dataset. In **step 6**, the GCN model is fed with Tr, G and the model hyperparameters like *learning rate, layer size, ReLu activation function*. The input layer is fed with an aggregation function defined as $A.X$. The hidden layer further performs feature aggregation using a layer size $4,096 \times 4,096$ at a learning rate 0.0001. The ReLu activation function is applied to obtain the convoluted vector (neighborhood aggregation) matrix at each layer. At each layer gradients are determined and based upon the error function gradients, using backpropagation algorithm weights are adjusted. This whole process iterates till the error gradient functions at each layer evaluates out to be zero. In this condition, we obtained a finalized weight vector matrix at output layer and the trained model m. Finally, in **step 7**, the trained model is tested over Te using *SoftMax()* classification function to generate the classification report.

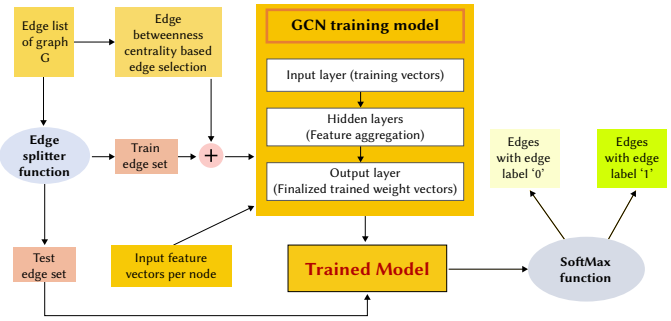


Fig. 8. Pictorial block diagram for Bet-GCN model.

V. Results and Analysis

This section mainly focuses on the experimental results and performance of the proposed Bet-GCN model. It highlights the significant results of the model in three benchmark datasets, namely Cora, Citeseer and PubMed, and the comparative analysis with the respective state-of-the-art methods. The results of our proposed model Bet-GCN (Edge betweenness centrality with Graph convolutional networks) are summarized in Table IV. The performance of the model improves considering that the model performs well on a large test data set. All of the state-of-the-art methods discussed work over 5 10% test data. The Bet-GCN based results are analyzed over upto 45% test data with at least 30% unseen node pairs in the test dataset.

TABLE IV. Citation Networks Accuracy

Method	Cora	Citeseer	PubMed	Test Dataset
VGAE [25]	0.920	0.914	0.965	-
MTGAE [27]	0.946	0.949	0.944	5-10%
GLP [32]	0.9455	0.8612	-	5-55%
GCN [33]	0.9050	0.8701	0.9694	-
GAT [33]	0.8979	0.8731	0.9436	-
EdgeConv [33]	0.8528	0.8294	0.8665	-
EdgeConvNorm [33]	0.9178	0.8754	0.8991	-
Bet-GCN(proposed)	0.9508	0.9507	0.953	upto 45%

Algorithm 1. Bet-GCN

Require: An input graph dataset $G(V, E)$ where V represents set of vertices and E represent set of edges

Output: The trained model, m

Step 1. $Tr \leftarrow edgesplitter(G)$

▷ Generating training set Tr by random selection of edges from graph G

Step 2. $Te \leftarrow edgesplitter(G)$

▷ Generating test set Te by random selection of edges from graph G

Step 3. $e \leftarrow edge\_betweenness\_centrality(G)$

▷ Evaluating edge betweenness centrality of edges of graph G

Step 4. $e' \leftarrow sorted(e[1{:}k])$

▷ Selecting top k edges based on edge betweenness centrality of edges of graph G

Step 5. $Tr \leftarrow Tr \cup e'$

▷ Adding edges form step 4 to Tr

Step 6. $m \leftarrow GCN(G, Tr, learningrate, layersize, ReLu)$

▷ Obtaining the trained GCN model m

Step 6. $classication\_report \leftarrow SoftMax(m, Te)$

▷ Testing the model over test set and generating classification report

As summarized in Table IV, most models consider methods such as random walks (where only local node similarity is used) or maximum likelihood estimation methods for link prediction. It can be observed that none of these methods materialise the node features, the structure of the underlying network, or the importance of the edges completely. In comparison, GCN, which considers the structure of the dataset as a graph, significantly improves link prediction performance. Traditional Graph Convolutional Networks (GCN) directly convolve the structure of the connected graph as a filter to perform neighbourhood mixing. Graph Attention Networks (GAT), on the other hand, apply a shared linear transformation to each node, followed by a computation of attention coefficients using a joint attention mechanism. The performance of link prediction with these two models is impressive and promising. A more recent state of the art, the Variational Graph Auto-Encoder (VGAE), uses a graph convolutional network as an encoder that maps the node features into a latent representation, followed by a decoder that generates conditional probabilities of the adjacency matrix [25]. While Multi-Task Graph Autoencoders (MTGAE [27]) learns a joint representation of latent embeddings from a local graph and explicit node features. These two methods are significantly better than the traditional GCN model due to the inclusion of autoencoders. In addition, GLP [32], a gravitational link based unsupervised approach is used. Here, the main idea is to decompose the graph into a local structure (by extracting subgroups) and a global structure (by detecting communities). The method showed promising results on large complex networks, but is highly dependent on the network structure. Two recent link prediction methods based on Graph Convolution Learning were also proposed: *EdgeConv* and *EdgeConvNorm* [33]. The methods performed well on the three networks, as the over-smoothing of Edge Convnorm helps to better learn link prediction based on the node and its neighborhood representation.

Our proposed model (Bet-GCN) is a modification of the traditional GCN model, as it uses edges based on their betweenness centrality in the graph along with node features. Our proposed prediction model achieves an accuracy of 95.08% in Cora, 95.07% in Citeseer, and 95.32%in PubMed. These results are competitive with the current state-of-the-art models, which can be observed in Table IV.

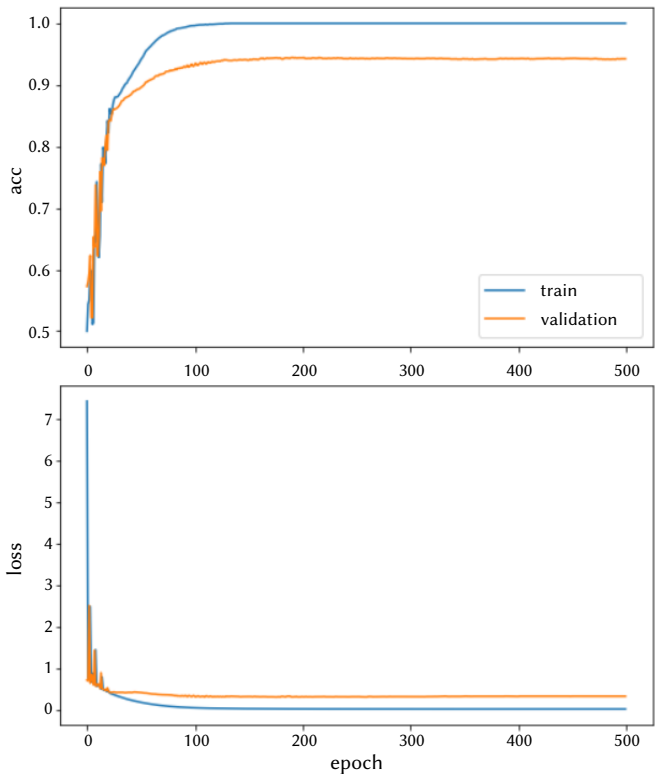


Fig. 9. CORA: Training and Loss curve.

The model extrapolates the structure of the underlying graph for sampling positive edges when training the model for prediction. BET-GCN architectural hyperparameters were fine-tuned for the Cora, Citeseer, and PubMed networks. A 0.70 and 0.35 fraction of the original network is randomly sampled for positive and negative edges as training and test edges, respectively. The positively sampled training edges are replaced with the edges sorted based on the edge betweenness centrality score. A two-layer GCN model is used, where 4, 096 is the dimension of the node features in each hidden layer. The Rectified Linear Unit (ReLU) activation function is used. For the final link classification, a pair of node embeddings from the GCN model is used and the binary operator inner product (ip) is applied. This produces the corresponding link embedding, which is passed through a dense layer. A learning rate of 0.0001 for Adam Optimizer is used to train the model. Our model is trained with 500 epochs. These hyperparameter settings are the same for Cora and Citeseer citation networks. The PubMed dataset consists of 10x more edges and therefore has different hyperparameter values. The training accuracy and loss curves of the model for the three datasets are shown in Fig. 9, Fig. 10, and Fig. 11, respectively. Based on the obtained results, it can be confirmed that the proposed method performs best for the three collaboration networks.

Area Under the Curve (AUC) curves of the model obtained for the three datasets are shown in Fig. 12, Fig. 13 and Fig. 16. The AUC curves show the ability of the classifier to distinguish correctly between positive and negative classes. The high AUC value for all three datasets CORA (94.02%), Citeseer (94.24%), and PubMed (97.96%) indicates the consistency of the model in terms of performance.

The hyperparameters' settings depend upon the size and structure of the network for training GCN models. The basic parameter settings' have been considered based upon Thomas N Kipf and Max Welling [1] paper. The parameters that are varied are layer size, learning rate and iterations due to varied network structures and sizes. In general, ReLu activation function has been used for 4, 096 × 4, 096 layer size

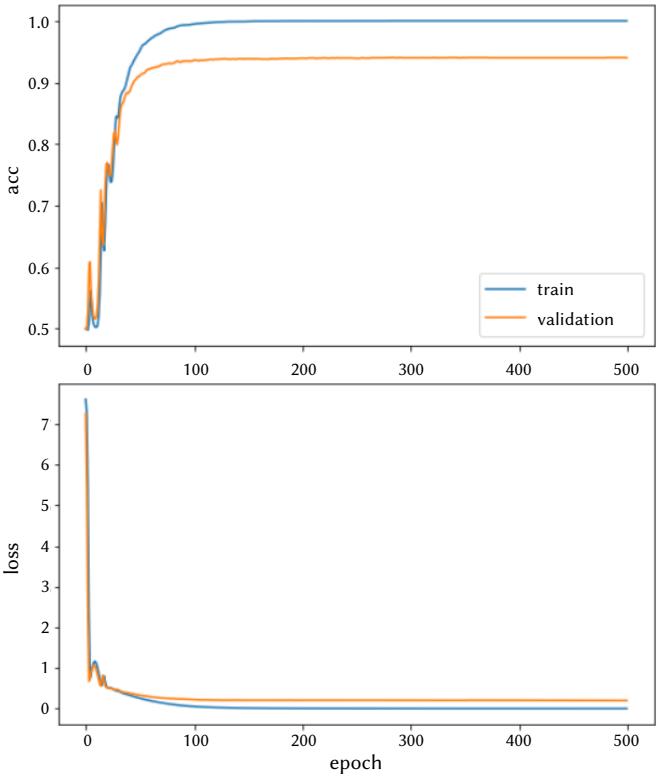


Fig. 10. Citeseer: Training and Loss curve.

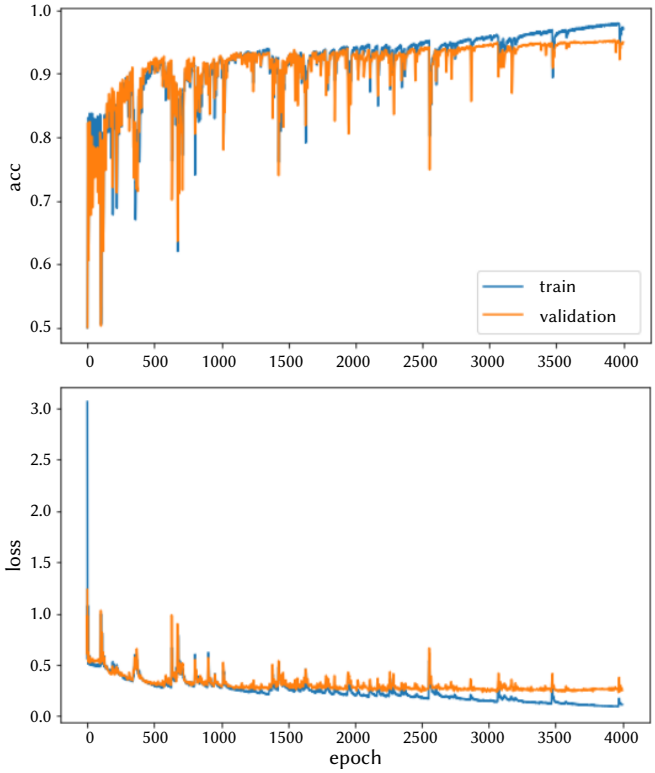


Fig. 11. PubMed: Training and Loss curve.

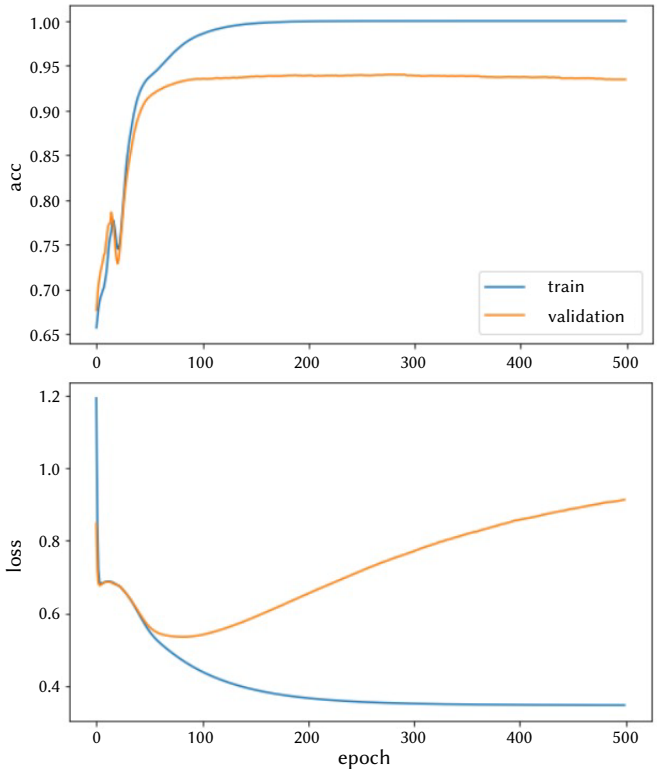


Fig. 12. AUC Curve for CORA network with accuracy (94.02%).

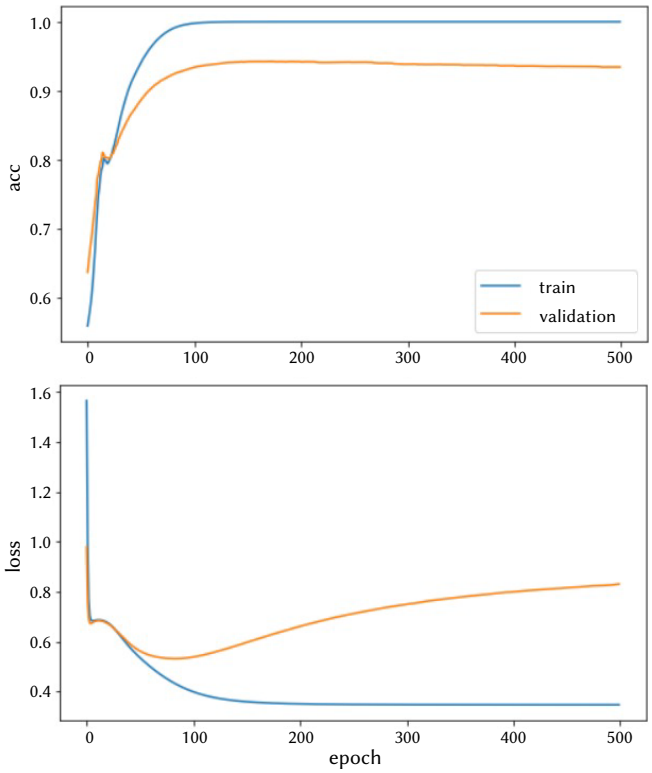


Fig. 13. AUC Curve for Citeseer network with accuracy (94.24%).

of hidden layer which yield the best results. The number of iterations is identified based upon the training accuracy curve trajectory and, hence, the number of iterations is different for each network. The iterations' convergence happens when the training accuracy starts to dip for several continuous iterations. So, further increasing the number of iterations will not yield good results and may tend the model to overfit. Similarly, the best results are obtained for a learning rate of 0.0001 for the three benchmark datasets into consideration (CORA, Citeseer and PubMed). Fig. 14 shows that further reducing the learning rate causes a drop in the performance efficiency of the model for CORA dataset. Fig. 15 presents the trend analysis of the performance of the model with number of epochs. At 500 epochs, keeping the learning rate fixed at 0.0001 and hidden layer size of 4096 4096, the performance attained by the model is optimum. Further increasing the number of epochs for model training is not helping the cause and the performance tends to deteriorate as the model starts *overfiting*. A similar analogy can be drawn for the size of hidden layer. Further, a similar kind of analysis can also be obtained for the two other kind of networks (Citeseer and PubMed). Thus, it can be inferred that learning rate of 0.0001 and hidden layer size of $4,096 \times 4,096$ is suitable for networks of different variety and structural formation in order to have an efficient training through Bet-GCN model. Number of epochs to attain the optimum accuracy may differ depending on the size of the network. However, all of these parameter settings are network dependent and vary slightly depending on the nature of the task.

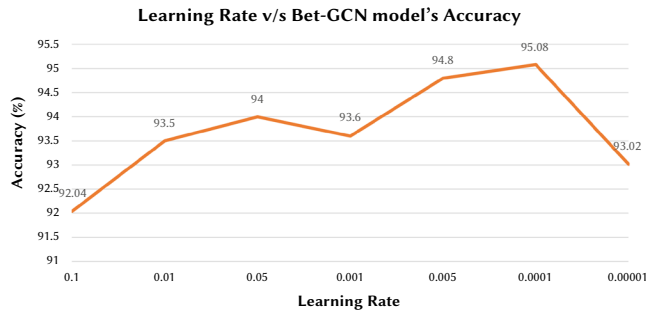


Fig. 14. CORA: Learning Rate v/s Bet-GCN accuracy curve.

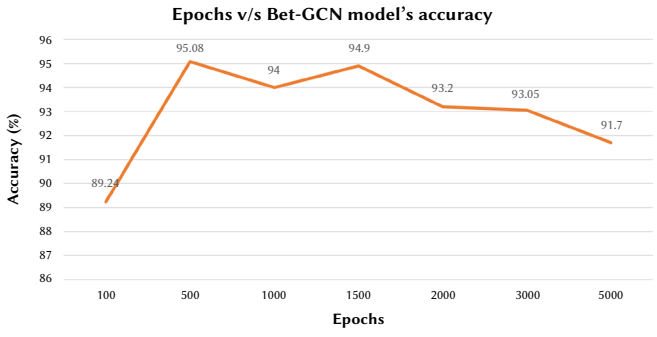


Fig. 15. CORA: Epochs v/s Bet-GCN accuracy curve.

VGAE and GAE [25] uses a Gaussian prior distribution over the input features to learn embeddings. However, this has not proven to be a very good choice. MTGAE [27] gives impressive results for link prediction, but the accuracy of the method decreases when a larger number of edges are removed from the graph. This is because only the contribution of the available edges is considered. GLP [32] involves a lot of preprocessing, such as community identification, followed by extraction of optimized subgraphs. The link prediction task is then performed over these distributed subgraphs. The method is not suitable for networks with large diameters. The other link prediction strategies mentioned in the work of Gu et al. [33] are GCN-based methods where the selection of the training set is random. Our proposed method Bet-

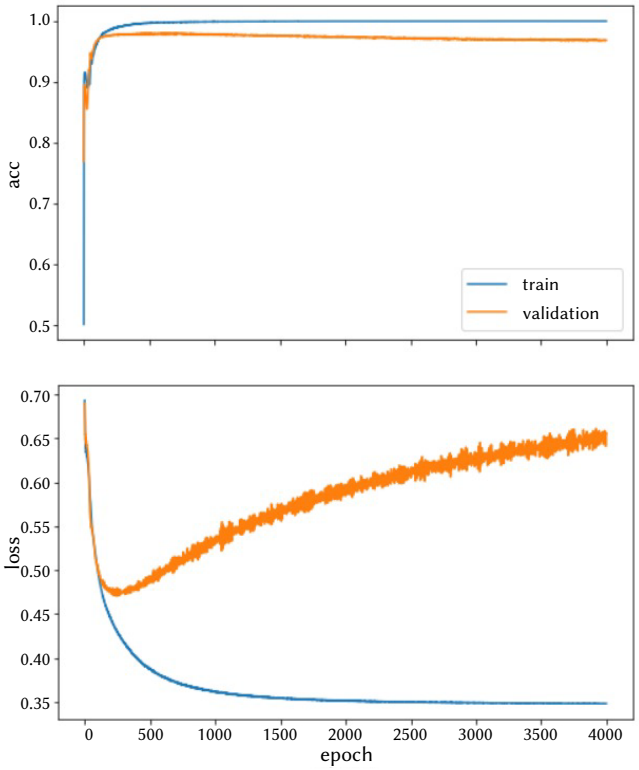


Fig. 16. AUC Curve for PubMed network with accuracy (97.96%).

GCN is also a variation of GCN technique where the training set is selected based upon the betweenness centrality score. This helps in capturing more neighborhood contribution for the model's training. As a result, there is more neighborhood aggregation in the computational graphs. This will help the model to leverage the feature-based learning and generate more accurate embeddings. Traditional GCN approaches use random selection and, hence, they are not able to capture features which are betweenness centrality based. It is due to this reason that the method performs well in comparison to the other state of art methods.

In addition, the Bet-GCN model was also tested on two different types of networks (since all three networks mentioned above were citation networks): Amazon Product [39] and WikiCS [40]. The Amazon Product network was collected by crawling the Amazon website and contains product metadata and review information for 548552 different products (Books, music CDs, DVDs, and VHS video tapes). WikiCS [40] is a web graph of Wikipedia hyperlinks collected in September 2011. Bet-GCN link prediction model for both datasets perform equally well as for the citation networks. Table V lists the accuracy and respective F1-score values for the network.

TABLE V. Accuracy and F1-score Values for Amazon Product and WikiCS Networks

Network	Accuracy	F1-Score	Test Dataset
Amazon Product	0.879	0.8801	upto 45%
WikiCS	0.9113	0.90	upto 45%

Fig. 17 and Fig. 18 show the training accuracy curves for both networks using the Bet-GCN model. The results for the network indicate that the approach is scalable with network size and applicable to graphical networks of different domains. The results for these two networks were evaluated with the same parameter settings used for CORA, Citeseer, and PubMed, except for the layer size. Fig. 19 refers to the confusion matrix for all the five graphical networks, which shows the prediction capability of the proposed model Bet-GCN. Given the

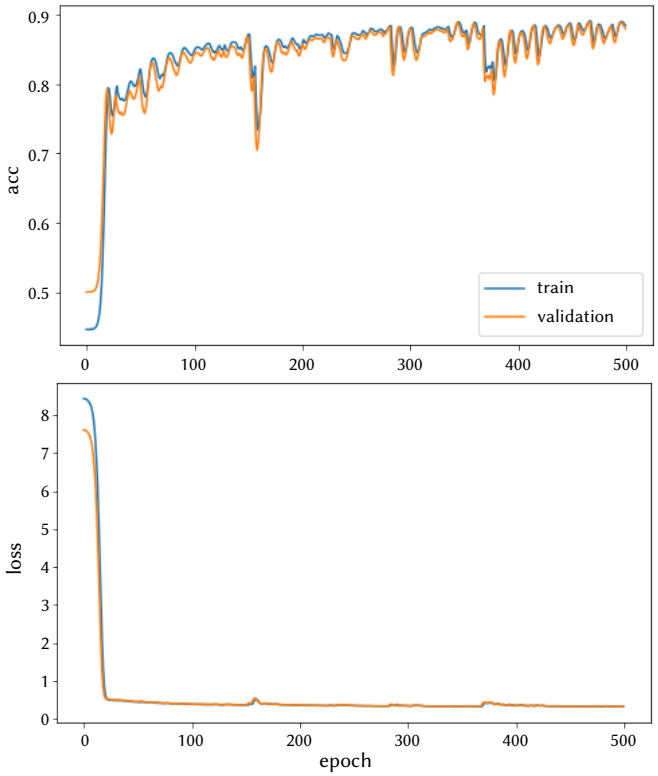


Fig. 17. Training and Loss Accuracy Curves for Amazon Product Network.

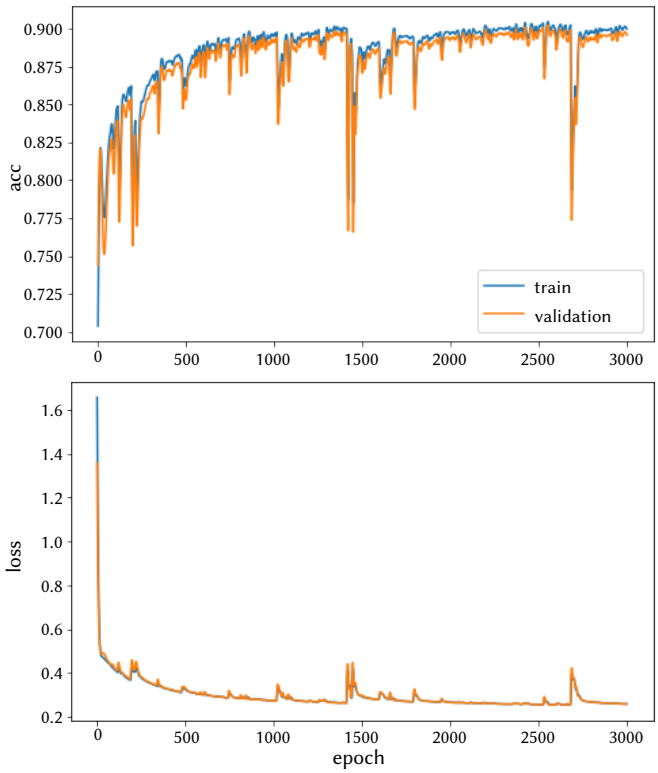


Fig. 18. Training and Loss Accuracy Curves for WikiCS Weblink Network.

CORA		
	True Positive	**True Negative**
Predicted Positive	2609	105
Predited Negative	163	2551

Citeseer		
	True Positive	**True Negative**
Predicted Positive	2233	124
Predited Negative	109	2248

PubMed		
	True Positive	**True Negative**
Predicted Positive	20356	1813
Predited Negative	344	23625

Amazon		
	True Positive	**True Negative**
Predicted Positive	277349	17684
Predited Negative	47810	247223

WikiCS		
	True Positive	**True Negative**
Predicted Positive	252262	6773
Predited Negative	39511	219524

Fig. 19. Confusion Matrix for the graphical networks.

large number of edges in the networks, the convolutional layer size used for the hidden layer is 512×512. As one increase the number of layers, the number of parameters that can be trained also increases, and so does the execution time. This may improve the performance of the model by a small percentage, but the tradeoff is very high.

The Betweenness centrality range for the networks in consideration is shown in Table VI. The betweenness centrality measure denotes that how often a particular edge (say 'x') gets visited among the total paths in the network across any two nodes. This value, thus, will be in the range 0 to 1. Also, availability of such paths passing through edge 'x' in comparison to the total number of paths between any two nodes in the network will be very low. Hence, the betweenness centrality value evaluated for each edge as per explanation in subsection B of section 3, this value will be a very small number. However, the values can be normalized to any range/interval, but it will not affect the result as the magnitude of the betweenness centrality value increases for each edge by same factor.

TABLE VI. Citation Networks Accuracy

Network	Minimum	Maximum
CORA	0	0.0359
Citeseer	0	0.0462
PubMed	0	0.0134
Amazon	0	0.0055
WikiCS	0	0.0165

Bet-GCN performance over Facebook-Pages-Food Dataset: To further demonstrate the generalizability of Bet-GCN model in the perspective of social links of a social media platform, the model has been tested over Facebook-Pages-Food [43] network dataset.

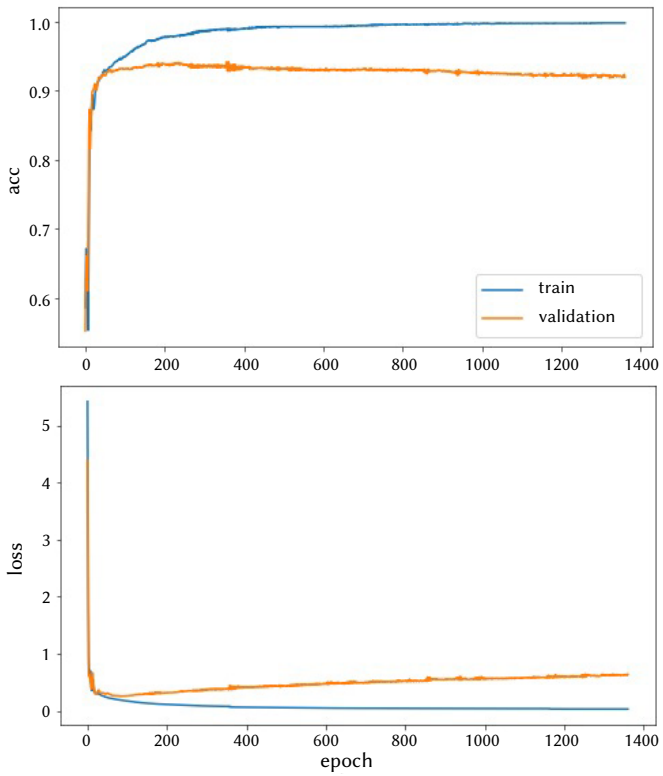


Fig. 20. Training and Loss Accuracy Curves for Facebook FoodWeb Pages.

Most social media platforms, including Facebook, can be structured as graphs. The registered users are interconnected in a universe of networks. The objective of link prediction is to identify pairs of nodes that will either form a link or not in the future. Here, we worked on a graph dataset in which the nodes are Facebook pages of popular food joints and well-renowned chefs from across the globe and if any two pages (nodes) like each other, then there is an edge (link) between them. For calculating node embeddings we have applied node2vec [44] on the graph. Then, Bet-GCN model is trained on 2259 edges and tested for 2522 edges. On training for 1500 epochs we get f1-score of 0.9442, which is a major improvement when compared to f1-score of 0.7817 for logistic regression in [43]. The model hyperparameter settings have been kept same as for the above models. The training and loss accuracy curves have been shown in Fig. 20 represents the training and loss accuracy curve for the same. This further demonstrates the prediction capability of Bet-GCN model with high accuracy on a different variety of real world graphical networks.

Reason of selection of GCN based methods over classical Machine learning and neural network techniques: The problem with social media data is the availability of the feature for every node in the graphical network. So, using correlational analysis and belief propagation techniques are not suitable as these techniques require features to be compared to calculate the similarity between the nodes and their behavior. In the absence of feature-based information, graphical structure information needs to be employed. The proposed high betweenness edge centrality based selection will lead to generation of computation graphs with more number of nodes (in average) during training. Since, feature set aggregation is directly proportional to number of nodes in the underlying computational graph, a better training of the GCN model is guaranteed using proposed approach. This in turn enhances the prediction capability of the model. In the state of the art literature there are many evidences where GCN based methods are outperforming traditional machine learning methods. Jiang et al. [45] have shown that the performance of GCN model to predict synergistic drug combinations in particular cancer cell lines in comparison to classical machine learning algorithms like Support Vector Machine, Radial Basis Function, Deep Neural Networks etc. is much better. Tayal et al. [46] have shown that the performance of GCN based techniques for text classification task is superior in comparison to other ML and DL techniques like TF-IDF with Logistic regression, CNN, Char CNN etc. The performance improvement is of approximately 2% with reduced dataset for training. Cao et al. [47] have shown in their comprehensive review article that how GCNs surpassed the performance of various CNN models. From these discussions, we can conclude that GCNs have high prediction capability due to added power of network structural information. Moreover, they can work well with limited feature availability and information about many data points in the network.

Lastly, lets have a look on the computational complexity of the model. The time complexity for calculating betweenness centrality of edges in the network is given as O(|V|.|E|) [48], where, |E| are the number of edges and |V| are the number of nodes or vertices in the network. Further, the time complexity of GCN based training is given as $O(L.|V|.|F^2|)$ [49]. Here, 'L' represents the number of layers of the neural network, 'V' represents number of vertices and 'F' represent feature vector corresponding to each node of the graphical network. Then, overall complexity for the algorithm can be given as:

$$T = O(|V|.|E|) + O(L.|V|.|F^2|) \tag{26}$$

For real world networks, $|E| >> |V|$, but $|E| < |V|^2$. Therefore,

$$O(|V|^2) << O(|V| * |E|) < O(|V|^3) \tag{27}$$

Also, $L<<|V|$ and is a constant value, so it can be omitted. Since, $F<|V|$, this means that $F^2<<V^2$. Therefore, from equations (26) and (27), we have,

$$O(|V|.L.|F^2|) \approx O(|V| * |F^2|) < O(|V|^3) \tag{28}$$

Hence, the overall time complexity of Bet-GCN model evaluates out to be of cubic order as a function of number of vertices. This means that solution is attainable in polynomial time. Moreover, the training process takes into consideration only 50 - 55% nodes into consideration. Given the advancements in computational power of modern day computers having GPU processors, the task can be accelerated significantly despite of cubic order time complexity of the process. Also, it is to be noted that even in case of traditional GCN the time complexity will be upper bounded by $O(L.|V|.F^2) \approx O(|V|^3)$. So, betweenness centrality based calculation do not hurt the overall time complexity of the task.

VI. Conclusions

The paper presents a variation of the traditional Graph Convolutional Network approach for the task of link prediction. An approach based on betweenness centrality was chosen for the selection of the edges to be trained. Thus, the top-k edges are selected to create the training set of edges that have a high value for edge centrality. This idea contributes to a significant improvement in model accuracy. The proposed model outperforms other state of the art based deep learning methods as the results are promising even with a high percentage of test dataset. The accuracy of the model was tested for up to 45% test dataset, while most state of the art models have reported accuracy over 5 - 10% test dataset. The reason for this improvement is the increased neighborhood span, which helps in generating rich node embeddings in GCN-based training for the model. The effectiveness of the results in the three datasets: CORA Citeseer and PubMed, was confirmed by the AUC curves. Moreover, the model has achieved impressive results on Amazon Product, WikiCS and Facebook Food

Web Page networks, which are very large and belong to a different category than the previous three, showing that the method is generic and can be applied to graphical networks of different domains. In summary, the key contributions of the manuscript are:

- Proposing an efficient GCN-based link prediction technique where the training set is selected based on *edge betweenness centrality*.
- Mathematical and experimental justifications of the improvement in GCN based training for link prediction.
- Detailed comparison of the results with the current state of the art methods for link prediction by performing experimental simulations over 6 different networks.

In future, the model can be tested with a larger number of complex network datasets to further verify the robustness of the proposed model. Moreover, the same model can be tested to determine the performance improvement on other tasks such as node classification, graph classification etc.

References

[1] M. Sun, J. Chen, Y. Tian, Y. Yan, "The impact of online reviews in the presence of customer returns," *International Journal of Production Economics*, vol. 232, p. 107929, 2021, doi: 10.1016/j.ijpe.2020.107929.

[2] M. S. Ullal, C. Spulbar, I. T. Hawaldar, V. Popescu, R. Birau, "The impact of online reviews on e-commerce sales in india: A case study," *Economic Research- Ekonomska Istraživanja*, vol. 34, no. 1, pp. 2408–2422, 2021, doi: 10.1080/1331677X.2020.1865179.

[3] M. Caro-Martínez, G. Jiménez-Díaz, J. A. Recio- García, "Local model-agnostic explanations for black- box recommender systems using interaction graphs and link prediction techniques," *International Journal of Interactive Multimedia and Artificial Intelligence*, pp. 1–11, 2021, doi: 10.9781/ijimai.2021.12.001.

[4] D. Medel, C. González-González, S. V. Aciar, "Social relations and methods in recommender systems: A systematic review," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 4, pp. 7– 17, 2022, doi: 10.9781/ijimai.2021.12.004.

[5] N. N. Daud, S. H. Ab Hamid, M. Saadoon, F. Sahran, N. B. Anuar, "Applications of link prediction in social networks: A review," *Journal of Network and Computer Applications*, vol. 166, p. 102716, 2020, doi: 10.1016/j.jnca.2020.102716.

[6] S. Sledzieski, R. Singh, L. Cowen, B. Berger, "Sequence- based prediction of protein-protein interactions: a structure-aware interpretable deep learning model," *bioRxiv*, 2021, doi: 10.1016/j.cels.2021.08.010.

[7] M. Lim, A. Abdullah, N. Jhanjhi, M. K. Khan, M. Supramaniam, "Link prediction in time-evolving criminal network with deep reinforcement learning technique," *IEEE Access*, vol. 7, pp. 184797–184807, 2019, doi: 10.1109/ACCESS.2019.2958873.

[8] H. Huang, J. Tang, L. Liu, J. Luo, X. Fu, "Triadic closure pattern analysis and prediction in social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 12, pp. 3374–3389, 2015, doi: 10.1109/TKDE.2015.2453956.

[9] M. S. Granovetter, "The strength of weak ties," *American journal of sociology*, vol. 78, no. 6, pp. 1360– 1380, 1973.

[10] Y. Bi, W. Wu, L. Wang, "Community expansion in social network," in *International Conference on Database Systems for Advanced Applications*, 2013, pp. 41–55, Springer.

[11] E. Abbe, A. S. Bandeira, G. Hall, "Exact recovery in the stochastic block model," *IEEE Transactions on information theory*, vol. 62, no. 1, pp. 471–487, 2015, doi: 10.1109/TIT.2015.2490670.

[12] C. Matias, V. Miele, "Statistical clustering of temporal networks through a dynamic stochastic block model," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 79, no. 4, pp. 1119–1141, 2017.

[13] A. K. Gupta, N. Sardana, "Significance of clustering coefficient over jaccard index," in *The International Conference on Contemporary Computing*, 2015, pp. 463– 466, IEEE.

[14] D. Liben-Nowell, J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007, doi: 10.1145/956863.956972.

[15] S. Cohen, B. Kimelfeld, G. Koutrika, "A survey on proximity measures for social networks," in *Search computing*, 2012, pp. 191–206, Springer.

[16] S. Zhang, H. Tong, J. Xu, R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019, doi: 10.1186/s40649-019-0069-y.

[17] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020, doi: 10.1109/TNNLS.2020.2978386.

[18] T. Derr, Y. Ma, W. Fan, X. Liu, C. Aggarwal, J. Tang, "Epidemic graph convolutional network," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 160–168.

[19] H. Kashima, T. Kato, Y. Yamanishi, M. Sugiyama, K. Tsuda, "Link propagation: A fast semi-supervised learning algorithm for link prediction," in *Proceedings of the 2009 SIAM international conference on data mining*, 2009, pp. 1100–1111, SIAM.

[20] R. Raymond, H. Kashima, "Fast and scalable algorithms for semi-supervised link prediction on static and dynamic graphs," in *Joint european conference on machine learning and knowledge discovery in databases*, 2010, pp. 131–147, Springer.

[21] A. K. Menon, C. Elkan, "Link prediction via matrix factorization," in *Joint european conference on machine learning and knowledge discovery in databases*, 2011, pp. 437–452, Springer.

[22] S. Gao, L. Denoyer, P. Gallinari, "Temporal link prediction by integrating content and structure information," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 1169–1174.

[23] Z. Zeng, K.-J. Chen, S. Zhang, H. Zhang, "A link prediction approach using semi-supervised learning in dynamic networks," in *The International Conference on Advanced Computational Intelligence*, 2013, pp. 276–280, IEEE.

[24] L. Berton, J. Valverde-Rebaza, A. de Andrade Lopes, "Link prediction in graph construction for supervised and semi-supervised learning," in *The International Joint Conference on Neural Networks*, 2015, pp. 1–8, IEEE.

[25] T. N. Kipf, M. Welling, "Variational graph auto- encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[26] H. Yang, S. Pan, P. Zhang, L. Chen, D. Lian, C. Zhang, "Binarized attributed network embedding," in *IEEE International Conference on Data Mining*, 2018, pp. 1476– 1481, IEEE.

[27] P. V. Tran, "Multi-task graph autoencoders," *arXiv preprint arXiv:1811.02798*, 2018.

[28] R. Hisano, "Semi-supervised graph embedding approach to dynamic link prediction," in *International Workshop on Complex Networks*, 2018, pp. 109–121, Springer.

[29] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," *arXiv preprint arXiv:1802.04407*, 2018.

[30] X. Di, P. Yu, R. Bu, M. Sun, "Mutual information maximization in graph neural networks," in *The International Joint Conference on Neural Networks*, 2020, pp. 1–7, IEEE.

[31] T. Zhang, K. Zhang, X. Li, L. Lv, Q. Sun, "Semi- supervised link prediction based on non-negative matrix factorization for temporal networks," *Chaos, Solitons & Fractals*, vol. 145, p. 110769, 2021, doi: 10.1016/j.chaos.2021.110769.

[32] E. Bastami, A. Mahabadi, E. Taghizadeh, "A gravitation-based link prediction approach in social networks," *Swarm and evolutionary computation*, vol. 44, pp. 176–186, 2019, doi: 10.1016/j.swevo.2018.03.001.

[33] W. Gu, F. Gao, R. Li, J. Zhang, "Learning universal network representation via link prediction by graph convolutional neural network," *Journal of Social Computing*, vol. 2, no. 1, pp. 43–51, 2021, doi: 10.23919/JSC.2021.0001.

[34] M. Shabaz, U. Garg, "Predicting future diseases based on existing health status using link prediction," *World Journal of Engineering*, 2021, doi: 10.1108/WJE-10-2020- 0533.

[35] M. Wang, L. Qiu, X. Wang, "A survey on knowledge graph embeddings for link prediction," *Symmetry*, vol. 13, no. 3, p. 485, 2021, doi: 10.3390/sym13030485.

[36] F. J. Roethlisberger, W. J. Dickson, *Management and the worker*, vol. 5. Psychology press, 2003.

[37] R. Saxena, M. Jadeja, "Network centrality measures: role and importance

in social networks," in *Principles of Social Networking*, 2022, pp. 29–54, Springer.

[38] U. Brandes, S. P. Borgatti, L. C. Freeman, "Maintaining the duality of closeness and betweenness centrality," *Social Networks*, vol. 44, pp. 153–159, 2016, doi: 10.1016/j.socnet.2015.08.003.

[39] O. Shchur, M. Mumme, A. Bojchevski, S. Günnemann, "Pitfalls of graph neural network evaluation," *arXiv preprint arXiv:1811.05868*, 2018.

[40] P. Mernyei, C. Cangea, "Wiki-cs: A wikipedia-based benchmark for graph neural networks," *arXiv preprint arXiv:2007.02901*, 2020.

[41] Z. Zhang, X. Wang, W. Zhu, "Automated machine learning on graphs: A survey," *arXiv preprint arXiv:2103.00742*, 2021.

[42] M. Kaur, H. Kaur, "Implementation of enhanced graph layout algorithm for visualizing social network data using networkx library," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 3, 2017, doi: 10.26483/ijarcs.v8i3.2998.

[43] R. Rossi, N. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[44] A. Grover, J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[45] P. Jiang, S. Huang, Z. Fu, Z. Sun, T. M. Lakowski, P. Hu, "Deep graph embedding for prioritizing synergistic anticancer drug combinations," *Computational and structural biotechnology journal*, vol. 18, pp. 427–438, 2020, doi: 10.1016/j.csbj.2020.02.006.

[46] K. Tayal, R. Nikhil, S. Agarwal, K. Subbian, "Short text classification using graph convolutional network," in *NIPS workshop on Graph Representation Learning*, 2019.

[47] P. Cao, Z. Zhu, Z. Wang, Y. Zhu, Q. Niu, "Applications of graph convolutional networks in computer vision," *Neural Computing and Applications*, pp. 1–19, 2022, doi: 10.1007/s00521-022-07368-1.

[48] N. Kourtellis, G. D. F. Morales, F. Bonchi, "Scalable online betweenness centrality in evolving graphs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2494–2506, 2015, doi: 10.1109/ICDE.2016.7498421.

[49] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, C.-J. Hsieh, "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 257–266.

Rahul Saxena

He is currently working as an Assistant Professor in Department of Information technology, Manipal University Jaipur, since 2015 and pursuing PhD from Malaviya National Institute of Technology, Jaipur since 2019. He completed his Masters from Manipal University Jaipur in the year 2015. He has been awarded with Gold Medal for Excellence in Education in Masters. He completed his B.E. from Birla Institute of Technology, Mesra in year 2013. His areas of research and interest includes Social Networks Analysis, Machine Learning, Graph Algorithms, Parallel processing etc. He has several conference, journal articles and book chapters published in Springer, IEEE etc. in the related domains of research.

Spandan Pankaj Patil

She received B.tech in Electrical engineering degree in 2022 from NIT Jaipur. She is currently working as a full-time Data Scientist at Micron Technology. Her research interests include Graph neural networks, social network analysis, machine learning, and computer vision.

Pranshu Vyas

He received his B. Tech. in computer science and engineering from MNIT Jaipur. He is currently working as a software developer at D. E. Shaw India Private Limited. His fields of interest are Data structure, Algorithms, and Machine learning, with a special focus on Neural network approaches for graphical data such as GCN.

Atul Kumar Verma

He received his B.Tech degree in computer science and engineering from VBS Purvanchal University, Jaunpur, UP, India in 2009 and M.Tech degree in computer science and engineering from Dr. A.P.J. Abdul Kalam Technical University UP, India in 2016. He is currently pursuing PhD at the Department of computer science and engineering, Malaviya National Institute of Technology, Jaipur India. His areas of interest are Social Networks Analysis, Machine Learning, Deep Learning and Graph Algorithms.

Mahipal Jadeja

He received his Ph.D. degree from Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT) in the field of Theoretical Computer Science. He currently works at Malaviya National Institute of Technology (MNIT Jaipur) as an assistant professor. His research interests include Theoretical Computer Science, Social Network Analysis, and Machine Learning on Graphs (Graph Neural Networks). He has published several journal articles, book chapters, and a reference book (Springer) in these domains. His research work is presented at reputed international conferences including GSB-SIGIR 2015 (Chile), WAAC 2016 (Japan), and SCAI-ICTIR 2017 (Netherlands).

Vikrant Bhateja

Vikrant Bhateja is associate professor in Department of Electronics Engineering Faculty of Engineering and Technology, Veer Bahadur Singh Purvanchal University, Jaunpur, Uttar Pradesh, India. He holds a doctorate in ECE (Bio-Medical Imaging) with a total academic teaching experience of 19+ years with around 190 publications in reputed international conferences, journals and online book chapter contributions; out of which 35 papers are published in SCIE indexed high impact factored journals. Among the international conference publications, four papers have received "Best Paper Award ". Among the SCIE publications, one paper published in Review of Scientific Instruments (RSI) Journal (under American International Publishers) has been selected as "Editor Choice Paper of the Issue" in 2016. He has been instrumental in chairing/co-chairing around 30 international conferences in India and abroad as Publication/TPC chair and edited 50 book volumes from Springer-Nature as a corresponding/co-editor/author on date. He has delivered nearly 20 keynotes, invited talks in international conferences, ATAL, TEQIP and other AICTE sponsored FDPs and STTPs. He has been Editor-in-Chief of IGI Global--International Journal of Natural Computing and Research (IJNCR) an ACM & DBLP indexed journal from 2017-22. He has guest edited Special Issues in reputed SCIE indexed journals under Springer-Nature and Elsevier. He is Senior Member of IEEE and Life Member of CSI.

Jerry Chun-Wei Lin

He received his Ph.D. from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan in 2010. He is currently a full Professor with the Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway. He has published more than 500+ research articles in refereed journals (with 90+ ACM/IEEE transactions journals) and international conferences (IEEE ICDE, IEEE ICDM, PKDD, PAKDD), 16 edited books, as well as 33 patents (held and filed, 3 US patents). His research interests include data mining and analytics, natural language processing (NLP), soft computing, IoTs, bioinformatics, artificial intelligence/machine learning, and privacy preserving and security technologies. He is the Fellow of IET (FIET), ACM Distinguished Member (Scientist), and IEEE Senior Member.

Sentiment Analysis and Classification of Hotel Opinions in Twitter With the Transformer Architecture

Sergio Arroni, Yeray Galán, Xiomarah Guzmán-Guzmán, Edward Rolando Núñez-Valdez*, Alberto Gómez

Department of Computer Science, University of Oviedo, Oviedo (Spain)

Received 7 May 2022 | Accepted 25 October 2022 | Early Access 7 February 2023

Abstract

Sentiment analysis is of great importance to parties who are interested is analyzing the public opinion in social networks. In recent years, deep learning, and particularly, the attention-based architecture, has taken over the field, to the point where most research in Natural Language Processing (NLP) has been shifted towards the development of bigger and bigger attention-based transformer models. However, those models are developed to be all-purpose NLP models, so for a concrete smaller problem, a reduced and specifically studied model can perform better. We propose a simpler attention-based model that makes use of the transformer architecture to predict the sentiment expressed in tweets about hotels in Las Vegas. With their relative predicted performance, we compare the similarity of our ranking to the actual ranking in TripAdvisor to those obtained by more rudimentary sentiment analysis approaches, outperforming them with a 0.64121 Spearman correlation coefficient. We also compare our performance to DistilBERT, obtaining faster and more accurate results and proving that a model designed for a particular problem can perform better than models with several millions of trainable parameters.

Keywords

Artificial Intelligence, Machine Learning, Natural Language Processing, Sentiment Analysis, Transformer Architecture.

DOI: 10.9781/ijimai.2023.02.005

I. Introduction

In the last few years, there has been immense growth in the field of Natural Language Processing (NLP) and, especially, in the application of machine learning methods to NLP problems.

With the increase in popularity of deep learning, for some years the focus was on the research of neural network structures that make use of convolutional layers [1] and recurrent layers [2] for language understanding and processing. These architectures brought about a big wave of research for their application to NLP, since they allowed much more detailed representations that the standard feed-forward models [3]. Convolutional networks allow looking at text by parts through filters, which then are aggregated for a global interpretation. Recurrent networks, on the other hand, process the text input sequentially but, aside from the part of the input being currently processed, they take into consideration outputs from previous parts as an additional input, hence the name recurrent networks.

A few years ago, however, the proposal of attention-based neural network models by Vaswani et al. [4] has shifted great part of the research in deep learning for NLP towards the development of transformer structures and pre-trained models with hundreds of millions of trainable parameters that only require some fine-tuning training to be applicable to a wide range of NLP tasks [5]–[7].

* Corresponding author.
E-mail address: nunezedward@uniovi.es

However, NLP can be of special interest to businesses or parties who are interested in knowing the public opinion about something in general purpose social networks (e.g., a restaurant might be interested in knowing whether customers like or dislike its food, but not so much in being able to generate AI-written text or in artificial question answering). For that purpose, a simpler and smaller model might suffice or even obtain more accurate results than models which are pre-trained for multiple NLP tasks.

We explore sentiment analysis, which is a subfield in NLP that deals with processing a piece of text and obtaining the general sentiment included in it. Several advances have taken place lately towards more precise sentiment analysis, ranging from basic and rudimentary approaches [8] to complex neural network systems.

In this work, we implement a neural network system using the state-of-the-art attention-based Transformer structure, with a dramatically lower number of trainable parameters and size than those of the previously mentioned pre-trained models. In contrast to other machine learning NLP approaches like recurrent [2] and convolutional [1] neural networks, which tend to lose information when the text is too large, this structure manages to process the text input in a single iteration, which increases the speed and the ability to understand the context of the whole text.

Thus, our aim is to make use of the Transformer neural network architecture and obtain a model that greatly improves the prediction accuracy of those more basic methods, while not resorting to the complexity of training millions or billions of parameters, proving that a simpler and faster model crafted for the task at hand can perform better if trained for a particular problem. We tackle the problem of

predicting the general opinion about hotels in Las Vegas from Twitter data, making use of the dataset provided by Philander & Zhong [8].

The study by Philander & Zhong [8] served as a motivation for the possibility of improving the obtained results, so we came up with the idea of trying to use a neural network to predict the sentiment in the tweets. Our first approach to this work consisted in following a similar approach to the original study while adding the computation of bigrams and trigrams, but the results obtained were not significantly better compared to those of the original work. We improved on this by adding a machine learning algorithm. Knowing the real-world applications of sentiment analysis, we want to obtain the best possible result for a problem we found interesting.

In a first instance, we opted for using a HuggingFace pre-trained DistilBERT [6] model, which can tokenize and prepare the input and uses a transformer neural network model. However, as it is a pre-trained model, its structure and parameters are not modifiable, which in combination with the fact that the model contains millions of parameters for general NLP tasks that are not needed for our specific problem, motivated our decision to create our own attention-based model from scratch. We then compared our model to DistilBERT for evaluation against a strong state-of-the-art model.

The remainder of this work is structured as follows: Section II discusses the related work in the state of the art about NLP, sentiment analysis, machine learning applied to NLP and the transformer architecture. Section III presents our proposal, including the dataset used, the model architecture and the evaluated metrics. Section IV details the experiments carried out and the results obtained in terms of the metrics and execution time. Lastly, Section V offers our conclusions and some proposed future work in relation to our study.

II. Literature Review

In this section, we discuss the main aspects that we deal with in this work: sentiment analysis, deep learning, and the transformer architecture for NLP.

A. Natural Language Processing and Sentiment Analysis

NLP has attracted a lot of attention in recent times, and great advances are being achieved in this field. NLP is a field of study that is being researched since more than 50 years ago and it is one of the most widely spread topics in which artificial intelligence is applied. Its purpose is to enable computers to understand words written by humans and process them to reach conclusions related to the problem at hand.

NLP approaches usually involve several linguistic aspects, like semantics, phonology [9], morphology [10] or syntax [11] of written or spoken natural language. Nowadays, however, most of the research is oriented towards the application of machine learning to NLP problems [3]. Some deep learning models are developed almost exclusively for NLP tasks [4], considering the needs for text processing and sequence generation, and brought a breakthrough to the field achieving great results in several NLP tasks like translation or question answering.

We can find two different cases in NLP: natural language understanding and natural language generation. A particular case of natural language understanding is text classification, which deals with the problem of assigning a category to a text. Sentiment analysis can be seen as a generalization of text classification, as it attempts to analyze a piece of text and find the general sentiment included in it. Other subfields in text classification are topic detection or language detection. We will focus on sentiment analysis, as our goal is finding the general opinion present in a text, that is, assigning a label to a text.

Going further into the field of sentiment analysis, it tries to identify and obtain subjective information from a given text as input. This analysis provides us with information that gives us a result of emotional tone, such as positive, negative, happy, sad, angry, etc.

There are mainly two state-of-the-art approaches to sentiment analysis. One of them is the lexicon and rule-based method, which consists in making a decision on the sentiment in the tweet according to whether specified conditions are met [12]–[14]. However, most state-of-the-art approaches to sentiment analysis nowadays include the previously mentioned deep learning models [5], [15], [16], one of which is the transformer architecture that we study in this work. We will conduct experimentation comparing methods from both groups and attempt to show that our deep learning model has greater potential.

By obtaining this result we can categorize the text within an emotional spectrum, being able to group them by feelings. This has a wide range of direct applications, including product or service reviews [17], analysis of social network data [18], marketing and branding [19], financial analysis and forecasting [20], detection of emotions in conversations [21] and many others.

As shown in the previously mentioned studies, sentiment analysis also finds great use in fields that do not necessarily have a direct relationship with computer science and is often used for the processing and analysis of Big Data.

As Philander & Zhong [8] say in their work, this analysis that we do can have a great impact on the hotels we analyze, as it provides them with very useful information that they would take a long time to get by hand. Sentiment analysis as a whole possesses great applications in industry, in fields where customer opinions are of great relevance such as product or service reviews on the web [22]–[25] or prediction of stock markets and prices [26], [27] and even in fields like opinion analysis in politics [28] or medicine [29].

B. Deep Learning for Natural Language Processing

Machine learning is being applied to NLP tasks for over two decades. Some years ago, standard machine learning approaches like random forests [30] and support vector machines [31] were the state-of-the-art methods for learning text representations.

However, with the increase in computational power and the popularity of neural network models, deep learning soon took over the field. Apart from the conventional feed-forward models, the development of convolutional and recurrent neural network models brought about a whole new world of approaches to the processing of natural language, both in the form of text and voice [32].

Convolutional neural networks, as first proposed by Kunihiko Fukushima [1], process an input by looking at different parts of the whole through a filter and shifting the filter through the input. Those results are then aggregated in different ways for obtaining the desired output. As could be expected by this brief description, convolutional neural networks have found the most success in the processing of images or computer vision [33]. However, the model can be applied to NLP in the same way, as the processing of text greatly benefits from a partial look at different parts of it for computing a representation of the input [34].

On the other hand, recurrent neural networks, as proposed by Rumelhart et al. [2], process an input iteratively one by one, but compute each representation by using the previous output as well. Thus, the final output contains information about the whole sequence. However, the earliest information is sometimes lost because of the vanishing gradient problem. To address this problem, researchers come up with models like LSTM (Long Short-Term Memory) [35], which introduces an additional input that contains the previous

unprocessed inputs. As expected, this model has found great success in the processing of sequential data, one of which is text.

One of the problems with those models, however, is the concept of distance in the sequences, e.g., the first word in a sentence will always be furthest away from the last word when being processed by recurrent models, despite that not being necessarily the case for meaning in language processing, since two words can be closely related even if there are several words between them. For solving this, the attention mechanism is introduced, which is able to compute dependencies equally between every single element in the sequences. This mechanism is mostly used in combination with recurrent or convolutional models until 2017, when the transformer architecture is introduced by Vaswani et al. proving that using attention is enough and that recurrency and convolutions are not needed [4].

C. Transformers

Transformers are a neural network architecture based solely on the attention mechanism which was introduced first in a work by Vaswani et al. in 2017 called Attention Is All You Need [4].

Transformers, like other neural network architectures [36], are based on an encoder-decoder model, where the encoder is responsible for analyzing a sequence of input data and obtaining an encoding, and the decoder is responsible for obtaining an output from the encoding. As its structure suggests, this model is mainly aimed towards the translation or transformation of the input into a similar output [37], that is, computing different representations of the input data. However, the decoding process can be adapted to a much greater variety of tasks.

In the field of NLP, the transformer architecture is used to solve multiple tasks including text classification [38], translation [39], question answering [40], summarization [41] or text generation [42]. In this study, we focus on text classification.

As described in the study by Vaswani et al. and applied to our problem, given an embedding matrix E with embeddings of size d_e for tokenized texts, self-attention is calculated as described in (1).

$$a(E) = \sigma\left(\frac{EE^T}{\sqrt{d_e}}\right)E \tag{1}$$

As we can see in equation (1), σ is the softmax function. For multi-head attention, tree matrices Q, K, V will be linearly projected from E for each head $i \in \{1, ..., h\}$ by means of transformation matrices W_i (three in total for each head). Therefore, there will be $3h$ matrices that will be learned by the model. Attention is computed then not as self-attention but as regular attention, as shown in (2).

$$a(Q_i, K_i, V_i) = \sigma\left(\frac{Q_i K_i^T}{\sqrt{d_e}}\right)V_i \tag{2}$$

The results are then concatenated and projected back to the original shape. The original proposal uses 8 attention heads. This process does not imply a big increase in the total operation time as the size of the computations in each head is reduced by the projections. Finally, the outputs are passed through densely connected layers until the final probabilities for text classification are obtained.

In other architectures like recurrent networks or convolutional networks, the text input is analyzed sequentially or by segments, which often causes the loss of early information due to the problem of vanishing gradient [43]. In the transformer architecture, by only making use of attention as described above, the whole text is processed in one go, which allows greater precision for analyzing the relative position and meaning of big texts.

Not only that, but transformers have also found great success in other fields that do not involve the processing of text, like the processing of audio, mainly in the field of speech recognition [44], the processing of images [35], [36] or even other kinds of sequence generation like chemical chains [45]. Lin et al. [46] present a survey of the most relevant contributions to the attention mechanism and the transformer architecture over the past few years.

In the field of NLP, most research has been centered around developing pre-trained models [47]. These models are still being researched as of today for the creation of more accurate language representation models that can be fine-tuned for a large number of different problems.

In this work, our goal is to craft a transformer architecture model much simpler and smaller than the mentioned models, and show that for specific problems, there is no need to obtain a model that can contain a whole language representation, just the information needed for said problem. Additionally, we want to show that the transformer architecture is very powerful no matter the size of the model and that it also works for smaller problems without millions or billions of trainable parameters.

III. Proposal

This section details our proposal, offering a view of the process of preparing the dataset, an in-depth description of the model created and the details of the computation of ratio score and evaluation metrics used for comparisons.

A. The Datasets

This subsection details the datasets used for this work: the tweet dataset, the Datafiniti review dataset [48] and the Amazon review dataset [49].

1. Tweet Dataset

In our study, we use the tweet dataset offered by Philander & Zhong [8]. The dataset contains the tweets tagging hotels in Las Vegas within two periods of time: from August 16, 2013, to September 13, 2013, and from October 25, 2013, to November 15, 2013. However, no kind of labelling was included, so we opted for manually classifying some of them to be able to employ a machine learning algorithm. We started with 2701 classified tweets, 2014 of which are positive, 250 are negative and 437 are neutral.

We use this dataset as the main dataset so that we can compare our method with the method proposed by Philander & Zhong under similar data, so that the results are not biased towards our model, and we can make a fair comparison between their manual classification method and our deep learning model.

We want to clarify that the manual classification took into account emojis, exclamation marks, ironies and other colloquial expressions. In order to carry out this classification, a standard was followed as it was done by several members of the team in different periods of time. This standard is divided into four different classifications for each tweet:

Positive tweets: we decided to classify as positive tweets those that had a clear and undoubtedly positive feeling towards their stay at the hotel. We encountered many tweets of people that are just happy or attracted by some celebrity or contest that is popular at that moment. Those tweets do not necessarily have anything to do with their opinion on the hotel and therefore have not been classified as positive. Some other tweets talk about their excitement of staying in the hotel for the first time, these tweets have not been categorized as positive either as we understand that they do not provide any information on features of the hotel or their stay as they have not been there yet. Tweets that speak positively about the hotel's facilities or the hotel's service have been classified as positive.

Negative tweets: following the pattern of positive tweets, both events and famous people mentioned negatively in the tweets have

not been classified as negative. There was a problem mentioned in the work by Philander & Zhong [8], which alerted us of the low number of negative tweets. Due to this, in this classification we were less strict in classifying a tweet as negative, classifying one as such at the slightest hint of doubt or discomfort from a customer.

Neutral tweets: in this category are all tweets that do not meet the conditions for the other sentiments since we understand as neutral all tweets that do not talk about the hotel or do not say anything significant about it. Tweets that only tag the hotel but do not explicitly talk about it are classified as neutral since they do not reflect a sentiment on the hotel but are useful nonetheless for learning to separate opinions about the hotel from opinions about something else. Tweets that say both positive and negative things about the hotel but do not clearly emphasize one of the two are also classified as neutral.

Tweets that did not fit in any field were deleted, as there are some that are either empty or do not include any type of information, neither about the hotel nor about any other topic, or were written in a language other than English. There were also tweets that were too ambiguous, and we would not be able to add them to any of the sentiments described, so in order not to include unnecessary noise in the training data, these tweets were also deleted. Table I shows some examples of tweets as manually classified by us.

TABLE I. Examples of Tweets Classified by Hand

Positive
"@AriaLV loved every minute about staying at the Aria very safe modern and overall great atmosphere will stay there again!!" *"So excited for Vegas now! Looking forward to staying at the best hotel on the strip @TheMirageLV"* *"What a beautiful day in #Vegas. The sun is shining our pool is #Shimmering and we have #rooms to sell @TropLV ! What could be better?"*
Negative
"Hey @HardRockHotelLV your customer service leaves MUCH to be desired. If #Pubcon is smart you won't be the partner hotel next year." *"@RivieraLasVegas Did you ever replace the lamp in room 3533? Might wanna clean the puke off the walls too. So gross!"* *"@AriaLV @myVEGAS no.. but a nice stay would thankful for once :)"*
Neutral
"*Hey @TropLV I love your hotel, but the service in your beach cafe was atrocious tonight. You guys are better than that."* *"Went to @Rock_Vault @LVHHotelCasino...photo with @RobinMcauley from Survivor #80srock"* *"The Eiffel Tower at @parisvegas will award lucky 10 Millionth visitor with trip for 2 to Paris, France. Learn..."*

2. Datafiniti Review Dataset

The Datafiniti reviews [48] in the dataset were categorized into star ratings between 1 and 5. We refer to the Likert scale [51], which stipulates that on a 5-point scale, each extreme represents an opposing opinion, in this case, positive and negative, at the intersection of these extremes, point 3, represents an opinion that is indifferent to the subject or, in many cases, neutral.

Thus, following this scale, we decided to classify 1-to-2-star reviews as negative, 3-star reviews as neutral and 4-to-5-star reviews with positive sentiment. In the end, the dataset has 6048 negative reviews, 5709 neutral reviews and 22429 positive reviews.

3. Amazon Review Dataset, Hugging Face

Similarly, to the Datafiniti dataset [48], the Amazon dataset [49] contained reviews with star ratings. We used the same method of classification as before, marking reviews from 1 to 2 stars as negative, reviews with 3 stars as neutral and those from 4 to 5 stars as positive.

B. The Attention-based Transformer Model

We use a neural network structure based on self-attention as proposed by Vaswani et al. [4]. The transformer architecture makes use of self-attention, which by computing the dot product of every pair of tokens, is able to process relationships between elements at any distance, something that other models like recurrent and convolutional networks struggle with. This is a very important feature in the representation and understanding of natural language, which is the main reason why we opt for the transformer architecture in this study [4].

According to Vaswani et al. [4], the computational efficiency per layer and precision also improves that of other models, which added to the fact that our available computational power is not great, and tweets are short and require higher precision, serves as another reason for our choice. We consider using other architectures, classical recurrent networks like LSTM or convolutional networks, but since those are nowadays less efficient and, unlike transformers, have trouble scaling to bigger inputs, we in the end opted for the transformer architecture being a successful newer and very interesting approach.

The network structure consists of a single transformer block with 11 attention heads as described in their study, with the main difference being that we use learned embeddings of dimension 12 for positional encoding (as well as token encoding) instead of sine and cosine functions. These vectors of dimension 12 are able to capture the information and context of the tokens in a continuous space without unnecessarily over-increasing the complexity of learning the transformation. Token embeddings and position embeddings are added and fed to the multi-head attention layer, the output of which is then passed through a dropout layer and a normalization layer before being used as an input for the feed-forward layers. The fully connected layers inside the transformer blocks contain 768 neurons as proposed by Devlin et al. [25]. After experimenting with different sizes, we decided against decreasing the size of these layers, since they contribute greatly to the mapping of features extracted by the attention layers to the outputs. Residual connections are used around the multi-head attention layer and the feed-forward layer inside the transformer block. The output is fed to an average pooling layer to reduce its dimensionality to a 1d vector of size 12 (the dimension of the learned embeddings) which integrates all the information obtained from the transformer block, and through a 16-neuron feed-forward layer that is fully connected to the final 3 probabilities, which are obtained by a SoftMax function. Fig. 1 offers a visualization of the described model structure.

The data goes through a preprocessing stage before entering the neural network. The tweet texts are first cleaned of all tags starting with "@" and hyperlinks starting with "http" since these only add noise to the tweets, and then we make use of the Keras [52] text tokenizer for tokenizing the tweets in our dataset. We decide to leave the hashtags starting with "#" since they can carry information related to the sentiment expressed in the tweets [53]. The vocabulary size used for tokenizing the tweets is 20,000 words, which we found to be an appropriate size for our problem, given that tweets are usually not very long, but they contain a large range of words that are not necessarily part of the English language (like abbreviations, Internet slang, emoticons, and so on). Tokenized tweets are then padded to a length of 20 tokens. This length was chosen for allowing a slightly above average number of words in a maximum-length tweet (the average for the English language was 17 words when the maximum tweet length was 140 before being increased to 280 in 2017) while not dramatically increasing the neural network input size. For the analysis of longer format reviews or posts, the text will still be padded to 20 tokens, possibly causing some loss of information in some cases where the sentiment is expressed in the latter part of the text. In those cases,

we recommend increasing the maximum input length and slightly increasing the complexity of some layers so that the whole text can be processed with similar representational power.

Our contribution consists in the use of a manually created neural network based on the transformer architecture. We perform a study of the different parameters and structures of the model using a grid search method and arrive at the model described above. With this approach, we manage to achieve better results than those obtained in the study by Philander & Zhong [8] and other more complex models while drastically lowering the number of trainable parameters and thus, the time needed for training and making predictions.

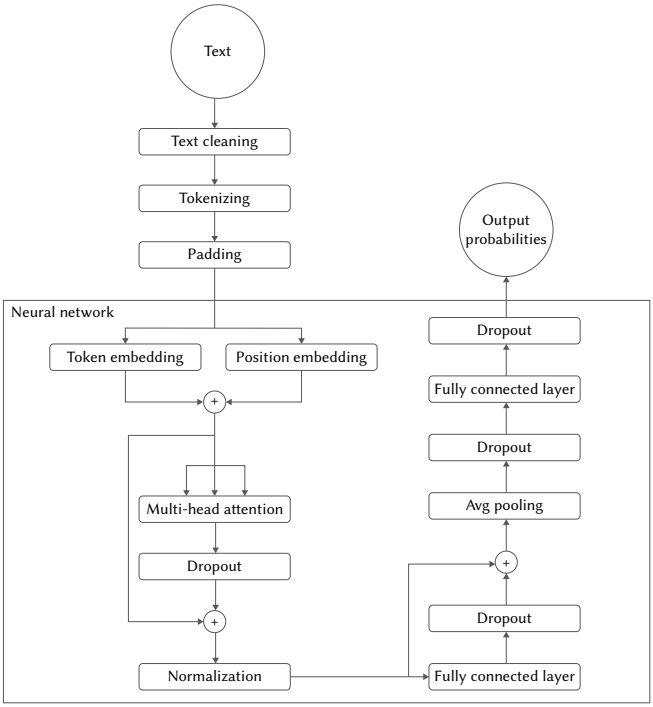


Fig. 1. Model architecture.

C. Ratio Score

The resulting classified tweets are used for computing a ratio score for each of the hotels in the dataset. The calculations are similar to the ones proposed by Philander & Thong [8]. They propose the ratio score of a hotel as the quotient between the total number of positive tweets and the total number of negative tweets related to a hotel. It is formally defined in (3), where n is the total number of tweets related to hotel h, p_i is the number of positive words in tweet t and n_i is the number of negative words in tweet i. For a predicate p, the function 1_p is defined in (4).

$$score(h) = \frac{\sum_{i=1}^{n} \mathbf{1}_{p_i - n_i > 0}}{\sum_{i=1}^{n} \mathbf{1}_{p_i - n_i < 0}} \tag{3}$$

$$\mathbf{1}_p = \begin{cases} 1 & p \\ 0 & \neg p \end{cases} \tag{4}$$

We use this definition in some of our experiments for a fair comparison between the method proposed by Philander & Thong [8] and our model.

However, since our model does not only obtain the label of the text, but instead obtains the probability that the model considers for each of the labels, we propose a new method for calculating the ratio score of each hotel that is much more flexible and makes use of the probabilities to obtain a much more informative representation of the ratio score.

Instead of computing the ratio between the number of positive tweets and the number of negative tweets for each hotel, we consider the probability obtained by the model of each label for each tweet. That is, the new ratio of a hotel is calculated as shown in (5), where n is the total number of tweets related to hotel h and l_i is the label obtained for tweet i.

$$score(h) = \frac{\sum_{i=1}^{n} P(l_i = positive)}{\sum_{i=1}^{n} P(l_i = negative)} \tag{5}$$

The regular ratio score is a very rigid method of scoring hotels since it can only classify tweets as positive or negative and every tweet has the same weight towards the final score. Since not every positive tweet is equally positive and not every negative tweet is equally negative, we put forward this method that computes "how positive" and "how negative" each tweet is, regardless of its final classification. Thus, we intend for this method to perform better at predicting relative performance between hotels than the regular ratio score.

D. Evaluation Metrics

In the following experimentation, we propose the usage of two metrics to evaluate and compare the performance of different models: accuracy on the validation set and the Spearman correlation coefficient with a TripAdvisor ranking.

Validation accuracy is a very popular metric for the evaluation of machine learning algorithms in classification problems. It computes the ratio of correctly predicted samples among all the samples in the validation set, that is, not including the samples used for training. More formally, the validation accuracy is defined in (6), where V is the validation set, y_i is the expected output of sample i and $\hat{y}_i$ is the predicted output for sample i.

$$accuracy = \frac{\sum_{i \in V} \mathbf{1}_{y_i = \hat{y}_i}}{|V|} \tag{6}$$

We make use of this metric as a method of gauging how reliable the models are at correctly predicting the sentiment in individual tweets.

On the other hand, for a more global evaluation, we compute the Spearman correlation coefficient. We choose this metric for a fair comparison to the study by Philander & Zhong, since it is the metric that they propose for comparing rankings. This coefficient is defined as the Pearson correlation coefficient between the ranking of the values in each variable. We can take the Pearson coefficient as the expression in (7), where σ is the standard deviation and cov is the covariance.

$$r_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{7}$$

Equation (8) then shows the definition of the Spearman correlation coefficient, where $R(x)$ is the ranking of variable x ordered by its value.

$$\rho_{X,Y} = r_{R(X),R(Y)} \tag{8}$$

We compute this coefficient by taking as X the ratio score of every hotel, calculated as described in the previous section, and taking as Y the TripAdvisor score obtained from the study of Philander and Zhong [8]. The coefficient is always contained in the interval [-1, 1], meaning a weak correlation in values close to 0, a strong negative correlation (rankings move in opposite directions) in values close to -1 and a strong positive connection (rankings move in the same direction) in values close to 1.

This metric obtains a more general view on how accurate the models are at predicting the relative quality of hotels. We opt for using both proposed metrics as they serve as a means of evaluating different aspects of the goodness of the models. However, in the experimentation phase, we put more focus on the Spearman coefficient for comparison purposes, given that the more general goal of the study [8] we aim to compare our models with, is obtaining an accurate overview of the relative performance of hotels between themselves.

IV. Experimentation

Our goal is the application of machine learning to tweet labelling to improve the results obtained by Philander & Zhong [45]. This study makes use of a random selection of tweets from the dataset presented by them, containing a total of 2701 tweets, of which 2014 are positive, 250 are negative and 437 are neutral. Only part of the tweets has been selected so that enough of them are left for predicting. We use a random selection of 80% of those tweets as training samples for our machine learning system and the remaining 20% for testing.

After training, the remaining tweets are classified and labelled by the system. In another approach for attempting to train our model with more data, we also make use of an external dataset from Datafiniti [48] containing hotel reviews and classify our tweets by the model trained with said dataset. After all the tweets are classified, we obtain the ratio score of each of the hotels with the two methods described in the proposal. With these ratios we compute the proposed metrics, and, in the end, we compare the complexity and time needed for each model. In brief, we train the same model with the two described datasets separately and predict the remaining tweets for comparing the results.

As the datasets are imbalanced towards positive labels, we attempt to mitigate the difference in number of samples by class by applying greater training weights to the samples that have "negative" and "neutral" labels. However, we find that despite not affecting the average results much, it has a negative impact on the prediction for relative performance of hotels. We believe this is possibly caused by overfitting on negative and neutral tweets due to applying greater weights to each of them, which would in turn cause the model to mistakenly predict some tweets as negative or neutral that are related to the overfit samples. When this happens for a tweet that is closely related to one particular hotel, it can cause its ratio score to drop dramatically. For this reason, we leave the number of tweets for each label as is, since we observe that the models still manage to correctly learn to predict negative and neutral tweets as well.

A. Hardware

We run the computational experiments on a 64-bit Windows Server 2016 with two Intel Xeon Silver 4208 CPUs at 2.1GHz and 6GB of RAM memory. It should be noted that no CUDA-compatible graphics unit is present, so all computing is done in the specified CPU. Every instance of the experiments is run on the same machine for a fair comparison of time and efficiency.

B. Metric Measurement

This section presents the results obtained in terms of validation accuracy and Spearman correlation coefficient.

1. Single Results

In a first instance of experimentation and for a better adjustment of our model, we carry out a single execution of the training and evaluation process several times. In order to ensure that the results are deterministic and that they are not influenced by random variation, we employ a random seed and execute every instance over it.

Philander & Zhong [8] offer the Spearman correlation coefficient (ρ from now on) between their obtained score for each hotel and the TripAdvisor score at the time as the validation metric. For a fair comparison, in a first instance, we compute the ratio score as proposed by them. With the same TripAdvisor score (obtained from their article [8]) and our own obtained ratio score for each hotel, we manage to obtain a ρ of 0.601 (p = 0.0001) between our hotel ranking and the TripAdvisor ranking, which we consider a high correlation considering that the mean score among TripAdvisor reviews does not always necessarily match the general sentiment expressed by people on Twitter. Table II shows some examples of the classification of tweets by our model.

TABLE II. Examples of Tweets Classified by Our Model

Positive
"*VEGAS I just got back and stayed at and it was AMAZING Vegas is perfect for bachelorette parties*" - Pos: 86.93%; Neg: 2.29%; Neu: 10.78%. "*loved every minute about staying at the Aria very safe modern and overall great atmosphere will stay there again*" - Pos: 82.36%; Neg: 3.56%; Neu: 14.08%. "*Absolutely loved the rooms Luxury*" - Pos: 69.1%; Neg: 7.22%; Neu: 23.68%. "*I've stayed in a lot of nice places but might just be the nicest... Only problem is I get lost ALL THE TIME. #itshuge*" - Pos: 41.99%; Neg: 22.26%; Neu: 35.73%.
Negative
"Where is the 'cl*ean window*' button in my room? ;)" - Pos: 23.77%; Neg: 41.06%; Neu: 35.17%. "*Good morning from It's a beautiful day, but I think the windows need washing :/*" - Pos: 33.14%; Neg: 36.08%; Neu: 30.78%. "*#APALAcon13 has joined workers ; No contract, no peace #CantStopWontStop*" - Pos:18.37%; Neg:47.56%; Neu:34.06%. "*on my do not serve list. Doorman talked 2 fares into a shuttle who we're asking for a cab. Then told me to shut the fuck up*" - Pos: 0.71%; Neg:82.28%; Neu:10.59. "*No microwave AND no fridge? I know this is only the Manor Motor Lodge at but STILL...*" - Pos: 18.63%; Neg:46.46%; Neu:34.90%.
Neutral
"Hey-o happy hour Drinking a 312 Urban Wheat Ale by atwashing :/" - Pos: 28.95%; Neg: 32.6%; Neu: 38.44%. "See you in November :))" - Pos: 30,44%; Neg: 33.09%; Neu: 36,46%. *"Caught some of the set. Awesome* voice. #tingling" - Pos: 38.31%; Neg: 19.79%; Neu:41.88%. "The commercial on MTV with ML's original don playing in the background literally just blew my mind #ilovevegas" - Pos: 28.57%; Neg: 30.41%; Neu: 41.00%.

We make a ratio score calculation as they did in the original work, in which we add one to the total score of that hotel if the tweet mentioning that hotel is classified as positive and subtract one if the tweet is negative, if the tweet is neutral, we neither add nor subtract, it remains the same, as seen in the ratio score formula given by Philander & Zhong [8].

Further experimentation was carried out for more reliable validation. We applied the pre-trained state-of-the-art transformer model developed by Sanh et al. [6] to our problem, following two different approaches: fine-tuning the model with further training on a dataset containing Amazon reviews [49] and their sentiment and fine-tuning it with our own manually classified tweets. This model was created following the template provided by HuggingFace [7] and adapted to the problem by us. The fine-tuning on the Amazon dataset [27] manages a 0.735 accuracy on our classified tweets, and a borderline non-significant ρ of 0.338 (p=0.05) with the TripAdvisor ranking. The fine-tuning on our own tweets reports a 0.808 accuracy and a ρ of 0.447 (p=0.007). As these results show, the accuracy on the validation data is slightly higher in the fine-tuned model than in our own model, but the Spearman correlation with the TripAdvisor ranking is significantly lower. Despite the accuracy being roughly the same, we can observe that the fine-tuned model makes more negative predictions. This greatly influences the score since the positive-

negative ratio is usually greater than 1, and so this model does not do as well as our own in terms of the Spearman coefficient.

For our own model, we studied the option of training the neural network with a different bigger dataset from Datafiniti [48]. This dataset contains around 35000 hotel reviews that we classify as positive, negative, and neutral according to their star rating. We decided to observe the result that this new training obtains to rule out one of the possible areas for improvement, which would be increasing the number of manually classified tweets and having a larger dataset. The results were slightly worse than those of the tweet dataset but still better than the other models and methods, obtaining a validation accuracy of 0.706 on our own tweets and a ρ of 0.569 (p = 0.0004) between the obtained ranking and the TripAdvisor ranking, which a priori rules out the problem of having a small dataset used for training in our approach. Table III shows the comparison between all the results presented thus far.

TABLE III. Validation Accuracy and Spearman Correlation for Each Model

Model	Spearman p	Val accuracy
Philander & Zhong	0.501	
DistilBERT (Amazon)	0.338	0.735
DistilBERT (tweets)	0.447	0.808
Our model (tweets)	0.601	0.803
Our model (Datafiniti)	0.569	0.706

However, we suspect that the improvements achieved by the adjustments made over a single deterministic instance, while relevant to the problem at hand, might not completely reflect the general performance of the model for other cases, since we are optimizing the parameters for only that one case. Thus, for further validation, we make several random executions and compute the average results obtained.

2. Average Results

To get more representative data, we run 25 iterations of each model, under the same conditions. As expected, we obtain somewhat lower results in some models than those previously mentioned using a specific random seed. We obtain a mean of the iterations for making more representative comparisons between models. Table IV shows the average validation accuracy and average Spearman correlation coefficient obtained.

TABLE IV. Average Validation Accuracy and Spearman Correlation for Each Model

Model	Spearman ρ	Val accuracy
Philander & Zhong	0.501	
DistilBERT (Amazon)	0.2855	0.8288
DistilBERT (tweets)	0.4386	0.8288
Our model (tweets)	0.4713	0.7639
Our model (Datafiniti)	0.5763	0.7108

In terms of the Spearman coefficient, the best result is obtained by our model trained with the Datafiniti dataset [48] with a ρ of 0.5763, the next would be our model trained with the tweets dataset [4] with a result of ρ=0.4713, and finally, both models that use DistilBERT [8] either trained with our data [4] or with Amazon reviews [49], for which Spearman results are ρ=0.4385 and ρ=0.2855 respectively.

After observing the data, we can conclude that our approach is superior to the one used in the original work [4], which obtains a ρ of 0.501, and superior to DistilBERT's model [6] which, when trained with our data [4], does not reach the spearman result obtained in the original work [4] (ρ=0. 4385). And lastly, within our model, the one trained with Datafiniti [48] (ρ=0.5763) is superior to the one trained with our tweets [8] (ρ=0.4713), so we can suggest that the results would be greatly benefitted from possessing more training data and that the formal review format could potentially be more accurate for predicting the general opinion on hotels than tweets. A possible reason for this phenomenon might be that formal language is more standardized and carefully written than informal language seen in social networks, which usually includes slang and misspelt words are very frequent, making training harder.

After these experiments, we obtain the average ratio score for each hotel across all executions for obtaining a more representative ranking of the hotels. In Table V we present a view of the TripAdvisor scores, the ratio scores by Philander & Zhong [8] and our ratio scores.

TABLE V. Ratio Score of Hotels by Each Model

	TripAdvisor	Philander & Zhong	Our model (tweets)	Our model (Datafiniti)
Palazzo Las Vegas	88	7.04	97.4168	48.6633
Bellagio Las Vegas	88	9.6	27.4728	35.9129
M Resort Spa Casino	87	5.96	33.4462	38.7507
Red Rock Las Vegas	85	10.22	80.2095	27.3824
Venetian Las Vegas	85	8.66	229.6565	27.6740
Aria Las Vegas	84	10.72	39.7088	36.5600
Wynn Las Vegas	84	6.79	44.8549	19.3175
South Point Hotel	84	5.88	103.8230	21.4435
The Mirage	84	4.9	75.6752	19.6242
MGM Grand Hotel	81	4.41	210.1098	29.3816
Tropicana Las Vegas	80	11.1	70.2524	22.4831
NYNY Vegas	79	5.08	48.8507	25.1627
Golden Nugget	79	3.13	14.6022	15.7709
The Cosmopolitan	77	5.2	75.4652	29.7664
Mandalay Bay Resort	74	5.78	79.9043	25.5889
Paris Las Vegas	73	9.9	18.0115	23.6796
Treasure Island	72	6.66	25.2909	18.3458
Caesars Palace	72	6.08	34.5140	24.0737
Monte Carlo Resort	70	6.81	29.6825	19.8006
Planet Hollywood	68	3.39	33.5736	16.6929
Bally's Las Vegas	68	2.62	21.1096	8.6210
Stratosphere Hotel	66	6.05	77.9540	22.8341
Palms Casino Resort	66	5.24	240.7661	29.1991
Hard Rock Hotel LV	64	2.64	46.5190	16.5550
Harrah's Las Vegas	63	4.03	14.7001	13.2363
Luxor Hotel & Casino	60	6.1	72.4207	23.3794
Circus Circus	60	5.68	28.3268	17.2063
Excalibur Las Vegas	60	5.53	18.1248	22.3499
Rio Las Vegas	59	4.61	55.3870	16.3660
Flamingo Las Vegas	55	5.42	13.6495	11.7607
Hooters Casino Hotel	55	4.64	16.1221	10.7504
Riviera Las Vegas	48	5.23	34.4220	26.4814
LVH Hotel & Casino	47	5.86	26.7072	20.7079
The Quad Las Vegas	43	2.57	5.2856	9.0395

With the scores in the fourth and fifth columns, we calculate the Spearman correlation coefficient again against the TripAdvisor scores in the second column, obtaining 0.48099 by training with our tweets and 0.60969 by training with the Datafiniti dataset [48] in this case. From these results, we can observe that the extensiveness of the training data affects the model's capability for obtaining relative performances to a great extent. Table VI offers a visualization of the rankings predicted by each model.

TABLE VI. Ranking of Hotels by Each Model

	TripAdvisor	Philander & Zhong	Our model (tweets)	Our model (Datafiniti)
Palazzo Las Vegas	1	7	5	1
Bellagio Las Vegas	2	5	24	4
M Resort Spa Casino	3	14	21	2
Red Rock Las Vegas	4	3	6	9
Venetian Las Vegas	5	6	2	8
Aria Las Vegas	6	2	17	3
Wynn Las Vegas	9	9	16	23
South Point Hotel	7	15	4	19
The Mirage	8	25	9	22
MGM Grand Hotel	10	28	3	6
Tropicana Las Vegas	11	1	12	17
NYNY Vegas	12	24	14	12
Golden Nugget	13	31	32	29
The Cosmopolitan	14	23	10	5
Mandalay Bay Resort	15	17	7	11
Paris Las Vegas	16	4	29	14
Treasure Island	18	10	26	24
Caesars Palace	17	12	18	13
Monte Carlo Resort	19	8	22	21
Planet Hollywood	20	30	20	26
Bally's Las Vegas	24	33	27	34
Stratosphere Hotel	22	13	8	16
Palms Casino Resort	21	21	1	7
Hard Rock Hotel LV	23	32	15	27
Harrah's Las Vegas	25	29	31	30
Luxor Hotel & Casino	26	11	11	15
Circus Circus	27	18	23	25
Excalibur Las Vegas	28	19	28	18
Rio Las Vegas	29	27	13	28
Flamingo Las Vegas	30	20	33	31
Hooters Casino Hotel	31	26	30	32
Riviera Las Vegas	32	22	19	10
LVH Hotel & Casino	33	16	25	20
The Quad Las Vegas	34	34	34	33

3. Ratio Score Computation Using Probabilities

We propose a new method for calculating the ratio score of hotels, as described in the proposal. Using this method, we repeat the average measurements conducted previously, using again an average of 25 iterations. This method of calculating the ratio score will only be applied to our model as it is the one with the best ratio score results.

We do not take into account the validation accuracy and time since this change only affects the Spearman coefficient and the rest remains unchanged. We prefer to focus on the ratio score of the hotels as it is the measurement we use to compare ourselves with both TripAdvisor and the approach by Philander & Zhong [8], and for practical purposes, it is a very good indicator for a hotel to know its evaluation.

TABLE VII. Comparison of Spearman Coefficients With Different Ratio Score Calculations

System	Spearman ρ
Philander & Zhong [8]	0.5010
Our model (tweets, regular method)	0.4713
Our model (Datafiniti, regular method)	0.5763
Our model (tweets, new method)	0.5311
Our model (Datafiniti, new method)	0.6106

As can be observed in Table VII, our new method of computing the ratio score improves the average results obtained, especially for the tweet dataset. As we did before, we also obtain the average ratio scores by means of this new method for each hotel, obtaining the results shown in Table VIII.

TABLE VIII. New Ratio Score of Hotels by Each Model

	TripAdvisor	Philander & Zhong	Our model (tweets)	Our model (Datafiniti)
Palazzo Las Vegas	88	7.04	21.3194	6.7282
Bellagio Las Vegas	88	9.6	16.9642	7.2440
M Resort Spa Casino	87	5.96	15.4200	6.5445
Red Rock Las Vegas	85	10.22	22.1321	6.4168
Venetian Las Vegas	85	8.66	22.6369	6.4919
Aria Las Vegas	84	10.72	15.5908	6.9443
South Point Hotel	84	5.88	17.3179	5.3535
The Mirage	84	4.9	19.4905	6.2074
Wynn Las Vegas	84	6.79	14.4221	5.7305
MGM Grand Hotel	81	4.41	17.9841	5.4712
Tropicana Las Vegas	80	11.1	18.5516	7.5564
NYNY Vegas	79	5.08	16.8442	5.8826
Golden Nugget	79	3.13	8.8946	4.9302
The Cosmopolitan	77	5.2	18.5545	6.0684
Mandalay Bay Resort	74	5.78	17.2724	5.8667
Paris Las Vegas	73	9.9	12.3846	5.9012
Caesars Palace	72	6.08	14.5638	6.5781
Treasure Island	72	6.66	13.3477	5.4290
Monte Carlo Resort	70	6.81	13.6617	5.9256
Planet Hollywood	68	3.39	12.8227	5.2006
Palms Casino Resort	66	5.24	22.1213	5.4528
Stratosphere Hotel	66	6.05	19.8981	5.5520
Hard Rock Hotel L	64	2.64	15.4029	4.8764
Bally's Las Vegas	68	2.62	10.3494	3.8094
Harrah's Las Vegas	63	4.03	10.1939	4.7356
Luxor Hotel & Casino	60	6.1	17.3746	5.6681
Circus Circus	60	5.68	14.2205	5.6515
Excalibur Las Vegas	60	5.53	11.1029	5.5503
Rio Las Vegas	59	4,61	13,4598	4,8241
Flamingo Las Vegas	55	5,42	9,7044	4,7637
Hooters Casino Hotel	55	4,64	10,4948	4,1472
Riviera Las Vegas	48	5,23	14,9554	6,4636
LVH Hotel & Casino	47	5,86	15,1780	5,4890
The Quad Las Vegas	43	2,57	4.3803	4,2604

Lastly, we compute the Spearman coefficient with these new ratio scores, obtaining 0.57978 for the tweet dataset and 0.64121 for the Datafiniti dataset [48] . With these results, we can conclude that our model performs much better in terms of accurately predicting the relative positiveness of opinions about hotels. Table IX offers a visualization of the rankings obtained by different models compared to the actual TripAdvisor ranking.

TABLE IX. New Ranking of Hotels by Each Model

	TripAdvisor	Philander & Zhong	Our model (tweets)	Our model (Datafiniti)
Palazzo Las Vegas	1	7	4	4
Bellagio Las Vegas	2	5	13	2
M Resort Spa Casino	3	14	16	6
Red Rock Las Vegas	4	3	2	9
Venetian Las Vegas	5	6	1	7
Aria Las Vegas	6	2	15	3
South Point Hotel	7	15	11	25
The Mirage	8	25	6	10
Wynn Las Vegas	9	9	21	16
MGM Grand Hotel	10	28	9	22
Tropicana Las Vegas	11	1	8	1
NYNY Vegas	12	24	14	14
Golden Nugget	13	31	33	27
The Cosmopolitan	14	23	7	11
Mandalay Bay Resort	15	17	12	15
Paris Las Vegas	16	4	27	13
Caesars Palace	17	12	20	5
Treasure Island	18	10	25	24
Monte Carlo Resort	19	8	23	12
Planet Hollywood	20	30	26	26
Palms Casino Resort	21	21	3	23
Stratosphere Hotel	22	13	5	19
Hard Rock Hotel LV	23	32	17	28
Bally's Las Vegas	24	33	30	34
Harrah's Las Vegas	25	29	31	31
Luxor Hotel & Casino	26	11	10	17
Circus Circus	27	18	22	18
Excalibur Las Vegas	28	19	28	20
Rio Las Vegas	29	27	24	29
Flamingo Las Vegas	30	20	32	30
Hooters Casino Hotel	31	26	29	33
Riviera Las Vegas	32	22	19	8
LVH Hotel & Casino	33	16	18	21
The Quad Las Vegas	34	34	34	32

C. Complexity Results

Table X shows the average execution time of every model. As we can observe, our model obtains again the best results with an average of 19.28 seconds for the model trained with our tweets [8] and an average of 124.94 seconds for our model trained with Datafiniti [48] , while the DistilBERT model [6] takes 2092 seconds for training with our tweets and 2228 seconds for training with the Amazon dataset [49].

TABLE X. Average Run Time

Model	Time (seconds)
DistilBERT (Amazon)	2228
DistilBERT (tweets)	2092
Our model (tweets)	19.28
Our model (datafiniti)	124.94

This gap between the models is due to the fact that the complexity of our proposed model in terms of number of parameters is a lot lower than that of general-purpose pre-trained models. Models like DistilBERT [6] greatly rely on running on CUDA-compatible graphics, which makes training on personal computer CPUs or small servers ridiculously slow even for very small datasets like our case.

As we can see in Table XI, which compares our model with some of the most used models. This comparison measures the number of parameters that each neural network uses to train each model, as we can see our model has only 0.2665 million parameters, compared to the 530000 million of Megatron-Turing NLG [7]. This is because we saw that we do not need such a complex network for our problem. Thanks to this, our execution time is negligible compared to the rest of the models.

TABLE XI. Number of Trainable Parameters in Each Model

Model	Number of parameters (in millions)
GPT [56]	110
GPT-2 [57]	1500
GPT-3 [47]	175000
BERT base [5]	110
BERT large [5]	340
DistilBERT [6]	66
Megatron-Turing NLG [7]	530000
Our model	0.2665

V. Conclusions and Future Work

We have created a simple attention-based neural network model following the Transformer architecture and applied it to the problem of analyzing the sentiment in tweets about hotels.

We can conclude that our model, despite a low number of parameters of 266500, obtains a more accurate ranking score for the hotels than both the approach by Philander & Zhong and a big pre-trained model like DistilBERT, measured by the Spearman correlation coefficient. Both training datasets, tweets and Datafiniti reviews, manage to improve the results obtained by every other option, with the more extensive dataset of Datafiniti reviews achieving superior results that the tweets dataset.

In terms of validation accuracy on our own dataset, however, the DistilBERT model achieves slightly superior performance but is greatly outperformed in terms of training and execution time for processing the same dataset in the same system.

After reviewing the results obtained, we can conclude that for the problem of classifying the positivity of opinions about hotels in tweets, a very specific problem, using a neural network model based on Transformers with few parameters and simple layers is a better alternative than some basic NLP approaches and than complex pre-trained models.

As options for future work, we propose the following areas of research. Using other external datasets or increasing the number of classified tweets, might report more accurate results, since we have found that the tweet dataset performs considerably worse than the bigger review dataset. The model could also be trained with a dataset from Facebook, Instagram, or any other social network.

For the dataset used and our aim of efficiency, we believe to have reached a refined model configuration. However, increasing the model learning capabilities at the cost of greater complexity is bound to report more accurate results, as the results have shown that the DistilBERT model does.

In a similar way, attempting to implement convolutional layers, recurrent layers or other systems for text classification is also an area that could be explored. Attempting to train a different model to include the analysis of linked pictures to support the classification of the text or taking threads into account could prove especially beneficial when

dealing with tweets. For improving on the TripAdvisor comparisons, training with reviews from TripAdvisor itself should report greater ranking accuracy than training with comments from a completely different site.

The problem could also be updated to current times, obtaining the latest tweets tagging the hotel accounts and updating the TripAdvisor scores and ranking to reflect that of the current date. This could be done by means of a Domain Specific Language to automatically extracts tweets and generate a new dataset.

Acknowledgment

This research was funded by Fundación Universidad de Oviedo grant numbers PE.065.21 & PE.066.21.

References

[1] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980, doi: 10.1007/BF00344251.

[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.

[3] D. W. Otter, J. R. Medina, and 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, Feb. 2021, doi: 10.1109/TNNLS.2020.2979670.

[4] A. Vaswani *et al.*, "Attention is All you Need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[5] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 4171–4186, Oct. 2018.

[6] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, Oct. 2019.

[7] S. Smith *et al.*, "Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model," *arXiv preprint arXiv:2201.11990*, 2022.

[8] K. Philander and Y. Y. Zhong, "Twitter sentiment analysis: Capturing sentiment from integrated resort tweets," *International Journal of Hospitality Management*, vol. 55, pp. 16–24, May 2016, doi: 10.1016/J.IJHM.2016.02.001.

[9] S. Barke, R. Kunkel, N. Polikarpova, E. Meinhardt, E. Baković, and L. Bergen, "Constraint-based Learning of Phonological Processes," *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 6176–6186, 2019, doi: 10.18653/V1/D19-1639.

[10] O. Güngör, T. Güngör, and S. Uskudarli, "EXSEQREG: Explaining sequence-based NLP tasks with regions with a case study using morphological features for named entity recognition," *PLoS One*, vol. 15, no. 12, Dec. 2020, doi: 10.1371/journal.pone.0244179.

[11] E. M. Ponti, A. Korhonen, R. Reichart, and I. Vulić, "Isomorphic transfer of syntactic structures in cross-lingual NLP," *56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, vol. 1, pp. 1531–1542, 2018, doi: 10.18653/V1/P18-1142.

[12] C. Hutto, E. G.-P. of the international A. conference on, and undefined 2014, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," *Proceedings of the international AAAI conference on web and social media*, vol. 8, no. 1, pp. 216–225, 2014

[13] P. Chikersal, S. Poria and E. Cambria, "SeNTU: Sentiment Analysis of Tweets by Combining a Rule-based Classifier with Supervised Learning," *Proceedings of the 9th international workshop on semantic evaluation*, pp. 647–651, 2015.

[14] F. Wunderlich and D. Memmert, "Innovative Approaches in Sports Science—Lexicon-Based Sentiment Analysis as a Tool to Analyze Sports-Related Twitter Communication," *Applied Sciences*, vol. 10, no. 2, p. 431, Jan. 2020, doi: 10.3390/APP10020431.

[15] S. Lai, L. Xu, K. Liu and J. Zhao, "Recurrent convolutional neural networks for text classification," *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.

[16] H. Kim and Y. S. Jeong, "Sentiment Classification Using Convolutional Neural Networks," *Applied Sciences*, vol. 9, no. 11, p. 2347, Jun. 2019, doi: 10.3390/APP9112347.

[17] X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data*, vol. 2, no. 1, pp. 1–14, Dec. 2015, doi: 10.1186/S40537-015-0015-2/FIGURES/9.

[18] M. Imran, P. Mitra, and C. Castillo, "Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages," *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pp. 1638–1643, May 2016, doi: 10.48550/arxiv.1605.05894.

[19] X. Liu, H. Shin, and A. C. Burns, "Examining the impact of luxury brand's social media marketing on customer engagement: Using big data analytics and natural language processing," *Journal of Business Research*, vol. 125, pp. 815–826, Mar. 2021, doi: 10.1016/J.JBUSRES.2019.04.042.

[20] F. Z. Xing, E. Cambria, and R. E. Welsch, "Natural language based financial forecasting: a survey," *Artificial Intelligence Review*, vol. 50, no. 1, pp. 49–73, Oct. 2017, doi: 10.1007/S10462-017-9588-9.

[21] M. G. Huddar, S. S. Sannakki, and V. S. Rajpurohit, "Attention-based multi-modal sentiment analysis and emotion detection in conversation using RNN," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 6, pp. 112–121, 2021, doi: 10.9781/ijimai.2020.07.004.

[22] P. Dcunha, "Aspect Based Sentiment Analysis and Feedback Ratings using Natural Language Processing on European Hotels," *Doctoral thesis. Dublin, National College of Ireland*, 2019.

[23] T. Ghorpade and L. Ragha, "Featured based sentiment classification for hotel reviews using NLP and Bayesian classification," *Proceedings - 2012 International Conference on Communication, Information and Computing Technology*, 2012, doi: 10.1109/ICCICT.2012.6398136.

[24] B.-Ş. Posedaru, T.-M. Georgescu, and F.-V. Pantelimon, "Natural Learning Processing based on Machine Learning Model for automatic analysis of Online Reviews related to Hotels and Resorts," *Database Systems Journal*, vol. 11, no. 1, pp. 86–105, 2020.

[25] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, Dec. 2014, doi: 10.1016/J.ASEJ.2014.04.011.

[26] L. C. Yu, J. L. Wu, P. C. Chang, and H. S. Chu, "Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news," *Knowledge-Based Systems*, vol. 41, pp. 89–97, Mar. 2013, doi: 10.1016/J.KNOSYS.2013.01.001.

[27] M. Hagenau, M. Liebmann, and D. Neumann, "Automated news reading: Stock price prediction based on financial news using context-capturing features," *Decision Support Systems*, vol. 55, no. 3, pp. 685–697, Jun. 2013, doi: 10.1016/J.DSS.2013.02.006.

[28] I. Maks and P. Vossen, "A lexicon model for deep sentiment analysis and opinion mining applications," *Decision Support System*, vol. 53, no. 4, pp. 680–688, Nov. 2012, doi: 10.1016/J.DSS.2012.05.025.

[29] J. Wang *et al.*, "Systematic Evaluation of Research Progress on Natural Language Processing in Medicine Over the Past 20 Years: Bibliometric Study on PubMed," *Journal of Medical Internet Research*, vol. 22, no. 1, Jan. 2020, doi: 10.2196/16816.

[30] A. Alsudais, G. Leroy, and A. Corso, "We know where you are tweeting from: Assigning a type of place to tweets using natural language processing and random forests," *Proceedings - 2014 IEEE International Congress on Big Data*, pp. 594–600, Sep. 2014, doi: 10.1109/BIGDATA.CONGRESS.2014.91.

[31] Y. Goldberg and M. E. Ben, "splitSVM: Fast, Space-Efficient, non-Heuristic, Polynomial Kernel Computation for NLP Applications," *Association for Computational Linguistics*, pp. 237–240, Jun. 2008.

[32] C. Bartz, T. Herold, H. Yang, and C. Meinel, "Language Identification Using Deep Convolutional Recurrent Neural Networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10639, pp. 880–889, 2017, doi: 10.1007/978-3-319-70136-3_93.

[33] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," *2010 IEEE International Symposium on Circuits and*

Systems: Nano-Bio Circuit Fabrics and Systems, pp. 253–256, doi: 10.1109/ISCAS.2010.5537907.

[34] A. Conneau, H. Schwenk, Y. le Cun, and L. Lo¨ıc Barrault, "Very Deep Convolutional Networks for Text Classification," *Nature*, pp. 1–11, Jun. 2016, doi: 10.48550/arxiv.1606.01781.

[35] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/NECO.1997.9.8.1735.

[36] T. Wang, P. Chen, K. Amaral, and J. Qiang, "An Experimental Study of LSTM Encoder-Decoder Model for Text Simplification," *arXiv preprint arXiv:1609.03663*, Sep. 2016.

[37] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches," *8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, Sep. 2014, doi: 10.3115/v1/w14-4012.

[38] Z. Shaheen, G. Wohlgenannt, and E. Filtz, "Large Scale Legal Text Classification Using Transformer Models," *Computer Science ArXiV*, vol. abs/2010.12871, 2020.

[39] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *3rd International Conference on Learning Representations*, Sep. 2015, doi: 10.48550/arxiv.1409.0473.

[40] T. Shao, Y. Guo, H. Chen, and Z. Hao, "Transformer-Based Neural Network for Answer Selection in Question Answering," *IEEE Access*, vol. 7, pp. 26146–26156, 2019, doi: 10.1109/ACCESS.2019.2900753.

[41] U. Khandelwal, K. Clark, D. Jurafsky, and Ł. Kaiser, "Sample Efficient Text Summarization Using a Single Pre-Trained Transformer," *arXiv preprint arXiv:1905.08836*, 2019.

[42] T. Wang, X. Wan, and H. Jin, "Amr-to-text generation with graph transformer," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 19–33, Jan. 2020, doi: 10.1162/TACL_A_00297/43537/AMR-TO-TEXT-GENERATION-WITH-GRAPH-TRANSFORMER.

[43] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 2, pp. 107–116, Apr. 1998, doi: 10.1142/S0218488598000094.

[44] A. Gulati *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition," *Proceedings of the Annual Conference of the International Speech Communication Association*, vol. 2020-October, pp. 5036–5040, 2020, doi: 10.21437/Interspeech.2020-3015.

[45] A. Rives *et al.*, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 118, no. 15, Apr. 2021, doi: 10.1073/PNAS.2016239118/SUPPL_FILE/PNAS.2016239118.SAPP.PDF.

[46] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A Survey of Transformers," *OpenAI*, Jun. 2021.

[47] T. Brown *et al.*, "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 1877–1901.

[48] "Hotel Reviews - dataset by datafiniti | data.world." https://data.world/datafiniti/hotel-reviews (accessed Feb. 11, 2022).

[49] "amazon_reviews_multi · Datasets at Hugging Face." https://huggingface.co/datasets/amazon_reviews_multi (accessed Feb. 11, 2022).

[50] "Hotel Reviews - dataset by datafiniti." https://data.world/datafiniti/hotel-reviews (accessed Mar. 23, 2022).

[51] A. Joshi, S. Kale, S. Chandel, and D. K. Pal, "Likert Scale: Explored and Explained," *British Journal of Applied Science & Technology*, vol. 7, no. 4, p. 396, 2015, doi: 10.9734/BJAST/2015/14975.

[52] François Chollet, "Keras: the Python deep learning API," *Astrophysics Source Code Library*, 2018.

[53] A. Belhadi, Y. Djenouri, J. C. W. Lin, and A. Cano, "A data-driven approach for twitter hashtag recommendation," *IEEE Access*, vol. 8, pp. 79182–79191, 2020, doi: 10.1109/ACCESS.2020.2990799.

[54] K. Philander and Y. Y. Zhong, "Twitter sentiment analysis: Capturing sentiment from integrated resort tweets," *International Journal of Hospitality Management*, vol. 55, pp. 16–24, May 2016, doi: 10.1016/J.IJHM.2016.02.001.

[55] "Natural Language Processing with Transformers [Book]." https://www.oreilly.com/library/view/natural-language-processing/9781098103231/ (accessed Feb. 11, 2022).

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

[57] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," *OpenAI*, vol. 1, no. 8, p. 9, 2019.

Sergio Arroni

Sergio Arroni is a student of Software Engineering at the University of Oviedo, passionate about Machine Learning, Artificial Intelligence and NLP among other Software fields. He is currently working at the Foundation of the University of Oviedo, researching new advances in the field of Artificial Intelligence and Machine Learning.

Yeray Galán

Yeray Galán is a Software Engineering Graduate of the University of Oviedo and a Master's Degree in Mathematical Research and Modelling, Statistics and Computing student at the University of the Basque Country and the University of Zaragoza, currently doing research at Fundación Universidad de Oviedo on artificial intelligence, particularly machine learning, and Monte Carlo methods.

Xiomarah Guzmán-Guzmán

Xiomarah Guzmán-Guzmán is an Interim Professor and Ph.D. candidate in the Department of Computer Science at the University of Oviedo (Spain). She has a Master's in Website Management and Engineering from the International University of La Rioja, and B.S. in Computer Science from the Technological University of Santiago. She has published some articles in international journals and conferences. Her research interests include artificial intelligence, recommendation systems and machine learning.

Edward Rolando Núñez-Valdez

Edward Rolando Núñez-Valdez is an associate professor in the Department of Computer Science at the University of Oviedo (Spain). He has a Ph.D. in Computer Engineering from the University of Oviedo, a Master's in Software Engineering from the Pontifical University of Salamanca and a B.S. in Computer Science from the Autonomous University of Santo Domingo. He has participated in several research projects; he has taught computer science at various schools and universities, and he has worked in software development companies and IT consulting for many years. He has published several articles in international journals and conferences. His research interests include artificial intelligence, recommendation systems, decision support systems, health informatics, modeling software with DSL and MDE.

Alberto Gómez

Alberto Gómez works for the Department of Business Administration, at the School of Industrial Engineering of The University of Oviedo, Spain. His teaching and research initiatives focus on the areas of Production Management, Applied Artificial Intelligence and Information Systems. He has written several national and international papers. Journal of the Operational Research Society. Artificial Intelligence for Engineering Design, Analysis and Manufacturing. International Journal of Foundations of Computer Science. European Journal of Operational Research. International Journal of production economics. Engineering Applications of Artificial Intelligence. Concurrent Engineering- Research and Applications.

A Spatio-Temporal Attention Graph Convolutional Networks for Sea Surface Temperature Prediction

Desheng Chen, Jiabao Wen*, Caiyun Lv

School of Electrical and Information Engineering, Tianjin University, Tianjin (P.R. China)

Received 16 June 2022 | Accepted 10 February 2023 | Early Access 23 February 2023

Abstract

Sea surface temperature (SST) is an important index to detect ocean changes, predict SST anomalies, and prevent natural disasters caused by abnormal changes, dynamic variation of which have a profound impact on the whole marine ecosystem and the dynamic changes of climate. In order to better capture the dynamic changes of ocean temperature, it's vitally essential to predict the SST in the future. A new spatio-temporal attention graph convolutional network (STAGCN) for SST prediction was proposed in this paper which can capture spatial dependence and temporal correlation in the way of integrating gated recurrent unit (GRU) model with graph convolutional network (GCN) and introduced attention mechanism. The STAGCN model adopts the GCN model to learn the topological structure between ocean location points for extracting the spatial characteristics from the ocean position nodes network. Besides, capturing temporal correlation by learning dynamic variation of SST time series data, a GRU model is introduced into the STAGCN model to deal with the prediction problem about long time series, the input of which is the SST data with spatial characteristics. To capture the significance of SST information at different times and increase the accuracy of SST forecast, the attention mechanism was used to obtain the spatial and temporal characteristics globally. In this study, the proposed STAGCN model was trained and tested on the East China Sea.Experiments with different prediction lengths show that the model can capture the spatio-temporal correlation of regional-scale sea surface temperature series and almost uniformly outperforms other classical models under different sea areas and different prediction levels, in which the root mean square error is reduced by about 0.2 compared with the LSTM model.

Keywords

Data Forecast, Gated Recurrent Unit, Graph Convolutional Network, Sea Surface Temperature, Spatiotemporal Attention Graph Convolutional Network.

DOI: 10.9781/ijimai.2023.02.011

I. Introduction

The dynamics of the oceans, which make up about two-thirds of the planet, have an extremely important impact on climate, marine ecology, and the lives of the people around them. Sea surface temperature (SST) is an important index to detect ocean changes, predict SST anomalies, and prevent natural disasters caused by abnormal changes. Therefore, it's significant to predict the dynamic change of SST in the future. Moreover, SST has played an indispensable role in the ocean-atmosphere interaction, that is, the exchange of matter, energy, and momentum between the ocean and the atmosphere [1] [2] [3]. As a result, changes in SST have incalculable impacts on global climate and marine ecosystem [4] [5] [6] [7]. Besides, SST predictions also has implications for applications related to the ocean, such as weather forecasting, fisheries, and marine environmental protection. Therefore, it's critically necessary to predict dynamic changes of SST in the future to help people identify and prevent severe weather events such as drought in advance [8] [9], and it's also of great significance for scientific research and applications [10]. However, due to the influence of many complex factors, such as sea surface heat flow, radiation, and solar wind, the prediction of SST is quite indefinite and challenging.

In recent years, SST prediction methods have been widely applied and attracted much attention further become an attractive field of marine research. Three kinds of methods are generally used to predict SST including the numerical method based on the mathematical model, the data-driven method using the historical model to predict SST in the future, and the method combining the two methods [11]. The numerical methods generally use kinetic and thermodynamic equations to exposit the dynamic changes of SST and then solve a series of differential equations which are difficult to solve due to they are usually sophisticated and require a large amount of computation. The data-driven method is mainly used to predict the future SST value from the perspective of data. This method builds the model by learning the relationships and patterns from the historical SST observation data and further uses the learned relationships model to approximate the future SST data. The data-driven method is less complicated than the numerical method and is suitable for the prediction of SST in high-resolution areas. The data-driven method mainly predicts the future SST from the perspective of statistical data analysis, machine learning, and artificial intelligence algorithms. Among them, statistical data analysis techniques primarily contain the Markov model [12] [13], Empirical canonical correlation analysis [14] and regression model

* Corresponding author.
E-mail address: Wen_Jiabao@tju.edu.cn

[15] [16], etc. Classical machine learning methods including linear regression, support vector machine (SVM)[[17],nonlinear regression model [18], and artificial neural network [19] are used to forecast future SST. Support vector machine (SVM) is a generalized linear classifier that classifies data based on supervised learning. Particle swarm optimization (PSO) algorithm is a random and parallel optimization algorithm based on population. Those two kinds of artificial intelligence methods are commonly used in SST prediction [17] [20]. The numerical method and machine learning method can also be combined [11] to better predict SST, but the prediction effect is similar to that using the numerical method.

With the continuous development and innovation of deep learning, the deep learning method has been mostly used in SST prediction due to its powerful ability to learn and model the relationship between data [21] [22] [23] [24]. Recurrent neural network (RNN) can effectively deal with time series prediction problems, but serious gradient vanishing or outbreak problems will occur when processing long time series data. As a variant of the recurrent neural network, a long short-term memory (LSTM) network with recurrent structure and gating mechanism is proposed to solve the long-term time dependence problem, which can remember longer time series information and obtain better prediction results [25]. In order to simplify the complex structure of LSTM network, a gate recurrent unit (GRU) [26]with relatively unsophisticated gate structure was proposed. As an improved variant of the LSTM network, the GRU model not only retains the advantage of LSTM in long-term series memory but also has high computational efficiency, which alleviated the phenomenon of network overfitting and underfitting.Those model have been widely used in ocean surface temperature prediction [27] proposed a full-connected LSTM (FC-LSTM), which is composed of LSTM layer and Full Connected layer. [26]designs an adaptive mechanism based on deep learning and attention mechanism to predict SST, which uses GRU encoder-decoder to obtain the static change of SST and apply dynamic influence link to acquire the dynamic variation for realizing the long-term prediction of future SST. [9]proposes an integrated learning model (LSTM-Adaboost) that combines the deep LSTM neural network with the Adaptive Boosting (AdaBoost) algorithm to predict the daily SST in the short and medium-term. Feng et al used time-domain convolutional network to achieve short-term small-scale SST prediction of the Indian Ocean [28]. Han et al. used convolutional neural network method to achieve regional prediction of Sea surface temperature, sea surface height and ocean salinity in the Pacific [29].

However, although these SST prediction models have achieved a good prediction effect, they only consider the time correlation but ignore the spatial dependency, so they cannot achieve high accuracy in predicting the dynamic changes of SST sequence data. Beside, the association structures constructed when capturing the spatial influence of adjacent nodes on the central node are not all standardized grid structures. For example, topology structures based on spatial association can be constructed when missing values exist. Therefore, in order to capturing spatial dependencies from complex topologies, the GCN was applied to obtain the spatial dependence from the SST series data of the ocean location points, an original SST prediction method named spatio-temporal attention graph convolutional network (STAGCN) based on ocean location points network was proposed in this paper. Specifically, the GCN is applied to capture spatial correlation from the ocean positions network with the topological structure. The GRU is used to capturing the temporal dependence from the dynamic changes of the SST time series data. In addition, the STAGCN model introduces an attention mechanism to learn global correlation, adjust and integrate global temporal information of SST for realizing accurate SST prediction tasks eventually.

The contribution of this paper can be summarized in the following two aspects: (1) A STAGCN model is designed to capture the global spatial and temporal dependence simultaneously for accurate SST prediction, which combines the GCN deep learning model with the GRU learning model and introduces an attention mechanism. (2) The concept of a graph is applied to the field of SST prediction, and the topology structure network of ocean location points graph is constructed to obtain the spatial characteristics from SST time series by GCN model. Using 38-year time-series satellite data from some areas of the East China Sea, the experiments show that the STAGCN model can achieve preferable prediction results than the GRU model and the GCN model demonstrating that the STAGCN model has the ability to capture both time and space correlation from SST series data simultaneously, and can achieve desirable prediction effect for the short-term prediction of future SST.

The rest of the paper is organized as follows. Section II proposes the novel STAGCN model for SST prediction in detail. Section III describes the experiments and analyzes the results. Finally, Section IV gives the conclusion of the paper.

II. Method

A. Problem Clarification

In this study, the prediction of SST is to predict the sea surface temperature within a certain time in the future according to the historical SST time-series information.

The undirected graph G = (V, E) with no weights was used to describe a topological network composed of oceanic observation points, and each location represents a node in the graph, where $V = v_1, v_2, ..., v_N$ means all oceanic positions that correspond to the N vertices of the graph. E represents the correlation between ocean points corresponding to the edges between nodes in the graph, reflecting the connection between nodes at different positions. The connection relationship between nodes in the whole graph is represented by the adjacencies matrix A, the number of the rows and columns of which are determined by the number of nodes, and each value of which represents the connection relationship between nodes. The value of each item in the adjacency matrix A is either 0 or 1. 0 indicates that there is no direct connection between nodes and 1 expresses that there has a linkage between nodes.

The feature X^{P*N} of the node in the topology graph corresponds to the SST value of each location point on the ocean positions network. where N represents the length of the time dimension in oceanographic satellite data, and $X_t \in R^{t*N}$ represents the SST value at time t.

The SST prediction problem can be regarded as looking for a mapping function f, which map the SST value in the historical time n to the SST value in the next T periods under the conditions of the topology diagram G of the ocean location points network and feature matrix X. Equation (1) refers to the SST prediction process:

$$(X_{t-n}, \ldots, X_{t-1}, X_t) \quad f(\bullet)[X_{t+1}, X_{t+2}, \ldots, X_{t+T}] \tag{1}$$

The left side is the historical SST value with the length n, from which the model learns the variation trend of SST, and the right side of the equation is the predicted future SST value with length T under the mapping condition.

B. The STAGCN Model

To capture global dynamic changes of the SST information, a novel STAGCN model combines graph convolutional network capturing spatial correlation and gated recurrent unit obtaining temporal dependence with attention model at the same time is proposed in this paper.In the STAGCN model, a layer of GCN network can be used to

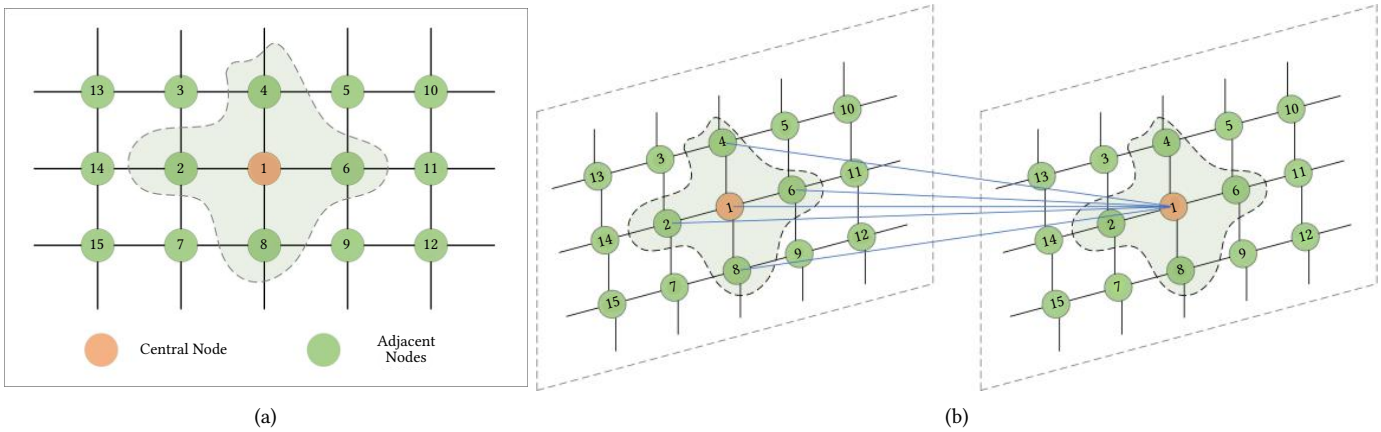


Fig. 1. Topology graph of an ocean location point (in the dotted shaded part). (a) The green nodes indicate the location points have connection relationship with the central point. (b) The topological structure between the central node 1 and its adjacent nodes is established through the GCN model to obtain the spatial features.

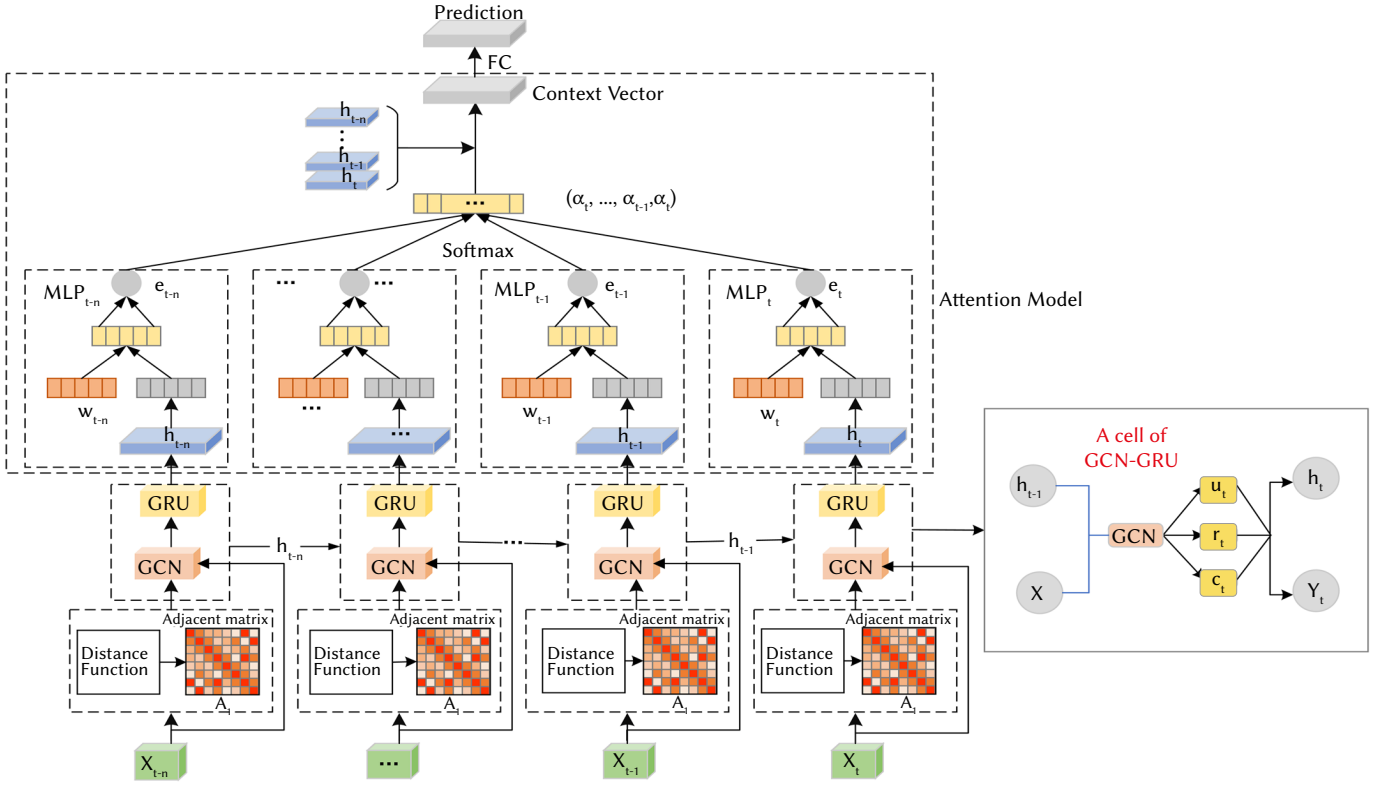


Fig. 2. The specific spatial-temporal prediction process of the STAGCN model.

obtain a better prediction effect, and Equation (2) refers to the specific convolution process:

$$f(X, A) = \sigma(\overline{A}ReLU(\overline{A}XW_0)W_1) \tag{2}$$

where $f(X, A)$ express the final output of the GCN model, X is the eigenmatrix, A means the adjacency matrix of the graph convolution. $ReLU(\cdot)$ represents an activation function used to add linear factors to improve model expression.is the form of the adjacent matrix after further renormalization to avoid causing gradient explosion. W_0 represents the weight that needs to be trained in the progress of the graph convolutional network, whose first dimension F means the time series length of ocean data and second dimension G means the number of neural units in the output layer.

Specifically, the area we study is ocean location points graphs composed of position points determined by longitude and latitude, and the topological connection relationship between location points can be captured by the GCN network. The shaded part of the dotted line in Fig. 1 is a topological graph of a point in the simulated ocean location. It is assumed that the red dot 1 represents the central node in the topology diagram, and the green dots (the green dot in the shaded part of the dotted line) around it are the adjacent nodes. The interaction degree between the central node and the adjacent nodes can be obtained by GCN model. Then the GCN model captures the SST characteristic attributes of the topological structure and further acquires the spatial correlation from the location points network. The GRU model was adopted in this study to obtain the temporal dependence of SST data. The attention model is used to screen which moments of SST data are relevant, that is, to distinguish the importance of data at different moments, which improves the accuracy of the prediction and realize the SST prediction task based on the structure of the ocean location points graph. The specific spatial-temporal prediction process of the STAGCN model is shown in Fig. 2, where the distance function (Equation) was applied to calculate the adjacency matrix A_1 that represents the connection relation between the position

points, the GCN-GRU represents the process of combining the output of the GCN model and the GRU model, and the FC express the fully connected layer.

First of all, the obtained adjacency matrix and the SST feature data X_i ($i = t-n, ..., t-1, t$) of n historical time series were taken as input, from which the GCN model capture the spatial information. Then, The input of the GRU model is replaced by the output of the GCN model to obtain the temporal characteristics of SST data. Equation (3) to (6) refer to the update gate, reset gate, cell state and output state at time t in the STAGCN model respectively.

$$u_t = \sigma(W_u \bullet f_{gcn}(A, [h_{t-1}, X_t]) + b_u) \tag{3}$$

$$r_t = \sigma(W_r \bullet f_{gcn}(A, [h_{t-1}, X_t]) + b_r) \tag{4}$$

$$c_t = tanh(W_c \bullet f_{gcn}(A, [(r_t * h_{t-1}), X_t]) + b_c) \tag{5}$$

$$h_t = (u_t * h_{t-1} + (1 - u_t) * c_t \tag{6}$$

where u_t and r_t express the update gate and reset gate, h_{t-1} means the output at previous time, h_t means the state output at present t moment, c_t represents the information be reserved from previous moment and present time. Function $f_{gcn}(\cdot)$ express the graph convolution, W_u, W_r and W_c are the connection weights between the output of graph convolution and the previous output h_{t-1}, b_u, b_r and b_c are the corresponding thresholds. Moreover, the final hidden state information of the GRU model is used as the input of the attention model, which is used to obtain the importance of the changes information of SST series data. Finally, we get the prediction from the full connection layer. In the attention model, multi-layer perception is used as the scoring function, in which w_i ($i = t-n, ..., t-1, t$) is the weight matrix of multi-layer perception. The score e_i ($i = t-n, ..., t-1, t$) of the multi-layer perception output is brought into the Softmax function to get the attention distribution probability. The last hidden state and its weight are weighted to obtain the final context vector C.

To sum up, we propose that the STAGCN model has the ability to obtain the global spatial dependence and temporal dynamic changes which can obtain a preferable SST prediction effect. The GCN model is applied to capture spatial information by building the structure of the interrelation between the position nodes. The GRU model is used to obtain the temporal dependence from the SST series data with spatial characteristics. Moreover, the attention model captures the global variation trends of the SST information which is significant to achieve accurate SST prediction tasks.

III. Experiments

A. Research Area and Data

Rich in natural resources, the East China Sea is the confluence of many rivers covering a wide area including China's Bohai Sea in the north and the Taiwan Strait in the south and is the strategic maritime area for China, Japan, South Korea, and other countries. Therefore, studying the dynamic changes of SST data in the East China Sea plays an extremely important role in national marine transportation and people's production and life in the surrounding countries. The research area selected in this study is the part area of the East China Sea with sea areas of 26.8755°N - 32.125°N and 123.125°E - 127.125°E covering most areas of the East China Sea and no land area, as is shown in Fig. 3. The selected area includes most of the East China Sea to facilitate the acquisition of grid-based SST data and the establishment of the topological structure of the ocean position points graph for SST prediction.

In this study, the data used in the experiments were derived from the daily Optimum interpolated SST (OISST Daily Edition 2.1) with a spatial resolution of 1/4° in the N ational Oceanic and Atmospheric

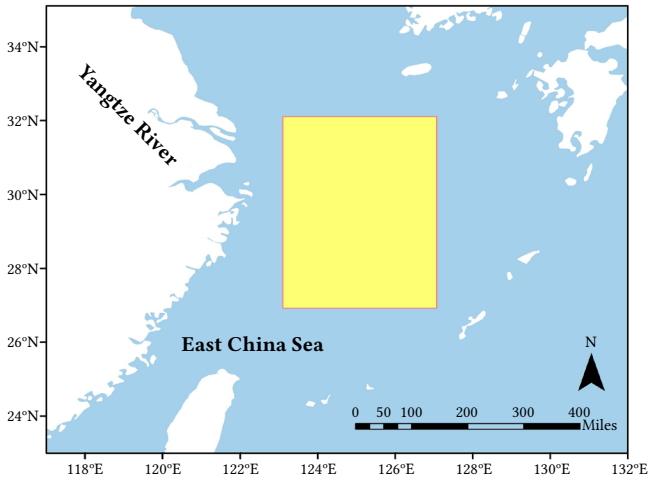


Fig. 3. Research area in the East China Sea (in yellow).

Administration (NOAA) platform. In this study, AVHRR-only daily SST data contained a total of 13,879 days SST data from 1982/01/01-2019/12/31 covered the spatial span of 26.875°N-32.125°N, 123.125°E-127.125°E (the most of the east sea) are used as the experimental dataset, with a total of 22*17 position points as the study nodes.

In this study, the experimental data is composed of two parts: the adjacency matrix and the eigenmatrix respectively. The former is an adjacency matrix of size 374*374, which depicts the spatial dependence between position points. Equation (7) refers to each value in the adjacency matrix W is derived by some scaling of the distance between each position point in the ocean anchor point network.

$$W_{ij} = \begin{cases} \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right), i \neq j \text{ and } \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) \geq \varepsilon \\ 0 \quad \text{, otherwise} \end{cases} \tag{7}$$

where W_{ij} is the weight of the edges in the position graph determined by the distance between position i and j (d_{ij}). σ^2 and ε are thresholds to control spacial arrangement and sparse arrangement of the adjacency matrix W, which are set to 0.1 and 0.4 respectively after testing in the experiments. The latter part of the experimental data is the eigenmatrix, which describes the dynamic changes of the SST value over time at the position nodes.

B. Experiment Setup

In order to better explain the superiority of the STAGCN model proposed in predicting SST, we chose five comparative models including the autoregressive moving average model (ARIMA), linear support vector machine model (SVR), graph convolutional network model (GCN), and gated recurrent unit model (GRU). In the experiments, we divided the dataset into the training set and test set, and the ratio of the two is 8:2. The Adam optimizer is used to train the model. The STAGCN model, GCN model, and GRU model use TensorFlow 1.5.0 (GPU version) as runtime environment during the training and testing progress. The ARIMA and SVR are respectively implemented using Statsmodels 0.12.2 and Scikit-learn 0.24.1 [31].

The real SST and the predicted SST of different nodes at time t are respectively expressed by Y_t and $\hat{Y}_t$. In the network training, the loss function value should be minimized as far as possible, which is beneficial for the predicted SST value of each ocean position node to be a better fit to the actual SST value. The loss function defined in the STAGCN model is shown in Equation. The first term $||\cdot||$ is the 2-norm of the real value and the predicted value, and the second term is the regularization term λL_{reg} with hyperparameter λ, which improved the prediction performance of the model and prevented overfitting occurrence during training.

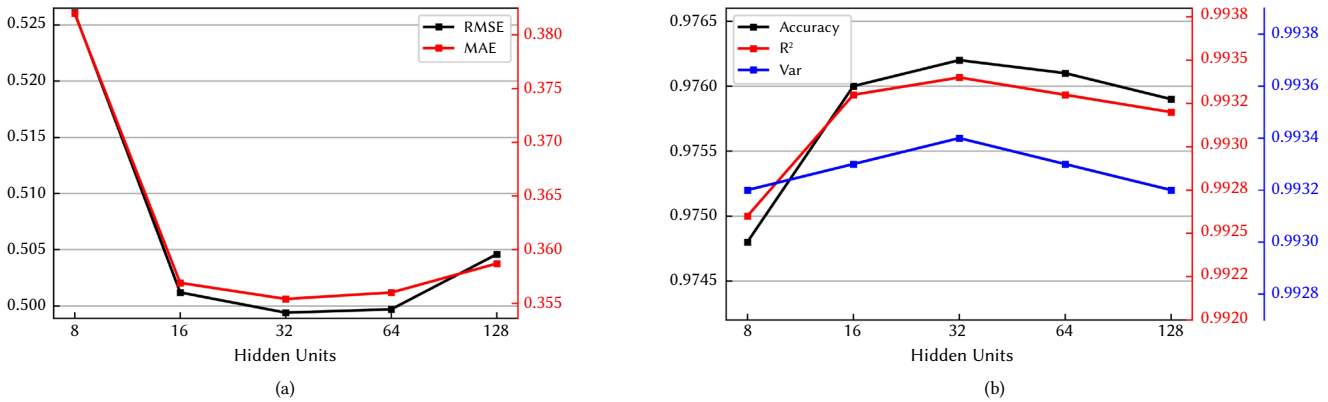


Fig. 4. Comparison of SST prediction results of the STAGCN model under different hidden units conditions. (a) Changes trend of RMSE and MAE under different hidden units conditions. (b) Variation trend in Accuracy, R^2 and Var under different hidden units conditions.

To prove the desirable prediction performance of the STAGCN model, five measurement criteria are used to compare the SST prediction performance of the proposed model with other models including root mean square error (RMSE), mean absolute error (MAE), accuracy (Accuracy), coefficient of determination (R^2) and explained variance score (Var).

$$RMSE = \sqrt{\frac{1}{TN}\sum_{t=1}^{T}\sum_{n=1}^{N}(y_n^t - \tilde{y}_n^t)^2} \tag{8}$$

$$MAE = \frac{1}{TN}\sum_{t=1}^{T}\sum_{n=1}^{N}|y_n^t - \tilde{y}_n^t| \tag{9}$$

$$Accuracy = 1 - \frac{||Y - \tilde{Y}||_F}{||Y||_F} \tag{10}$$

$$R^2 = 1 - \frac{\sum_{t=1}^{T}\sum_{n=1}^{N}\left(y_n^t - \hat{y}_n^t\right)^2}{\sum_{t=1}^{T}\sum_{n=1}^{N}\left(y_n^t - \overline{Y}\right)^2} \tag{11}$$

$$\text{var} = 1 - \frac{\text{var}\{Y - \hat{Y}\}}{\text{var}\{Y\}} \tag{12}$$

where Equation (8) to (12) refer to the calculation process of RMSE, MAE, Accuracy, (R^2) and (Var) respectively, T means the length of the SST time series, N represents the total number of position points. TN is the number of recorded changes in temperature values at all ocean locations with a time length of T. y_n^t represents the real SST value and the $\hat{y}_n^t$ means predicted SST value in the position point n at the moment t. The entire sets of y_n^t and $\hat{y}_n^t$ can be defined as Y and $\hat{Y}$ respectively. $\overline{Y}$ represents the average value of collection Y. Five criteria are used to measure the merits of the model from the perspective of error, accuracy, and fitting degree. The F in the accuracy indicator refers to the F-norm.

In the neural network model, the setting of model parameters is crucial to the model training, such as the number of neurons in the hidden layer, whose size determines the computational complexity and predictive performance of the whole model. Therefore, in order to improve the training efficiency and the accuracy of prediction, we used different hidden units to conduct experiments on the test set and select the corresponding number of hidden units with the best predicted effect from the predicted results. We set the number of neurons in the hidden layer as the value in [8, 16, 32, 64, 128], respectively. The variation trend of error indicators under different hidden units condition as shown in 4, from which we can see that the RMSE and MAE reach the minimum value when the number of hidden layer units is 32. At the same time, we can see from 4 that the prediction error including RMSE and MAE has a similar decreasing trend while hidden units increases from 8 to 32, and then it shows an increasing trend when hidden units exceed 32. On the contrary, the prediction accuracy has an opposite trend, rising first and then falling. The trend of the model measurement indexes shows that there is a critical value of the number of hidden layer cells in the model. When the critical value is exceeded, the complexity of the model will increase with the increase of the number of hidden layer cells, and the performance of the model will decline simultaneously. Since the prediction performance of the model has the best performance when the number of hidden units is 32, the following experiments will be carried out under the condition that the number of hidden units is set to 32.

C. Experiment Results and Discussion

We performed prediction experiments on 1 day, 7 days, 14 days, and 30 days SST values in the future and measure the performance of the proposed STAGCN model, the ARIMA model, SVR model, GCN model, and GRU model with five performance metrics. The SST prediction result of the STAGCN model is analyzed from the perspectives of prediction accuracy, temporal and spatial prediction ability, and long-term prediction ability.

Table I shows that the result of the five metrics using to measure the prediction result of the STAGCN model compared with other models on the East China Sea dataset for the 1-day, 7-days, 14-days, and 30-days prediction tasks, with boldface sections representing the optimal values of the various metrics of the model. - show that the value is negative, indicating that the model cannot predict well and can be ignored. Compared with other models, the proposed STAGCN model achieves excellent prediction performance under almost all conditions, indicating that the STAGCN model can capture the global spatio-temporal correlation, thus achieved an accurate prediction of SST. For different prediction lengths, the prediction effects of the STAGCN model are preferable to other models, demonstrating that the STAGCN model has the ability to predict SST accurately in both the short and long term.

As can be seen from Table I, the prediction performance of the GRU model is better than that of the GCN model. The RMSE of the GRU model is approximately 0.07 lower than that of the GCN model and the accuracy of the GRU model is increased by 0.4% compare with the GCN model for future one-day SST prediction. The forecasted effect of the GCN model is worse than that of the GRU model probably because the data itself has obvious time series features, but the GCN model merely captures the spatial dependency without considering the temporal characteristics from SST data.

TABLE I. The Prediction Result of the STAGCN Model Compared With Other Models

Time	metrics	ARIMA	SVR	GCN	GRU	LSTM	STAGCN
1 day	RMSE	7.2835	0.9280	0.5034	0.4217	0.5528	**0.3479**
	MAE	5.9413	0.7567	0.3574	0.2825	0.4316	**0.2354**
	Accuracy	0.6625	0.9588	0.9759	0.9799	0.9757	**0.9846**
	R^2	-	0.9670	0.9933	0.9953	0.9884	**0.9954**
	Var	-	0.9671	0.9933	0.9953	0.9910	**0.9954**
7 day	RMSE	7.9466	1.2042	0.8746	0.8034	0.7909	**0.6644**
	MAE	6.5981	0.9860	0.6450	0.5836	0.6077	**0.4907**
	Accuracy	0.6304	0.9464	0.9581	0.9615	0.9651	**0.9705**
	R^2	-	0.9447	0.9798	0.9830	0.9761	**0.9832**
	Var	0.0090	0.9477	0.9799	0.9831	0.9786	**0.9832**
14 day	RMSE	7.7821	1.3285	1.0467	0.9493	0.8941	**0.8122**
	MAE	6.4279	1.0855	0.7860	0.7086	0.6853	**0.6142**
	Accuracy	0.6386	0.9408	0.9497	0.9544	0.9604	**0.9640**
	R^2	-	0.9325	0.9711	0.9762	0.9691	0.9749
	Var	-	0.9351	0.9711	0.9762	0.9691	0.9749
30 day	RMSE	8.3632	1.5406	1.3253	1.1428	1.1605	**0.9694**
	MAE	7.0629	1.2533	1.0052	0.8662	0.8860	**0.7510**
	Accuracy	0.6128	0.9314	0.9364	0.9452	0.9492	**0.9568**
	R^2	-	0.9095	0.9538	0.9657	0.9469	0.9642
	Var	0.0040	0.9117	0.9539	0.9666	0.9471	0.9642

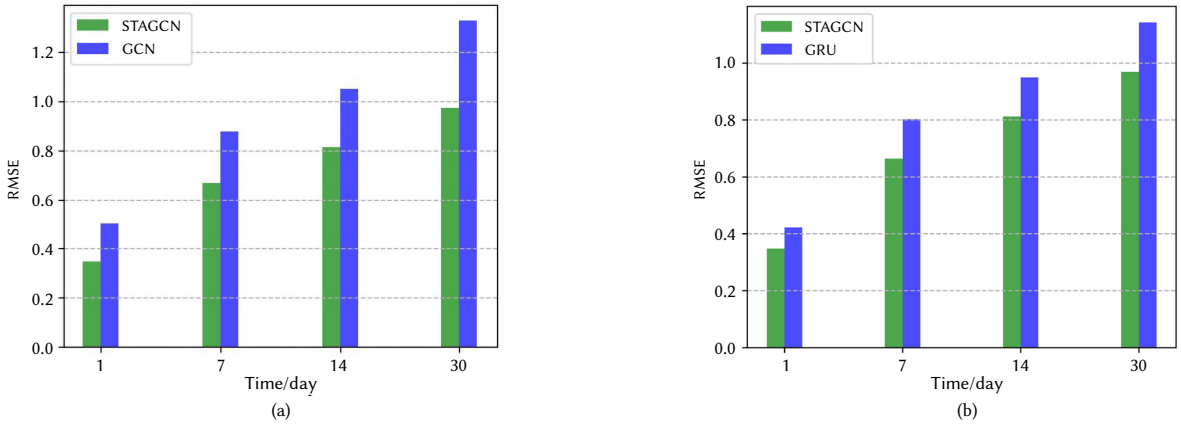


Fig. 5. Comparison of SST prediction errors between STAGCN model, GCN model and GRU model. (a) Comparison of RMSE values between STAGCN model and GCN model. (b) Comparison of RMSE values between STAGCN model and GCN model.

In order to better illustrate that the STAGCN model proposed in this paper can capture the temporal and spatial dependence from SST data and obtain satisfactory prediction effect simultaneously, we visualized the error index RMSE of STAGCN model, GRU model, and GCN model in predicting SST for next 1 day, 7 days, 14 days and 30 days and analyzed their prediction performance. The visualization results are shown in Fig. 5. Fig. 5 (a) and Fig. 5 (b) are visual comparison effects of the STAGCN model compared with the GCN model and the GRU model on RMSE metric respectively. As can be seen from the bar chart, the RMSE of the models increases with the increase of the predicted length, but the error of the STAGCN model is smaller than that of the other two models demonstrating that the STAGCN model has the ability to obtain spatio-temporal correlation from SST series data. For example, the RMSE values of STAGCN model are about 0.16, 0.21, 0.23 and 0.35 lower than that of the GCN model for 1-day, 7-day, 14-day and 30-day SST forecasting, indicating that the STAGCN model is capable to obtain spatial features from sequence data. The RMSE values of the GRU model considered single temporal characteristics are raised by about 0.07, 0.14, 0.14 and 0.17 for future 1-day, 7-day, 14-day and 30-day SST prediction respectively compared with the STAGCN model, indicated that the STAGCN model can obtain the time correlation.

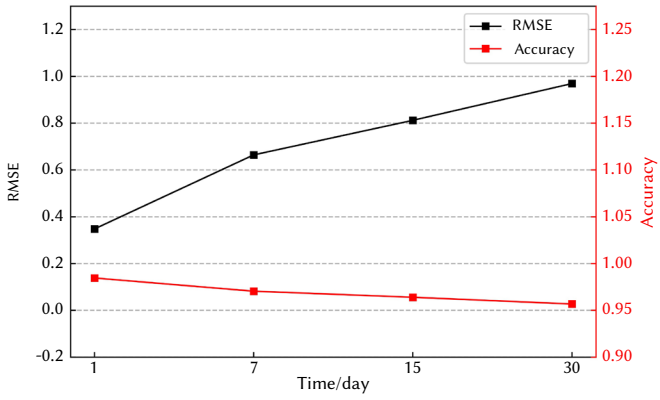


Fig. 6. The change of RMSE and Accuracy of STAGCN model for future SST prediction under different prediction interval conditions.

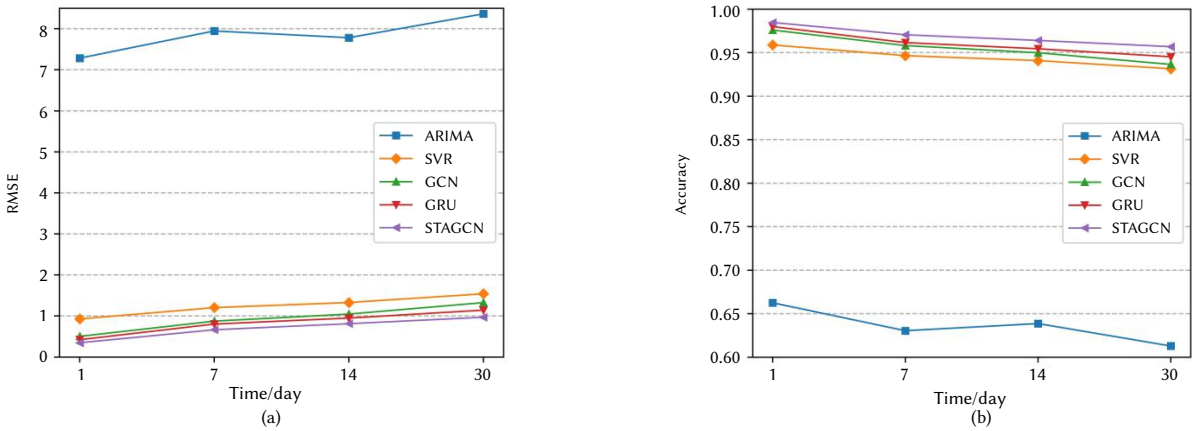


Fig. 7. The prediction performance of the STAGCN model and other methods under different prediction interval conditions.(a) Comparison of RMSE values between the STAGCN model and other methods. (b) Comparison of Accuracy between the STAGCN model and other methods.

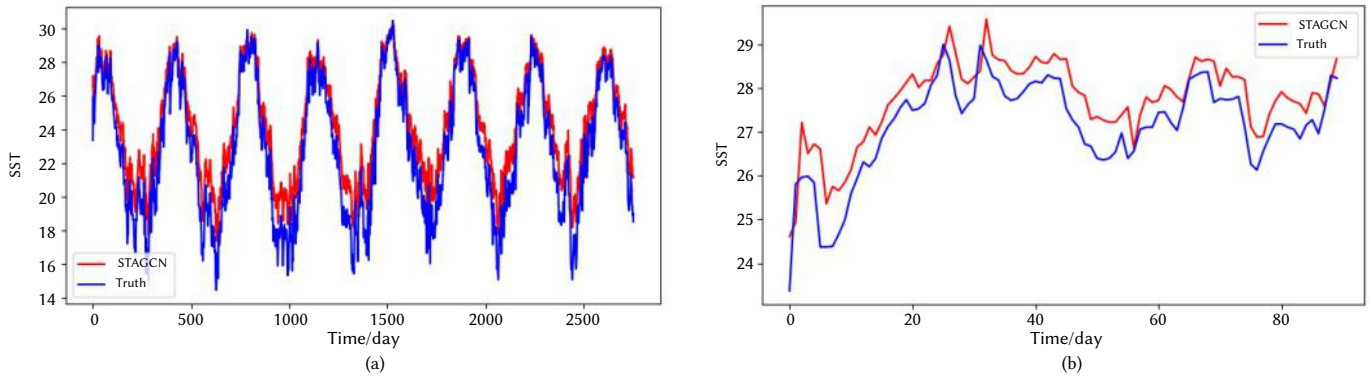


Fig. 8. The visualization comparison of the predicted and actual SST values for the next day.

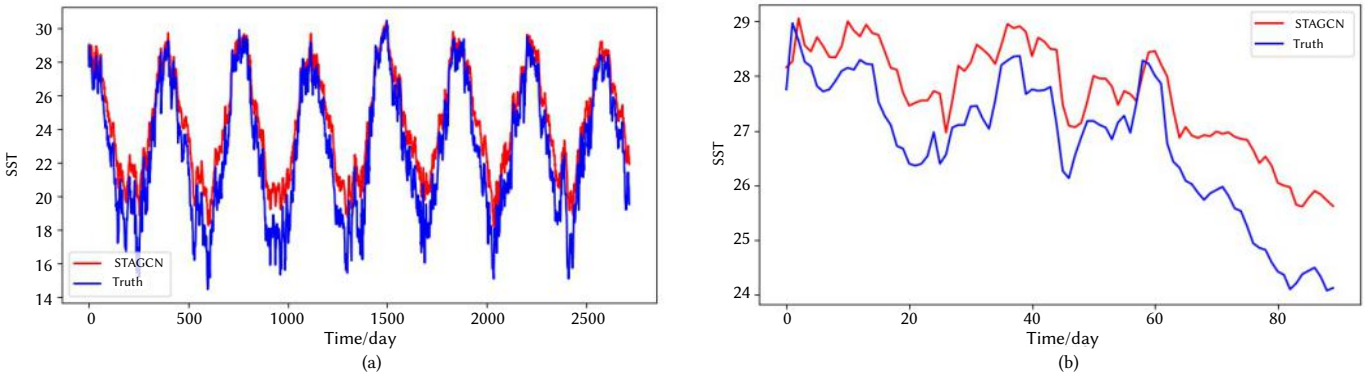


Fig. 9. The visualization comparison of the predicted and actual SST values for the 7 day.

To reflect the prediction effect of the STAGCN model, we visualized the variation trend of the RMSE and the accuracy in SST forecasting for the next 1 day, 7 days, 14 days and 30days. The results are shown in Fig.6. As can be seen from the figure, with the increase of prediction length, the error of the STAGCN model increases relatively large and the accuracy decreases slightly. Although the RMSE of the STAGCN model does not have a certain stability, its prediction accuracy is relatively stable, indicating that the STAGCN model can also achieve long-term SST prediction while it has better prediction ability for short-term SST prediction than long-term SST prediction.

Fig.7 (a) and Fig.7 (b) are RMSE and Accuracy results of SST prediction by different methods at different prediction horizons. It's observed that the STAGCN model achieved the lowest RMSE and the highest Accuracy for different predicted lengths.

To better explain the prediction performance of the STAGCN model, an ocean location point was randomly selected from the East China Sea dataset, in which we predicted the future SST for the next 1 day, 7 days, 14 days, and 30 days, and visualize the prediction effect of all the selected days and the next 90 days. Fig. 8 , Fig. 9 , Fig. 10 , and Fig. 11 show the visualization results of the SST for the 1-day, 7-days, 14-days and 30-days forecast intervals.

The prediction results of STAGCN model with the prediction length of 1 day, 7 days, 14 days and 30 days show that STAGCN model has poor prediction at peak and peak valley. The main reason may be that the GCN model in STAGCN model captures spatial features by constantly moving its defined smoothing filter, which will lead to excessive peak smoothing of the overall prediction results. At the same time, the model has a corresponding delay in the overall prediction.Although the prediction performance of STAGCN model decreases with the increase of prediction length, the fitting degree of predicted value and real value of STAGCN model is still high, and good prediction results can be obtained.

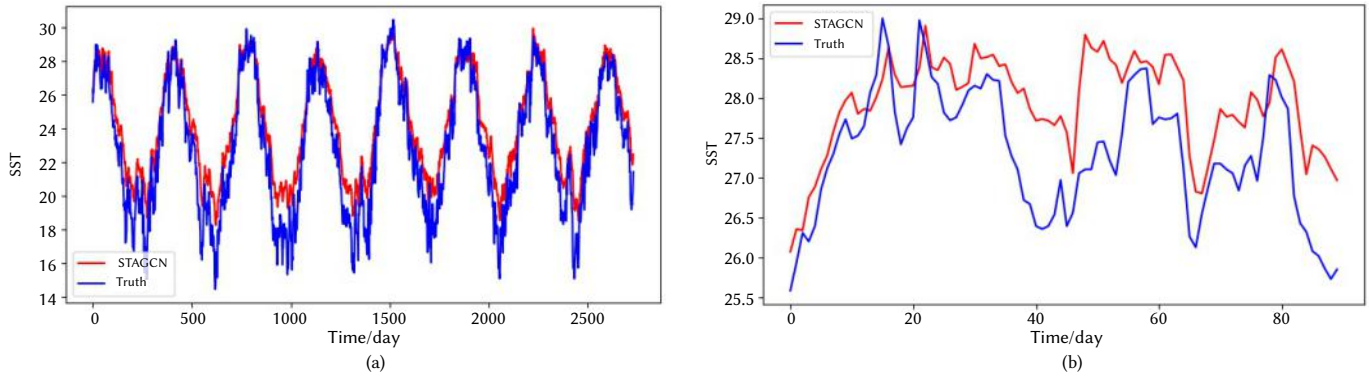


Fig. 10. The visualization comparison of the predicted and actual SST values for the 14 day.

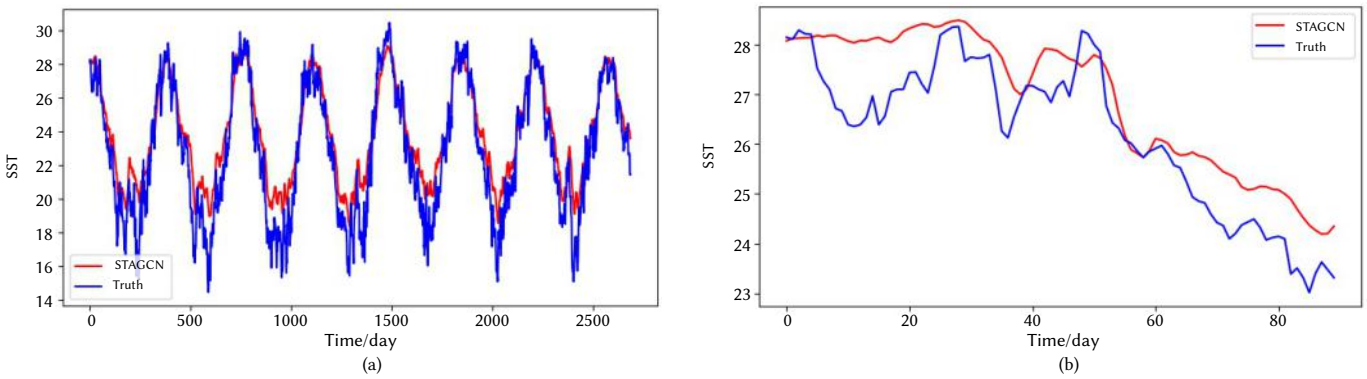


Fig. 11. The visualization comparison of the predicted and actual SST values for the 30 day.

STAGCN model can capture the temporal correlation and regional spatial characteristics from the TIME series of SST, and obtain the global spatiotemporal dynamic change trend by using the attention mechanism, which reduces the prediction error and improves the accuracy of prediction, and realizes the long-term and short-term prediction task of regional scale SST.

IV. Conclusion

The sea surface temperature is an important index to detect ocean changes, predict SST anomalies, and prevent natural disasters caused by abnormal changes, the dynamic variation of which have a profound impact on the whole marine ecosystem and the changes of climate. Therefore, it's essential to predict the future sea surface temperature. In order to achieve accurate SST prediction, a prediction model combining the GCN model with the GRU model and introduces the attention mechanism named the STAGCN model is proposed in this paper. We use the graph network to model the network of ocean location points. Nodes on the graph represent each ocean location point, edges on the graph represent that there have connections between location points. The GCN model is used to obtain the spatial correlation from the SST time series by constructing the spatial topology structure on the ocean points graph, which is obtained by the distance function between the nodes. The STAGCN model takes the GRU model to capture time dependence in the way of filtering and retaining historical and current SST information. Meanwhile, the attention model is applied to captures the importance of SST information from the output state and combines the global spatio-temporal characteristics from SST information. In this study, the experimental results of predicting the future short-term and long-term SST with STAGCN model on the data set indicating that the STAGCN model can achieve desirable prediction performance compared with the ARIMA model, SVR model, GCN model, and GRU model. In conclusion, the STAGCN model can acquire preferable forecasting results for future short-term and long-term SST prediction in the way of capturing global spatial characteristics and temporal dependence from SST series data.

References

[1] A. J. Constable, Melbourne-Thomas, "Climate change and southern ocean ecosystems i: how changes in physical habitats directly affect marine biota," *Global change biology*, vol. 20, pp. 3004–3025, 1 2014.

[2] M. Sumner, K. J. Michael, C. Bradshaw, M. A. Hindell, "Remote sensing of southern ocean sea surface temperature: implications for marine biophysical models," *Remote Sensing of Environment*, vol. 84, no. 2, pp. 161–173, 2003.

[3] J. Yang, J. Wen, Y. Wang, B. Jiang, H. Wang, H. Song, "Fog-based marine environmental information monitoring toward ocean of things," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4238–4247, 2020.

[4] M. Bouali, O. T. Sato, P. S. Polito, "Temporal trends in sea surface temperature gradients in the south atlantic ocean," *Remote Sensing of Environment*, vol. 194, pp. 100–114, 2017.

[5] M. A. Cane, A. Kaplan, D. Pozdnyakov, S. E. Zebiak, "Twentieth-century sea surface temperature trends," *Science*, vol. 275, no. 5302, pp. 957–960, 1997.

[6] G. A. Meehl, "Development of global coupled ocean- atmosphere general circulation models," *Climate Dynamics*, vol. 5, no. 1, pp. 19–33, 1990.

[7] S. L. Castro, G. A. Wick, M. Steele, "Validation of satellite sea surface temperature analyses in the beaufort sea using uptempo buoys," *Remote Sensing of Environment*, vol. 187, pp. 458–475, 2016.

[8] R. Salles, P. Mattos, A. Iorgulescu, E. Bezerra, L. Lima, E. Ogasawara, "Evaluating temporal aggregation for predicting the sea surface temperature of the atlantic ocean," *Ecological Informatics*, vol. 36, pp. 94–105, 2016.

[9] C. Xiao, N. Chen, C. Hu, K. Wang, Z. Chen, "Short and mid-term sea surface temperature prediction using time-series satellite data and lstm-adaboost combination approach," *Remote Sensing of Environment*, vol. 233, p. 111358, 2019.

[10] N. Chen, K. Wang, C. Xiao, J. Gong, "A heterogeneous sensor web node meta-model for the management of a flood monitoring system," *Environmental Modelling Software*, vol. 54, no. apr., pp. 222–237, 2014.

[11] K. Patil, M. C. Deo, M. Ravichandran, "Prediction of sea surface temperature by combining numerical and neural techniques," *Journal of Atmospheric and Oceanic Technology*, vol. 33, no. 8, pp. 1715–1726, 2016.

[12] Y. Xue, A. LeetmAa, "Forecasts of tropical pacific sst and sea level using a markov model," *Geophysical Research Letters*, vol. 27, no. 17, pp. 2701–2704, 2000.

[13] X. Yan, A. Leetmaa, J. Ming, "2000: Enso prediction with markov models: The impact of seal level," 2013, doi: https://doi.org/10.1175/1520-0442(2000)013%3C0849:EPWMMT%3E2.0.CO;2.

[14] D. C. Collins, C. J. C. Reason, F. Tangang, "Predictability of indian ocean sea surface temperature using canonical correlation analysis," *Climate Dynamics*, vol. 22, no. 5, pp. 481–497, 2004.

[15] T. Laepple, S. Jewson, "Five year ahead prediction of sea surface temperature in the tropical atlantic: a comparison between ipcc climate models and simple statistical methods," *Physics*, 2007, doi: https://doi.org/10.48550/arXiv.physics/0701165.

[16] K. Patil, M. C. Deo, S. Ghosh, M. Ravichandran, "Predicting sea surface temperatures in the north indian ocean with nonlinear autoregressive neural networks," *International Journal of Oceanography,2013,(2013-4-30)*, vol. 2013, pp. 1–11, 2013.

[17] I. D. Lins, M. Araujo, M. Moura, M. A. Silva, E. L. Droguett, "Prediction of sea surface temperature in the tropical atlantic by support vector machines," *Computational Statistics Data Analysis*, vol. 61, pp. 187– 198, 2013.

[18] A. Wu, W. W. Hsieh, B. Tang, "Neural network forecasts of the tropical pacific sea surface temperatures," *Neural Networks*, vol. 19, no. 2, pp. 145–154, 2006.

[19] S. G. Aparna, S. D'Souza, N. B. Arjun, "Prediction of daily sea surface temperature using artificial neural networks," *International Journal of Remote Sensing*, vol. 39, no. 11-12, pp. 4214–4231, 2018.

[20] Q. He, C. Zha, W. Song, Z. Hao, Y. Du, A. Liotta, C. Perra, "Improved particle swarm optimization for sea surface temperature prediction," *Energies*, vol. 13, 2020.

[21] J. Xie, J. Zhang, J. Yu, L. Xu, "An adaptive scale sea surface temperature predicting method based on deep learning with attention mechanism," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 5, pp. 740–744, 2020.

[22] X. Liu, T. Wilson, P. N. Tan, L. Luo, "Hierarchical lstm framework for long-term sea surface temperature forecasting," 2020.

[23] J. Liu, T. Zhang, G. Han, Y. Gou, "Td-lstm: Temporal dependence-based lstm networks for marine temperature prediction," *Sensors*, vol. 18, 2018.

[24] Park, Kim, Lee, Song, "Temperature prediction using the missing data refinement model based on a long short-term memory neural network," *Atmosphere*, vol. 10, no. 11, p. 718, 2019.

[25] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] R. Dey, F. M. Salemt, "Gate-variants of gated recurrent unit (gru) neural networks," pp. 1597–1600, 2017.

[27] Z. Qin, W. Hui, J. Dong, G. Zhong, S. Xin, "Prediction of sea surface temperature using long short-term memory," *IEEE Geoence and Remote Sensing Letters*, vol. 14, no. 10, pp. 1745–1749, 2017.

[28] Y. Feng, T. Sun, J. Dong, C. Li, "Study on long term sea surface temperature (sst) prediction based on temporal convolutional network (tcn) method," *ACM Turing Award Celebration Conference-China (ACM TURC 2021)*, pp. 28–32, 2021.

[29] M. Han, Y. Feng, X. Zhao, C. Sun, C. Liu, "A convolutional neural network using surface data to predict subsurface temperatures in the pacific ocean," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2019.

[30] M. Abadi, P. Barham, J. Chen, Z. Chen, X. Zhang, "Tensorflow: A system for large-scale machine learning," 2016.

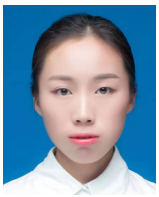
[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, "Scikit-learn: Machine learning in python," 2012.

Desheng Chen

He is currently pursuing the Ph.D. degree at the School of Electrical and Information Engineering, Tianjin University, China. His research interests include marine exploration, marine intelligent equipment control, artificial intelligence and Big data computing.

Jiabao Wen

He received the Ph.D. degree in information and communication engineering from Tianjin University, China, in 2021. He is currently a postdoctor at School of Electrical and Information Engineering, Tianjin University. His research interests include ocean information processing, pattern recognition,cloud computing.



Caiyun Lv

She received the Master degree in Electronics and Communication Engineering from Tianjin University, China, in 2022. Her research interests include ocean prediction and ocean information processing.

Using the Statistical Machine Learning Models ARIMA and SARIMA to Measure the Impact of Covid-19 on Official Provincial Sales of Cigarettes in Spain

Andoni Andueza[1], Miguel Ángel Del Arco-Osuna[1], Bernat Fornés[1], Rubén González-Crespo[2], Juan Manuel Martín-Álvarez[1]*

[1] Faculty of Economics and Business, Universidad Internacional de La Rioja, Logroño, La Rioja (Spain)
[2] School of Engineering and Technology, Universidad Internacional de La Rioja, Logroño, La Rioja (Spain)

Received 9 November 2022 | Accepted 16 February 2023 | Early Access 20 February 2023

Abstract

From a public health perspective, tobacco use is addictive by nature and triggers several cancers, cardiovascular and respiratory diseases, reproductive disorders, and many other adverse health effects leading to many deaths. In this context, the need to eradicate tobacco-related health problems and the increasingly complex environments of tobacco research require sophisticated analytical methods to handle large amounts of data and perform highly specialized tasks. In this study, time series models are used: autoregressive integrated moving average (ARIMA) and seasonal autoregressive integrated moving average (SARIMA) to forecast the impact of COVID-19 on sales of cigarette in Spanish provinces. To find the optimal solution, initial combinations of model parameters automatically selected the ARIMA model, followed by finding the optimized model parameters based on the best fit between the predictions and the test data. The analytical tools Autocorrelation Function (ACF), Partial Autocorrelation Function (PACF), Akaike Information Criterion (AIC), and Bayesian Information Criterion (BIC) were used to assess the reliability of the models. The evaluation metrics that are used as criteria to select the best model are: mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE), mean percentage error (MPE), mean error (ME) and mean absolute standardized error (MASE). The results show that the national average impact is slight. However, in border provinces with France or with a high influx of tourists, a strong impact of COVID-19 on tobacco sales has been observed. In addition, the least impact has been observed in border provinces with Gibraltar. Policymakers need to make the right decisions about the tobacco price differentials that are observed between neighboring European countries when there is constant and abundant cross-border human transit. To keep smoking under control, all countries must make harmonized decisions.

Keywords

ARIMA, Cigarette Sales, COVID-19, Machine Learning, SARIMA, Statistical Modeling, Time-series Forecast.

DOI: 10.9781/ijimai.2023.02.010

I. Introduction

FUNDAMENTALLY, there are two strategic reasons why the development of tobacco usage and behavior in any nation through time is a pertinent subject. First, smoking is addictive by nature, and it causes many different cancers, cardiovascular and respiratory conditions, reproductive problems, and a host of other harmful health impacts that result in thousands of deaths every year. As a result, the health system is burdened with significant costs related to the harm caused by tobacco use – on average, health spending accounts for 11.5% of the country's GDP [1]. Second, high-income nations' budgets are significantly impacted by the special taxes collected on tobacco; in Spain, tobacco is the product that provides the most to tax collection.

* Corresponding author.
E-mail addresses: juanmanuel.martin@unir.net

Additionally, a recent study that concentrated on the Spanish market demonstrates that some provinces do not have accurate official sales data that may be used to evaluate smoking control measures [2]. Furthermore, the empirical literature on regional heterogeneity in tobacco sales in Spain, concludes that areas of Spain bordering countries with high price differentials, such as Gibraltar and France, generate clusters of low and high per capita tobacco consumption, respectively [3]. In this regard, the border and tourist provinces in that study [2] are those in which sales are most impacted, supporting the prevalence of illegal commerce and substantial cross-border transactions. Thus, the findings of this study demonstrate the efficacy of shared policies adopted by the governments of neighboring nations that preserve a little price difference between them. In addition, the Spanish context is characterized by the strong impact that economic recessions have on cigarette sales [4]-[6]. Finally, a recent study suggests that in certain regions the demand for tobacco is not inelastic with respect to the price in the long term [7], which can generate large effects on provincial sales.

This scenario calls for advanced analytical tools to handle vast volumes of data and carry out highly specialized activities to eradicate tobacco-related health issues and the increasingly complicated environments of tobacco research. Due to this, some research has already used machine learning (hereafter ML) methods to analyze data pertaining to the tobacco market [8]. The definition of machine learning (ML) historically has been described as "a branch of research that offers computers the ability to learn without being explicitly programmed" to forecast future data or make decisions in uncertain situations [9]. The main goal of ML is to employ "brute force" instead of human supervision while analyzing data. Because ML requires far less human supervision than computer guidance, it can be considered as a natural extension of conventional statistical methodologies [10]. Unsupervised learning and supervised learning are categories found within machine learning. The two sets of ML approaches each have distinctive qualities that may be of interest to researchers studying tobacco. They are each geared toward resolving a certain difficulty. The focus of supervised learning is prediction. To predict the values of one or more output or response variables for a specific set of input or predictor variables, a model must be trained and validated [11]. In this sense, supervised learning techniques are used when the goal is to create a high-precision predictive model for future data. For example, supervised learning is useful for any tobacco market research that calls for extremely precise forecasts, such the creation of a public health surveillance program that predicts the likelihood of adolescent smoking beginning automatically [8]. Unsupervised learning, on the other hand, does not require an output variable because its goal is to ascertain the underlying probability distribution of the data (also known as density estimation) [8]. Examining tobacco-related social media discussions and identifying probable nicotine dependency subtypes by examining patient brain MRI data are two examples of unsupervised learning in tobacco research [8].

As stated, ML is a very powerful analytical tool for tobacco market researchers, the approaches can be broadly divided into supervised and unsupervised learning. However, in addition to this classification of techniques, studies that apply ML to tobacco market analysis can also be classified by the data (input) used. In this sense, we can find studies that analyze content on social networks, clinical report texts or administrative data [8]. In fact, several published papers that analyze the tobacco market focus on administrative data of the analysis [12]-[13]. Many of these studies apply supervised learning techniques to predict a binary phenomenon related to smoking cessation, including the intention to quit [14], adherence to smoking cessation therapies [15] and craving smoking highs or lows during a quit attempt [16]. However, few studies have applied supervised learning techniques with the aim of predicting continuous variables using, for example, regression or random forest [8], [17]-[19].

Although ML has been applied to the analysis of tobacco-related topics, to our knowledge, ML has never been applied to study the relationship between COVID-19 and tobacco. The COVID-19 pandemic has posed a unique opportunity to combat tobacco use [20]. Tobacco use and site bans, border closures, and lockdowns have had both positive and negative impacts on tobacco control. A recent study concludes that cigarette consumption decreased during the COVID-19 lockdown in 2020 [21]. However, other papers conclude the opposite. Specifically, one of the recent works concludes that the pandemic generated a 13% increase in tobacco sales [22]. Another paper indicates that this increase is because nicotine users use tobacco as their main mechanism to cope with stress and anxiety [23]. In addition, a paper indicates that the COVID-19 pandemic is related to higher tobacco sales and suggests research into whether smoking habits have changed since the pandemic lockdowns [24]. Regarding the use of time series analysis to analyze changes in cigarette sales, only one study has been found that addresses this problem and concludes that the sales observed during the pandemic are higher than expected [25]. Following on from this, in relation to the increase in tobacco sales, another study suggests that the intention to quit smoking has seen a post COVID-19 pandemic decrease [26]. Finally, other works that analyze smoking and COVID-19 suggest that tobacco sales should have been prohibited during the pandemic given the great opportunity that COVID-19 presented to eradicate smoking [27]-[29].

In Spain, although there are no works in which ML is applied to the tobacco market to explain the impact of COVID-19 on tobacco sales, there are papers that have analyzed the influence of COVID-19 on different aspects related to tobacco from another perspective. Some literature indicates that during the COVID-19 lockdown in Spain, tobacco consumption decreased [30]. In this same line of lower prevalence, another paper indicates that the success rate for quitting smoking went from 25% to 35% [31]. Another work, which focuses on analyzing smokers' perception of their exposure to the virus, suggests that many smokers may have changed their smoking patterns and it is possible that those who reduced their tobacco use outnumbered those who increased their consumption [32]. Another study that analyzes the impact of COVID-19 on tobacco consumption suggests that no significant effect of the pandemic on tobacco consumption is observed in Spain [33]. Finally, there is a group of works that indicate that the impact that COVID-19 has had on tobacco consumption depends on personal demographic issues and that not all people acted the same [34]. In addition, this block includes works that warn of the urgent need for tobacco consumers to give up smoking due to the damage to the health of consumers caused by this harmful product [35], [36].

To the best of our knowledge, no study has yet been done on the regional effects that COVID-19 has had on the Spanish tobacco market. In this study, we attempt to predict what the provincial tobacco market would have looked like in the absence of the COVID-19 pandemic. Then, we quantify the impact of the pandemic on cigarette sales as the difference between the forecast and the actual data. The data used in the current study comes from the Commission for the Trade of Tobacco and covers the period from January 2005 to December 2021 in terms of cigarette sales. The remainder of the document is structured as follows: Section II provides a description of the data and statistical models employed, together with information about the mathematics that underlies them, analytical tools, and evaluation measures. Section III discusses the computational architecture of the model parameter selection process. Section IV uses time series analysis to explore in depth the provincial impact of COVID-19 on the tobacco market. The conclusions reached from this investigation are provided in Section V.

II. Methods

To accomplish the goal outlined in this work, we generated an estimate of cigarette sales for the 48 Spanish provinces from January 2020 to December 2021 using the ML ARIMA and SARIMA statistical models. The ARIMA and SARIMA models as the best model over the uncorrelated ones and the models based on neural networks, because although these have a similar accuracy, the computational cost is much higher [37]. The suggested models have been optimized by choosing the most suitable parameters for each province. To ensure that the time series is the same length across all provinces, we used January 1, 2005, as the start date for each province. A minimum sample size of 30 observations is reportedly needed to provide a statistically significant forecast of time series data [38]. Given that each province's model was trained using data from January 2005 to December 2017 (168 observations), the sample size for estimating cigarette sales is significantly larger than the threshold set.

A. Data

A panel of monthly data from the Spanish provinces from January 2005 to December 2021 was used to build our empirical research. The Commission for the Trade of Tobacco's website's statistics section provided the cigarette sales data in euros and units. The National Institute of Statistics of Spain has been used to collect data on the population over the age of 18 to estimate provincial sales per capita.

The Islas Canarias, Ceuta and Melilla have been excluded from the analysis. As for the Islas Canarias, neither the tobacco market is regulated under a monopoly, nor is the price set by the Spanish government. That is, there is free trade, and the Spanish government does not intervene in the price. In addition, the restrictive regulations on consumption also have special features. In this sense, if that region is included in the study, the paper would present two important limitations. On the one hand, the behavior of Islas Canarias could be totally different as the population could more easily access tobacco consumption, given the free sale. On the other hand, the fact that the market is not regulated under a monopoly in these regions (singular), makes the data not homogeneous and reliable. As for Ceuta and Melilla, the data published by the Commission for the Trade of Tobacco is not homogeneous. Although sales of Ceuta and Melilla have been separated for a few years, until then the aggregate data was published, although they are two independent autonomous cities. Therefore, we do not have consistent data to analyze what happened in these autonomous cities.

B. Statistical Models and Description

Time series are collections of numerical values that each have a periodic component. Time series can be divided into two groups: stationary time series and non-stationary time series, depending on how the numerical values of the time series behave. Non-stationary time series have patterns that prevent the mean and/or variance from being constant, whereas stationary time series do not exhibit patterns in their mean and/or variance with respect to time. Seasonality or trend may be to blame for these trends. Calculating the difference between two succeeding observations can make non-stationary time series stationary. The trend and seasonality are eliminated from the time series using the differencing approach. First and second order differentiation are the two differentiation procedures that are most frequently employed; their calculation processes are described in equations (1) and (2):

$$y_t^{`} = y_t - y_{t-1} \tag{1}$$

$$y_t^{``} = y_t - 2y_{t-1} + y_{t-2} \tag{2}$$

where y_t are non-stationary time series data, $y_t^{`}$ is the time series after first order differentiation, $y_t^{``}$ is the time series after second order differentiation, y_{t-1} is the time series observation in period t-1, y_{t-2} is the time series observation in period t-2. Only when the time series is non-stationary after first-order differentiation is second-order differentiation required. There is also the option of seasonal distinction. In this instance, the distance between an observation and the identical observation from the prior year is used to calculate the difference (or period). Equation (3) provides a definition for the first degree of seasonal differentiation .

$$y_s^{`} = y_t - y_{t-m} \tag{3}$$

where $y_s^{`}$ is the time series after the first-order seasonal differentiation, y_{t-m} is the observation of the period t-m, m is the number of periods that exist between an observation and the same in the previous period. In this work, the time series were subjected to differentiation to eliminate seasonality and the resulting dataset is the one used to make the estimates. In addition, it must be taken into account that the estimation of the parameters of the ARIMA and SARIMA models is carried out assuming 4 basic assumptions: (i) the time series do not contain atypical points, (ii) the time series are composed of a single variable that is the one that, with its past values, helps to make the predictions; (iii) the time series are stationary, (iv) the model parameters and errors are constant throughout the time period.

Box and Jenkin created the ARIMA (p, d, q) model in 1976 [39], which can be used to predict stationary time series without seasonality. Three terms—p, d, and q—define this ARIMA model. The order of the moving average (MA) term is q, the order of the autoregression (AR) term is p, and the order of differentiation required to keep the time series stationary is d. The regression of the variable against itself to forecast its future behavior is known as autoregression. It involves comparing the value observed at a certain point to the values from earlier times. MA is a regression-like model that forecasts a variable in a later stage using the forecasting errors from an earlier time stage. The generalized equations for the p-th order AR model and the q-th order MA model are given below (Eqs. (4) and (5), respectively).

$$y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p + y_{t-p} + \varepsilon_t \tag{4}$$

$$y_t = C + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} \tag{5}$$

The AR model (Eq. (4)), the integration (I), and the MA model (Eq. (5)) are all combined to create ARIMA models in this study. To create the forecast, integration (I) uses differentiation in reverse. The mathematical formulation of the generalized ARIMA model is Eq (6).

$$y_t = C + \phi_1 y + \phi_p y_{t-p} + \cdots + \phi_n y_{t-n} + \theta_1 \varepsilon_{t-1} + \theta_q \varepsilon_{t-q} + \varepsilon_t \tag{6}$$

Where C is the independent term, ϕ_i ($i = 1, 2 \ldots p$) are the autoregressive model parameters, θ_i ($i = 1, 2 \ldots p$) are the moving average model parameters, y_t is the current time series, $y_{t-1}, y_{t-2}, \ldots y_{t-p}$ are past values and ε_t is random error of period t and is given by the following equation:

$$\varepsilon_t = y_t - y_{t-1} \tag{7}$$

To account for the seasonality of the time series, the seasonal ARIMA (SARIMA) model combines the non-seasonal ARIMA (p, d, and q) with additional seasonal terms (P, D, and Q). The seasonal AR term, seasonal moving average term, and seasonal differencing term are represented, respectively, by the P, Q, and D terms. The general SARIMA model is mathematically represented as follows:

$$\Phi_p(B^m)\phi_p(B)(1 - B^m)^D(1 - B)^d y_t = \Theta_Q(B^m)\theta_q(B)w_t \tag{8}$$

Where y_t is the non-stationary time series, w_t is the Gaussian white noise process, $\phi(B)$ is a non-seasonal autoregressive polynomial and $\theta(B)$ is a non-seasonal moving average polynomial, D is the seasonal differencing (the term is equal to 1 or 2, etc.). However, the value of D = 1 is sufficient to impose stationarity on the data, $\Phi(B^m)$ is a seasonal autoregressive polynomial, and $\Theta(B^m)$ is a seasonal moving average polynomial. Where B is defined as the backtracking operator which is expressed as follows:

$$B^k y_t = y_{t-k} \tag{9}$$

The expressions for the moving average model -Eq. (11)-, non-seasonal autoregressive model -Eq. 10-, seasonal AR model -Eq. 12-, and seasonal MA model -Eq. 13- are provided below.

$$\varphi(B) = 1 - \varphi_1 B - \varphi_2 B^2 - \cdots - \varphi_p B^{Pm} \tag{10}$$

$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q \tag{11}$$

$$\Phi_p(B^m) = 1 - \Phi_1 B^m - \Phi_2 B^{2m} + \cdots + \Phi_p B^{Pm} \tag{12}$$

$$\Theta_Q(B^m) = 1 + \Theta_1 B^m + \Theta_2 B^{2m} + \cdots + \Theta_Q B^{Qs} \tag{13}$$

Indicators are used to judge the accuracy of the time series analysis once the parameters of the ARIMA and SARIMA models have been estimated and the predictions have been produced. These indicators include the partial autocorrelation function (PACF), the Akaike information criterion (AIC), the autocorrelation function (ACF), and the Bayesian information criteria (BIC). These metrics show how the time series' observations relate to one another. While PACF correlates the time series with its own lagged values spaced by specific time units, ACF provides the correlation of the time series data with its prior time series data. The AIC and BIC penalized likelihood criterion's values are related; the lower they are, the more probable it is that the model will be accepted as a genuine model. Additionally, this study's evaluation criteria include mean error (ME), root mean square error (RMSE), mean absolute error (MAE), mean percentage error (MPE), mean absolute percentage error (MAPE), and scaled mean absolute error (MASE).

In a time series, autocorrelation is the relationship between the most recent observation and lagging observations. The ACF describes the linear relationship between the observation at time t and the observation at a previous time, and the autocorrelation plot is the time series' representation of autocorrelation vs delays (t-k). To illustrate, the ACF for the time series y_t is given by:

$$ACF(y_t, y_{t-k}) = \frac{Covariance\ (y_t, y_{t-k})}{variance\ (y_t)} \tag{14}$$

where k is the delay and is defined as the difference between y_t and y_{t-k}. On the other hand, in partial autocorrelation, the intermediate observations are considered when calculating the correlation between two observations at different times. For example, consider that a time series y_t, the PACF between two observations y_t and y_{t-2} (assuming k = 2) can be written as shown in the equation (15).

$$PACF(y_t, y_{t-2}) = \frac{Covariance\ (y_t, y_{t-2}|y_{t-1})}{\sqrt{variance\ (y_t|y_{t-1})}\sqrt{variance\ (y_{t-2}|y_{t-1})}} \tag{15}$$

Testing the created models is necessary to see how well they function in terms of elucidating the relationships between the variables. We have evaluated a model's ability to explain relationships using the information criteria. AIC and BIC are two widely used measures that assess the quality of models by rewarding those that have fewer mistakes and penalizing those that have too many parameters. The following is how AIC is mathematically represented:

$$AIC = -2logL(\hat{\theta}) + 2K \tag{16}$$

Where K is the total number of model parameters and $logL(\hat{\theta})$ is the likelihood function. BIC is a different model selection criterion in a similar vein. Compared to AIC, BIC imposes a lower penalty on the quantity of parameters. The model with the highest probability value is represented by the lower value in both the AIC and BIC settings. As a result, it aids time series analysts in selecting the optimal model from among the limited number of generated alternative models. The following is how BIC is mathematically represented:

$$BIC = -2logL(\hat{\theta}) + KlogN \tag{17}$$

Where N is the number of observations.

MAE, RMSE, MAPE, MPE, ME and MASE are often used to assess the accuracy of the ML models [40]-[41], which are given by the following equations:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{18}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{19}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{20}$$

$$MPE = \frac{1}{n}\sum_{i=1}^{n}\frac{y_i - \hat{y}_i}{y_i} \tag{21}$$

$$ME = \frac{1}{n}\sum_{i=1}^{n} y_i - \hat{y}_i \tag{22}$$

$$MASE = \frac{\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\frac{1}{T-1}\sum_{t=2}^{T}|y_t - y_{t-1}|} \tag{23}$$

Where $\hat{y}_i$ is the prediction made by the model and y_t is the actual value.

III. Computational Framework for Model Development

The scripts were created using the R programming language, which was set up in the RStudio environment, to accomplish the goal mentioned in this study [42]-[43]. The tidyverse and prediction libraries have also been used to clean the data, estimates, and graphic representations [44]–[46]. The appendix contains the R script that was utilized throughout key stages of the data analysis. Although there has already been a data cleansing phase, the actions used by the ML algorithm to accomplish the specified aim are detailed in this section. The algorithm initially determines whether each time series exhibits non-stationarity (if it had been done manually, this would have been checked using ACF and PACF plots). The time series is not stationary if the autocorrelation only slightly decreases as the number of delays increases. Next, the technique applies differences before executing ARIMA or SARIMA modeling if there is evidence that the time series is not stationary. Depending on which option best fits the time series, the algorithm selects either ARIMA or SARIMA. Given the substantial seasonal component present in the time series of tobacco sales, the method used SARIMA in the case study in this paper for all the series. The SARIMA models require an average processing time of 7 seconds to complete each simulation on the local computer.

The manual selection of the best parameter (p, d, q) $(P, D, Q)_m$ of the ARIMA and SARIMA models using ACF and PACF graphs can take a long time, since the models have been estimated for 48 provinces and 3 different variables (euros, packs and per capita packs). To select the appropriate combination of model parameter values, we perform a grid search using the forecast library, as indicated in the previous paragraph. This library uses AIC as an evaluation metric to choose the best model among several ARIMA and SARIMA models. Given that all the time series used begin in January 2005, end in December 2021 and tobacco sales show a strong seasonality, the parameter m took a value of 12 in all cases (Tables I, II and III).

The time series data of the 48 Spanish provinces was divided into two parts: the selected training dataset goes from January 2005 to December 2017 and the validation dataset goes from January 2018 to December 2019. Utilizing the training dataset, the model is constructed, and the validation dataset is used to estimate the model's performance. The following assessment metrics were used to assess the model: MAE, RMSE, MAPE, MPE, ME, and MASE. The model was used to forecast tobacco sales values from January 2020 to December 2021 (the period in which actual sales are altered due to lockdowns, restrictions in the hotel industry, and closure of borders for the 48 Spanish provinces), after the best model had been determined by training on the training dataset. Finally, to estimate the impact that COVID-19 has had on tobacco sales in Spain, the estimates made by the SARIMA models are compared with the actual sales observed from January 2020 to December 2021.

TABLE I. Selected SARIMA Models for Forecasting Euros

Province	SARIMA (p,d,q) (P,D,Q,m)	AIC	BIC	MAPE
Alava	(1,1,4)(2,0,0,12)	4,63E+03	4,66E+03	7,60E+00
Albacete	(2,1,2)(1,1,0,12)	4,25E+03	4,26E+03	7,63E+00
Alicante	(1,0,2)(2,1,1,12)	4,77E+03	4,79E+03	1,05E+01
Almería	(4,1,0)(2,1,2,12)	4,33E+03	4,35E+03	9,14E+00
Asturias	(2,1,1)(2,1,2,12)	4,48E+03	4,51E+03	6,97E+00
Ávila	(2,1,2)(1,1,0,12)	4,03E+03	4,05E+03	1,48E+01
Badajoz	(2,1,2)(2,1,0,12)	4,39E+03	4,41E+03	7,44E+00
Balears	(3,0,1)(2,1,0,12)	4,68E+03	4,70E+03	1,22E+01
Barcelona	(5,1,1)(2,0,0,12)	5,35E+03	5,38E+03	6,78E+00
Burgos	(0,0,0)(2,0,0,12)	4,78E+03	4,79E+03	9,73E+00
Cáceres	(2,1,2)(1,1,0,12)	4,28E+03	4,30E+03	9,32E+00
Cádiz	(2,1,2)(2,1,2,12)	4,44E+03	4,47E+03	1,19E+01
Cantabria	(2,1,2)(1,1,0,12)	4,36E+03	4,38E+03	8,78E+00
Castellón	(2,1,2)(2,1,0,12)	4,36E+03	4,39E+03	1,07E+01
Ciudad Real	(2,1,1)(2,1,0,12)	4,31E+03	4,33E+03	7,42E+00
Córdoba	(5,1,3)(2,0,0,12)	4,75E+03	4,79E+03	6,68E+00
Coruña (A)	(0,1,4)(2,0,0,12)	4,89E+03	4,91E+03	7,49E+00
Cuenca	(2,1,2)(2,1,0,12)	4,17E+03	4,19E+03	1,00E+01
Girona	(2,1,2)(2,1,0,12)	4,67E+03	4,69E+03	2,25E+01
Granada	(2,1,2)(2,1,0,12)	4,44E+03	4,46E+03	7,50E+00
Guadalajara	(2,1,2)(1,1,0,12)	4,09E+03	4,11E+03	7,84E+00
Guipúzcoa	(2,1,2)(1,1,0,12)	4,56E+03	4,58E+03	1,07E+01
Huelva	(2,1,2)(2,1,2,12)	4,30E+03	4,33E+03	1,04E+01
Huesca	(2,1,2)(2,1,0,12)	4,13E+03	4,15E+03	1,08E+01
Jaén	(2,1,2)(2,0,0,12)	4,78E+03	4,80E+03	5,86E+00
León	(2,1,2)(1,1,0,12)	4,33E+03	4,35E+03	9,12E+00
Lleida	(1,1,2)(1,0,0,12)	4,84E+03	4,86E+03	1,06E+01
Lugo	(2,1,2)(1,1,0,12)	4,23E+03	4,24E+03	8,42E+00
Madrid	(5,1,0)(2,0,0,12)	5,36E+03	5,38E+03	5,78E+00
Málaga	(2,1,1)(2,1,2,12)	4,62E+03	4,64E+03	1,20E+01
Murcia	(2,1,2)(2,1,2,12)	4,53E+03	4,56E+03	6,90E+00
Navarra	(2,1,2)(2,1,0,12)	4,54E+03	4,56E+03	1,08E+01
Ourense	(2,1,2)(1,1,0,12)	4,14E+03	4,16E+03	7,92E+00
Palencia	(2,1,2)(0,0,2,12)	4,51E+03	4,53E+03	9,06E+00
Pontevedra	(3,1,2)(2,1,2,12)	4,40E+03	4,43E+03	8,91E+00
Rioja (La)	(2,1,1)(2,0,0,12)	4,62E+03	4,64E+03	7,74E+00
Salamanca	(0,1,1)(0,0,2,12)	4,73E+03	4,74E+03	1,03E+01
Segovia	(2,1,2)(1,1,0,12)	4,03E+03	4,05E+03	1,02E+01
Sevilla	(2,1,2)(2,1,0,12)	4,60E+03	4,62E+03	6,71E+00
Soria	(2,1,2)(1,1,0,12)	3,94E+03	3,96E+03	9,94E+00
Tarragona	(1,0,0)(2,1,0,12)	4,57E+03	4,58E+03	1,37E+01
Teruel	(2,1,2)(1,1,0,12)	4,01E+03	4,03E+03	1,15E+01
Toledo	(2,1,2)(2,1,2,12)	4,37E+03	4,40E+03	7,16E+00
Valencia	(4,1,1)(2,0,0,12)	5,12E+03	5,15E+03	6,31E+00
Valladolid	(2,1,1)(2,0,0,12)	4,76E+03	4,78E+03	6,89E+00
Vizcaya	(4,1,3)(2,0,0,12)	4,88E+03	4,92E+03	5,48E+00
Zamora	(2,1,2)(2,1,0,12)	4,07E+03	4,09E+03	1,08E+01
Zaragoza	(2,1,2)(2,0,0,12)	4,89E+03	4,91E+03	6,29E+00

TABLE II. Selected SARIMA Models for Forecasting Packs

Province	SARIMA (p,d,q) (P,D,Q,m)	AIC	BIC	MAPE
Alava	(2,1,2)(2,0,1,12)	4,31E+03	4,33E+03	7,56E+00
Albacete	(2,1,2)(2,1,2,12)	3,92E+03	3,94E+03	7,62E+00
Alicante	(1,0,2)(2,1,1,12)	4,77E+03	4,79E+03	1,05E+01
Almería	(3,1,2)(2,1,1,12)	4,01E+03	4,04E+03	9,13E+00
Asturias	(2,1,2)(2,0,0,12)	4,54E+03	4,57E+03	6,94E+00
Ávila	(2,1,2)(2,1,0,12)	3,71E+03	3,73E+03	1,48E+01
Badajoz	(5,1,1)(2,1,1,12)	4,05E+03	4,08E+03	7,45E+00
Balears	(1,1,4)(2,1,0,12)	4,38E+03	4,40E+03	1,23E+01
Barcelona	(2,1,2)(2,0,0,12)	4,99E+03	5,02E+03	6,75E+00
Burgos	(4,1,1)(2,0,0,12)	4,39E+03	4,41E+03	9,71E+00
Cáceres	(3,1,1)(1,1,2,12)	3,94E+03	3,97E+03	9,31E+00
Cádiz	(2,1,3)(2,1,0,12)	4,15E+03	4,18E+03	1,19E+01
Cantabria	(4,1,1)(2,1,2,12)	4,02E+03	4,05E+03	8,74E+00
Castellón	(2,1,0)(2,1,1,12)	4,05E+03	4,07E+03	1,06E+01
Ciudad Real	(2,1,2)(2,1,1,12)	3,98E+03	4,00E+03	7,45E+00
Córdoba	(2,1,2)(2,0,0,12)	4,43E+03	4,45E+03	6,71E+00
Coruña (A)	(2,1,2)(2,0,0,12)	4,53E+03	4,55E+03	7,47E+00
Cuenca	(1,1,4)(2,1,2,12)	3,86E+03	3,89E+03	1,00E+01
Girona	(2,1,2)(1,1,0,12)	4,38E+03	4,40E+03	2,22E+01
Granada	(3,1,3)(2,1,2,12)	4,12E+03	4,15E+03	7,49E+00
Guadalajara	(3,1,2)(2,1,2,12)	3,75E+03	3,78E+03	7,84E+00
Guipúzcoa	(2,1,1)(1,1,2,12)	4,23E+03	4,25E+03	1,06E+01
Huelva	(2,1,1)(2,1,2,12)	4,01E+03	4,03E+03	1,03E+01
Huesca	(4,1,1)(2,1,1,12)	3,81E+03	3,83E+03	1,07E+01
Jaén	(2,1,2)(2,0,0,12)	4,43E+03	4,46E+03	5,82E+00
León	(4,1,3)(1,1,2,12)	3,96E+03	3,99E+03	9,10E+00
Lleida	(2,1,1)(2,0,0,12)	4,49E+03	4,51E+03	9,46E+00
Lugo	(4,1,1)(2,0,0,12)	4,22E+03	4,25E+03	8,41E+00
Madrid	(3,1,3)(2,0,0,12)	5,03E+03	5,06E+03	5,73E+00
Málaga	(2,1,2)(2,1,0,12)	4,32E+03	4,34E+03	1,19E+01
Murcia	(3,1,1)(2,1,1,12)	4,22E+03	4,25E+03	6,87E+00
Navarra	(2,1,2)(2,1,0,12)	4,23E+03	4,25E+03	1,07E+01
Ourense	(2,1,2)(1,1,1,12)	3,81E+03	3,83E+03	7,91E+00
Palencia	(2,1,2)(2,0,0,12)	4,13E+03	4,16E+03	9,01E+00
Pontevedra	(4,1,0)(2,1,2,12)	4,09E+03	4,12E+03	8,84E+00
Rioja (La)	(3,1,4)(2,0,0,12)	4,26E+03	4,29E+03	7,71E+00
Salamanca	(0,1,2)(2,0,0,12)	4,36E+03	4,37E+03	1,03E+01
Segovia	(2,1,2)(2,1,0,12)	3,71E+03	3,73E+03	1,01E+01
Sevilla	(2,1,0)(2,1,2,12)	4,31E+03	4,33E+03	6,80E+00
Soria	(3,0,1)(2,1,0,12)	3,63E+03	3,65E+03	9,90E+00
Tarragona	(3,1,1)(2,1,0,12)	4,22E+03	4,24E+03	1,36E+01
Teruel	(3,1,3)(2,1,0,12)	3,71E+03	3,73E+03	1,14E+01
Toledo	(2,1,2)(2,1,2,12)	4,06E+03	4,09E+03	7,17E+00
Valencia	(2,1,0)(2,0,0,12)	4,78E+03	4,79E+03	6,24E+00
Valladolid	(1,1,2)(2,0,0,12)	4,42E+03	4,44E+03	6,86E+00
Vizcaya	(2,1,2)(2,0,0,12)	4,54E+03	4,56E+03	5,45E+00
Zamora	(1,1,3)(1,1,2,12)	3,74E+03	3,77E+03	1,08E+01
Zaragoza	(2,1,2)(2,0,0,12)	4,53E+03	4,55E+03	6,26E+00

TABLE III. Selected SARIMA Models for Forecasting Per Capita Packs

Province	SARIMA (p,d,q) (P,D,Q,m)	AIC	BIC	MAPE
Alava	(2,1,2)(2,0,1,12)	4,30E+02	4,57E+02	7,67E+00
Albacete	(2,1,2)(2,1,2,12)	2,85E+02	3,11E+02	7,61E+00
Alicante	(2,1,0)(2,1,0,12)	3,72E+02	3,87E+02	1,03E+01
Almería	(3,1,2)(2,1,1,12)	2,47E+02	2,73E+02	9,19E+00
Asturias	(2,1,2)(2,0,0,12)	2,71E+02	2,95E+02	6,94E+00
Ávila	(5,1,1)(2,1,0,12)	2,94E+02	3,20E+02	1,48E+01
Badajoz	(5,1,1)(2,1,1,12)	2,61E+02	2,91E+02	7,44E+00
Balears	(2,1,0)(2,1,0,12)	4,96E+02	5,11E+02	1,36E+01
Barcelona	(2,1,2)(2,0,0,12)	2,42E+02	2,66E+02	6,82E+00
Burgos	(2,1,1)(2,0,0,12)	4,60E+02	4,78E+02	9,71E+00
Cáceres	(2,1,2)(2,1,1,12)	2,90E+02	3,13E+02	9,31E+00
Cádiz	(2,1,2)(2,1,0,12)	2,08E+02	2,29E+02	1,18E+01
Cantabria	(4,1,1)(2,1,2,12)	2,68E+02	2,98E+02	8,74E+00
Castellón	(3,1,2)(2,1,1,12)	3,03E+02	3,30E+02	1,06E+01
Ciudad Real	(2,1,2)(2,1,1,12)	2,69E+02	2,92E+02	7,45E+00
Córdoba	(2,1,2)(2,0,0,12)	2,77E+02	3,02E+02	6,67E+00
Coruña (A)	(2,1,2)(2,0,0,12)	2,45E+02	2,67E+02	7,46E+00
Cuenca	(1,1,2)(2,1,1,12)	3,91E+02	4,12E+02	1,00E+01
Girona	(2,1,2)(1,1,1,12)	5,82E+02	6,02E+02	2,21E+01
Granada	(4,1,0)(2,1,2,12)	2,57E+02	2,84E+02	7,43E+00
Guadalajara	(5,1,0)(2,1,1,12)	2,69E+02	2,96E+02	7,72E+00
Guipúzcoa	(2,1,1)(1,1,1,12)	4,26E+02	4,44E+02	1,06E+01
Huelva	(2,1,1)(2,1,0,12)	3,15E+02	3,33E+02	1,03E+01
Huesca	(4,1,1)(2,1,1,12)	3,23E+02	3,50E+02	1,07E+01
Jaén	(2,1,2)(2,0,0,12)	3,39E+02	3,63E+02	5,82E+00
León	(4,1,3)(1,1,2,12)	2,39E+02	2,72E+02	9,10E+00
Lleida	(2,1,1)(2,0,0,12)	5,22E+02	5,40E+02	1,07E+01
Lugo	(4,1,1)(2,0,0,12)	2,88E+02	3,12E+02	8,41E+00
Madrid	(2,1,4)(2,0,0,12)	2,31E+02	2,61E+02	5,88E+00
Málaga	(2,1,2)(2,1,0,12)	3,06E+02	3,27E+02	1,20E+01
Murcia	(5,1,0)(2,1,1,12)	2,28E+02	2,55E+02	6,99E+00
Navarra	(2,1,2)(2,1,0,12)	4,66E+02	4,87E+02	1,07E+01
Ourense	(2,1,2)(1,1,1,12)	1,95E+02	2,16E+02	7,91E+00
Palencia	(2,1,1)(2,0,0,12)	4,29E+02	4,47E+02	9,00E+00
Pontevedra	(4,1,0)(2,1,2,12)	2,03E+02	2,29E+02	8,84E+00
Rioja (La)	(2,1,2)(2,0,0,12)	3,88E+02	4,12E+02	7,77E+00
Salamanca	(0,1,2)(2,0,0,12)	4,37E+02	4,52E+02	1,03E+01
Segovia	(5,1,0)(2,1,2,12)	3,16E+02	3,46E+02	1,01E+01
Sevilla	(2,1,0)(2,1,0,12)	2,41E+02	2,55E+02	6,70E+00
Soria	(1,0,0)(1,1,0,12)	3,83E+02	3,95E+02	9,90E+00
Tarragona	(3,1,1)(2,1,0,12)	4,00E+02	4,21E+02	1,36E+01
Teruel	(5,0,2)(2,1,0,12)	3,40E+02	3,73E+02	1,14E+01
Toledo	(4,1,0)(2,1,2,12)	2,80E+02	3,07E+02	7,09E+00
Valencia	(2,1,2)(2,0,0,12)	2,68E+02	2,92E+02	6,33E+00
Valladolid	(2,1,2)(2,0,0,12)	3,76E+02	3,98E+02	6,88E+00
Vizcaya	(5,1,4)(2,0,0,12)	2,36E+02	2,73E+02	5,53E+00
Zamora	(1,1,3)(1,1,1,12)	2,84E+02	3,05E+02	1,08E+01
Zaragoza	(2,1,2)(2,0,0,12)	3,05E+02	3,26E+02	6,35E+00

IV. Results and Discussions

Table IV shows the results of the comparison between the actual sales observed after COVID-19 and the estimates made by the model (from January 2020 to December 2021). In this sense, the results of the gaps detected in terms of sales in euros, in packs and in per capita packs are shown. Positive gaps indicate that observed sales exceed the estimates made by the model, while negative gaps indicate that actual sales after COVID-19 are lower than the estimates made by the estimated SARIMA models. In the table, the minimum, maximum and average of the calculated provincial gaps can be observed. In addition, Fig. 1 graphically shows the dynamics of the time series together with the forecast made using the variable per capita packs.

If we focus on the calculated average gap, in some provinces the impact of COVID-19 on tobacco sales has been almost nil. Specifically, in Almería, Ávila, Cantabria, Coruña (A), Valladolid and Zaragoza, the impact of COVID-19 on per capita packs is less than 1% in absolute value. Given this situation, in Fig. 1 it can be seen how in these provinces, in which tobacco sales were not affected by COVID-19, the forecast lines and actual post-COVID-19 sales overlap. However, in other provinces the impact of COVID-19 has caused a significant negative effect on sales per capita packs that reaches, on average, up to -25.72%. The provinces in which this situation is observed are Alicante/Alacant, Baleares (Illes), Girona, Guipúzcoa, Lleida and Málaga, in which the average impact of COVID-19 on monthly tobacco sales has been -18, 95%, -25.72%, -22.71%, -14.98%, -16.66% and -11.41%, respectively. In the case of these provinces, Fig. 1 shows that the forecast line exceeds the Post COVID-19 sales line from January 2020 to December 2021. In all cases, the provinces in which these effects are observed are from areas with a high influx of tourists and border areas with France. These results are in line with previous literature indicating that tobacco sales in Spain are highly conditioned by sales to tourists and residents of France [2],[47].

Regarding the minimum value of the provincial gaps calculated in sales per capita packs, Table IV shows that the greatest impact, in absolute value, of COVID-19 on tobacco sales was observed in Alicante/Alacant, Balears (Illes), Girona, Guipúzcoa, Lleida and Navarra, in which the minimum value of the impact of COVID-19 on monthly tobacco sales was -39.11%, -58.20%, -66.74%, -54, 48%, -46.69% and -51.38%, respectively. In all cases, this minimum value was detected in the months of February and/or March 2020, months in which the borders of Spain were closed due to the COVID-19 pandemic. In other words, the greatest impact in absolute value of COVID-19 on tobacco sales is also observed in provinces bordering France and provinces with a high influx of tourists. On the other hand, the provinces in which the minimum impact has been smaller in absolute value are Cádiz and Sevilla, where said impact has been -11.37% and -5.41%, respectively. These results are also in line with previous literature that indicates that sales in Cádiz and Sevilla are affected by the proximity of these provinces to Gibraltar, an area with which there is a significant price differential [2].

Our results indicate that the restrictions implemented by governments due to COVID-19 have had a significant effect on provincial tobacco sales in Spain. In this sense, we find that the provinces in which sales are most affected are the border and tourist provinces, which seems to indicate that, regardless of the limitation of leisure, the restriction that has most affected sales is the closure of borders. The results suggest that in tourist and border areas with France, COVID-19 has caused a negative effect on tobacco sales that in most cases had not yet been reversed by December 2021.

Fig. 1 (A). 2020 and 2021 forecast of the per capita packs based on the best SARIMA models selected.

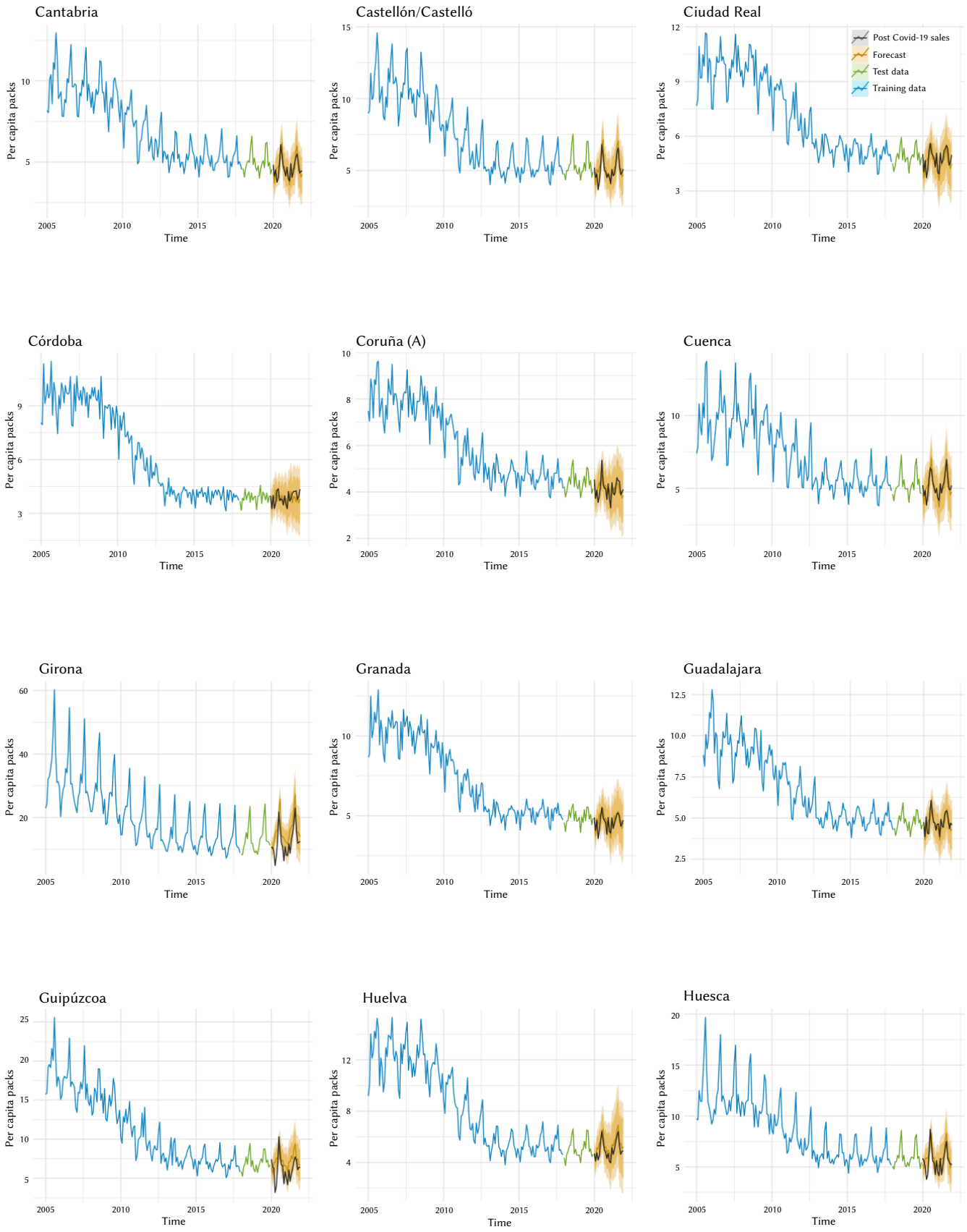


Fig. 1 (B). 2020 and 2021 forecast of the per capita packs based on the best SARIMA models selected.

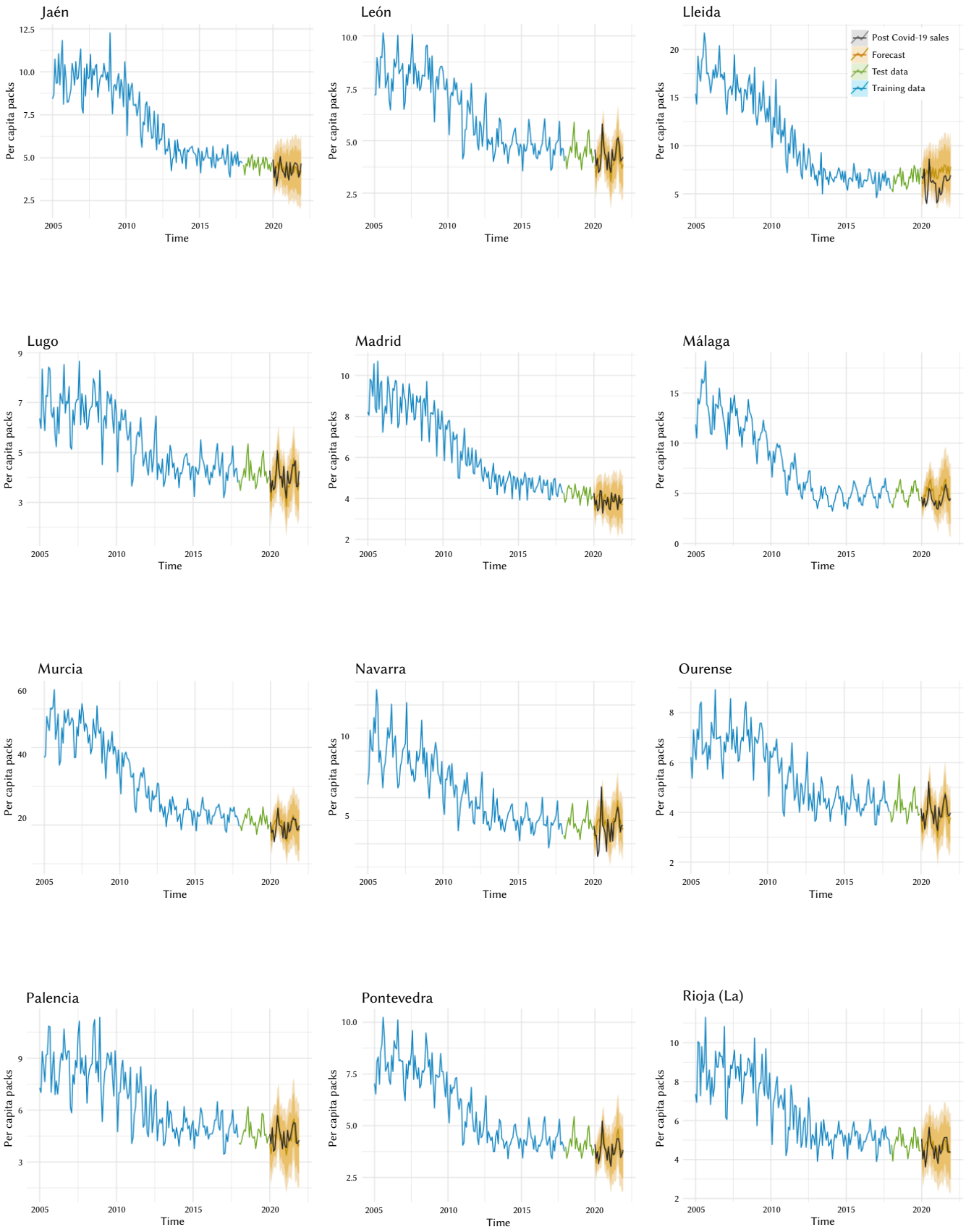


Fig. 1 (C). 2020 and 2021 forecast of the per capita packs based on the best SARIMA models selected.

Fig. 1 (D). 2020 and 2021 forecast of the per capita packs based on the best SARIMA models selected.

TABLE IV. Provincial Impact of COVID-19 on the Spanish Tobacco Market

Province	Gap in Euros (%)			Gap in Per capita packs (%)		
	Min	Max	Mean	Min	Max	Mean
Alava	-18,01	22,02	0,26	-19,00	27,09	2,87
Albacete	-21,68	12,96	-3,04	-5,83	59,23	26,89
Alicante	-38,13	11,16	-14,88	-39,11	8,52	-18,95
Almería	-23,63	10,38	-1,47	-21,65	14,01	0,22
Asturias	-18,21	10,08	-1,15	-15,70	19,49	5,17
Ávila	-26,73	18,05	-0,55	-25,08	15,36	0,29
Badajoz	-21,31	11,16	-0,03	-21,42	13,57	1,41
Balears	-58,08	8,50	-26,50	-58,20	5,41	-25,72
Barcelona	-20,96	5,34	-5,50	-18,91	9,57	-2,35
Burgos	-35,41	16,68	-5,01	-33,38	21,68	-2,26
Cáceres	-28,66	14,05	-2,10	-26,76	16,16	2,08
Cádiz	-8,68	22,96	5,74	-11,37	17,13	2,15
Cantabria	-20,81	12,65	-1,76	-19,17	13,81	-0,09
Castellón	-26,30	11,48	-3,67	-24,73	12,79	-3,13
Ciudad Real	-20,81	13,59	0,86	-19,36	16,78	3,36
Córdoba	-15,72	11,83	0,89	-16,35	17,01	1,72
Coruña (A)	-14,06	10,25	-1,17	-15,98	13,76	0,40
Cuenca	-22,24	15,22	2,42	-23,45	14,14	2,13
Girona	-65,26	9,61	-17,54	-66,74	4,76	-22,71
Granada	-21,43	9,42	-3,10	-20,43	8,63	-2,02
Guadalajara	-20,37	14,15	1,05	-17,27	14,58	3,00
Guipúzcoa	-53,43	22,25	-11,65	-54,48	20,10	-14,98
Huelva	-18,14	16,42	0,37	-18,38	2,95	-7,14
Huesca	-33,40	15,58	-5,25	-35,35	15,17	-6,60
Jaén	-25,28	6,99	-4,10	-23,43	12,57	-0,99
León	-22,22	15,01	-1,02	-16,86	22,07	4,79
Lleida	-47,06	10,75	-16,42	-46,69	11,46	-16,66
Lugo	-16,04	20,94	-0,76	-11,83	23,14	3,83
Madrid	-20,06	5,36	-4,66	-17,69	11,76	-1,33
Málaga	-29,07	8,13	-12,07	-29,53	9,74	-11,41
Murcia	-24,57	10,08	-4,24	-23,27	12,56	-2,45
Navarra	-50,82	32,66	-5,84	-51,38	30,92	-6,74
Ourense	-19,94	18,77	0,57	-14,68	18,05	3,62
Palencia	-22,39	23,28	0,82	-18,49	27,07	4,03
Pontevedra	-16,82	11,58	-0,36	-15,88	13,09	-1,25
Rioja (La)	-24,21	12,10	-3,58	-23,41	15,79	-1,49
Salamanca	-34,98	18,00	-7,76	-30,80	25,68	-3,07
Segovia	-23,38	14,12	0,25	-20,82	17,15	2,15
Sevilla	-3,79	15,71	4,32	-5,41	15,82	3,79
Soria	-24,35	27,71	1,85	-24,04	29,18	3,49
Tarragona	-33,09	8,87	-8,12	-30,66	9,99	-7,53
Teruel	-19,93	16,38	-2,85	-18,87	15,81	-2,27
Toledo	-16,61	12,20	2,59	-16,86	8,75	-3,70
Valencia	-20,97	9,35	-4,94	-22,27	4,08	-5,73
Valladolid	-26,57	6,10	-6,14	-22,44	16,80	0,61
Vizcaya	-17,89	7,13	-3,47	-18,76	6,88	-3,43
Zamora	-25,08	24,26	1,72	-22,25	23,76	4,65
Zaragoza	-23,41	5,65	-5,43	-20,92	11,84	-0,62

V. Conclusions

In this study we have predicted the impact that COVID-19 has had on tobacco sales in Spain (in euros, in packs and in per capita packs) from January 2020 to December 2021, using ARIMA and SARIMA Machine Learning statistical models. Our estimates indicate that the greatest impact of COVID-19 on cigarette sales is observed in tourist provinces and those bordering France, where, in the months of border closures, sales were up to 66.74% lower than the forecast made. On the other hand, in the provinces bordering Gibraltar, the impact of COVID-19 was very slight (5.41%). The reasons why COVID-19 may impact tobacco sales may be public awareness, leisure restrictions, border closures, etc. However, it seems that the greatest impact of COVID-19 has been caused by the closure of borders.

Along these lines, in provinces such as Alicante/Alacant, Baleares (Illes), Girona, Guipúzcoa, Lleida, Málaga and Navarra, a strong impact of COVID-19 on tobacco sales has been observed. In addition, the least impact has been observed in Cádiz and Sevilla. If the national average impact is observed, in Spain COVID-19 has had almost no effect. Specifically, the average provincial impact in Spain is close to -2%. This is because the forecast made with the SARIMA models and Post COVID-19 sales are almost the same in most Spanish provinces.

The results seem to show that the closure of borders has had a marked impact on provincial tobacco sales in Spain. Therefore, it seems that the effect of tourism and cross-border purchases between Spain and France and Spain and Gibraltar have been altered by the border restrictions caused by COVID-19. Based on our predictions and forecasts, policymakers must make the right decisions about the tobacco price differentials observed between European countries where there is constant and abundant cross-border movement. To keep smoking under control, harmonized decisions by all countries must be made.

This work is not without limitations. A recent work reveals that Philip Morris International, the world's leading tobacco manufacturer, is using heated tobacco products (HTPs) to replace the traditional cigarette. The results achieved may be influenced by this phenomenon [48]. In addition, a recent study also indicates that the affordability of cigarettes is a key factor for their demand in Spain. For this reason, part of the "no loss" in Seville and Cádiz may be motivated by the affordability effect [49].

Given the limitations indicated, the lines of future research can be summarized in three. First, it is interesting to analyze whether HTPs are causing part of the gaps detected in this paper. Secondly, it would be important to analyze the role that affordability plays in the gaps detected. Finally, the behavior of substitute products must be analyzed to find out if part of the effects detected in this paper may be due to the consumption of other alternative products.

Appendix

A. Snapshots of the R Script For the Forecasting of the Time-Series Data

```
### STEP 1: Data collection,Import required, Read data
into dataframe, define the variable Per Capita Packs.
library(tidyverse)
library(forecast)
df = read.xlsx("../TFM/tobaccosales.xlsx")

colnames(df) = c("Province","Month","Euros","Packs",
"Year","Population")

df <- data.frame(df)

df$PercapitaPacks = df$Packs/df$Population
```

```
### STEP 2: Create the descriptive statistics table.

estdescriptiv1 = df %>%
  group_by(Province) %>%
  summarise(meanPacks = mean(Packs),
            sdPacks = sd(Packs),
            q1Packs = quantile(Packs, c(0.25)),
            q2Packs = quantile(Packs, c(0.5)),
            q3Packs = quantile(Packs, c(0.75)))
estdescriptiv2 = total %>%
  group_by(Province) %>%
  summarise(meanEuros = mean(Euros),
            sdEuros = sd(Euros),
            q1Euros = quantile(Euros, c(0.25)),
            q2Euros = quantile(Euros, c(0.5)),
            q3Euros = quantile(Euros, c(0.75)))
estdescriptiv3 = total %>%
  group_by(Province) %>%
  summarise(meanPacksperCapita = mean(PacksperCapita),
     sdPacksperCapita = sd(PacksperCapita),
     q1PacksperCapita = quantile(PacksperCapita, c(0.25)),
     q2PacksperCapita = quantile(PacksperCapita, c(0.5)),
     q3PacksperCapita = quantile(PacksperCapita, c(0.75)))

estdescriptiv <- cbind(estdescriptiv 1, estdescriptiv2,
estdescriptiv3)


### STEP 3: Convert data into date-time format and
create the dataset of train, test and post COVID-19
sales and build the model using auto.arima.

timeserielAlava = df %>%
  filter(Province == "Álava")

timeserieAlavaEurosTrain = ts(timeserieAlava[c(1:156),]
$Euros, start = c(2005,01), frequency = 12)

timeserieAlavaEurosTest = ts(serietemporalAlava[c(157:180),
]$Euros, start = c(2018,01) , frequency = 12)

totalAlavaEuros = ts(timeserieAlava[c(1:180),]$Euros,
start = c(2005,01), frequency = 12)

postcovid19AlavaEuros = ts(timeserieAlava[c(181:204),]
$Euros, start = c(2020,01), frequency = 12)

STAlavaEurosTrain = auto.arima(serietemporalAlavaEurosTrain)
STAlavaEurosTest = auto.arima(serietemporalAlavaEurosTest)
predAlavaEuros = forecast(auto.arima(totalAlavaEuros), 24)

### STEP 4. Graphical representation of the time series
(train, test, forecast and post COVID-19 sales).

plotAlavaEuros <-autoplot(timeserieAlavaEurosTrain,
series = "train") +
  autolayer(timeserieAlavaEurosTest, series = "test")+
  autolayer(predAlavaEuros, series = "prediction") +
  autolayer(postcovid19AlavaEuros, series =
"observed")+
  guides(colour = guide_legend(""))+
  labs(x = "Time",
       y = "Euros",
       title = "Alava")+
  scale_color_manual(labels = c("Post Covid-19 sales",
"Forecast", "Test data", "Training data"),
                     values = c("#333333", "#db8100",
"#7fb433", "#0098cd"))+
  theme_minimal()

### STEP 5. Calculate the provincial impact of COVID-19
on the Spanish tobacco market.

impactAlavaEuros <- ((postcovid19AlavaEuros
-predAlavaEuros$mean)/predAlavaEuros$mean)*100
```

B. Others Accuracy Metrics of the ML Models

APPENDIX TABLE I. Selected SARIMA Models for Frecasting Euros

Province	ME	RMSE	MAE	MPE	MASE
Alava	-1,24E-09	4,84E+05	4,06E+05	-8,16E-01	1,39E+00
Albacete	1,48E+04	7,41E+05	5,86E+05	-7,22E-01	1,67E+00
Alicante	1,92E+05	5,40E+06	4,63E+06	-1,03E+00	2,77E+00
Almería	2,52E+04	1,70E+06	1,37E+06	-1,11E+00	3,16E+00
Asturias	-6,21E-09	1,77E+06	1,39E+06	-7,72E-01	2,14E+00
Ávila	5,35E+03	6,19E+05	4,91E+05	-3,01E+00	2,81E+00
Badajoz	-2,48E-09	1,12E+06	9,02E+05	-8,55E-01	2,01E+00
Balears	-1,31E+05	4,06E+06	3,42E+06	-2,50E+00	2,79E+00
Barcelona	-3,48E-08	7,78E+06	6,42E+06	-6,72E-01	1,81E+00
Burgos	-1,86E-09	8,14E+05	6,53E+05	-1,44E+00	2,94E+00
Cáceres	0,00E+00	9,77E+05	7,55E+05	-1,40E+00	2,22E+00
Cádiz	5,29E+04	2,27E+06	1,76E+06	-1,88E+00	1,83E+00
Cantabria	2,38E+03	1,40E+06	1,04E+06	-1,32E+00	2,72E+00
Castellón	2,32E+04	1,73E+06	1,31E+06	-1,71E+00	2,71E+00
Ciudad Real	-2,48E-09	9,32E+05	7,02E+05	-9,58E-01	2,58E+00
Córdoba	-4,97E-09	9,79E+05	7,72E+05	-7,15E-01	1,68E+00
Coruña (A)	-4,97E-09	1,90E+06	1,53E+06	-8,53E-01	2,52E+00
Cuenca	1,04E+04	5,92E+05	4,45E+05	-1,48E+00	2,57E+00
Girona	1,25E+05	1,07E+07	8,59E+06	-6,38E+00	2,39E+00
Granada	-7,45E-09	1,59E+06	1,28E+06	-8,76E-01	2,22E+00
Guadalajara	6,80E+03	4,46E+05	3,65E+05	-7,61E-01	1,96E+00
Guipúzcoa	6,36E+04	2,62E+06	2,12E+06	-1,37E+00	1,70E+00
Huelva	3,21E+04	1,31E+06	1,05E+06	-1,31E+00	2,28E+00
Huesca	1,54E+04	8,19E+05	5,96E+05	-1,77E+00	2,27E+00
Jaén	-3,73E-09	8,11E+05	6,57E+05	-5,22E-01	1,92E+00
León	-1,86E-09	1,07E+06	8,18E+05	-1,37E+00	2,54E+00
Lleida	-1,24E-09	1,33E+06	1,16E+06	-1,48E+00	1,18E+00
Lugo	-1,55E-09	6,57E+05	5,12E+05	-1,14E+00	2,03E+00
Madrid	-2,98E-08	7,38E+06	6,14E+06	-4,82E-01	1,61E+00
Málaga	1,70E+05	4,27E+06	3,71E+06	-1,34E+00	2,76E+00
Murcia	-2,48E-09	2,46E+06	2,02E+06	-6,98E-01	1,76E+00
Navarra	5,65E+04	2,41E+06	1,93E+06	-1,41E+00	2,72E+00
Ourense	-3,10E-10	6,07E+05	4,55E+05	-1,08E+00	2,25E+00
Palencia	7,06E+03	3,80E+05	2,95E+05	-1,10E+00	1,57E+00
Pontevedra	-6,21E-10	1,87E+06	1,41E+06	-1,34E+00	2,28E+00
Rioja (La)	-1,86E-09	5,70E+05	4,62E+05	-9,08E-01	3,15E+00
Salamanca	8,86E+03	7,89E+05	6,23E+05	-1,48E+00	1,91E+00
Segovia	2,00E+03	3,81E+05	2,89E+05	-1,64E+00	2,25E+00
Sevilla	-1,24E-09	2,00E+06	1,58E+06	-7,26E-01	1,14E+00
Soria	4,59E+03	2,20E+05	1,75E+05	-1,24E+00	1,57E+00
Tarragona	8,43E+04	3,09E+06	2,43E+06	-2,37E+00	3,13E+00
Teruel	5,78E+03	4,25E+05	3,33E+05	-1,80E+00	2,10E+00
Toledo	1,64E+04	1,10E+06	8,92E+05	-6,52E-01	1,74E+00
Valencia	-2,48E-09	3,72E+06	3,01E+06	-6,02E-01	1,89E+00
Valladolid	-1,24E-09	7,44E+05	6,17E+05	-6,92E-01	1,60E+00
Vizcaya	-6,21E-09	1,26E+06	1,09E+06	-4,04E-01	1,50E+00
Zamora	-1,09E-09	4,88E+05	3,61E+05	-2,01E+00	2,23E+00
Zaragoza	-7,45E-09	1,40E+06	1,17E+06	-5,70E-01	1,39E+00

APPENDIX TABLE II. SELECTED SARIMA MODELS FOR FORECASTING PACKS

Province	ME	RMSE	MAE	MPE	MASE
Alava	1,76E-14	3,78E-01	3,19E-01	-8,26E-01	1,35E+00
Albacete	9,69E-03	4,86E-01	3,85E-01	-7,16E-01	1,69E+00
Alicante	2,58E-02	7,44E-01	6,36E-01	-1,01E+00	2,97E+00
Almería	8,99E-03	6,42E-01	5,21E-01	-1,11E+00	3,44E+00
Asturias	-1,17E-15	4,12E-01	3,26E-01	-7,59E-01	2,29E+00
Ávila	8,21E-03	9,38E-01	7,44E-01	-3,01E+00	2,82E+00
Badajoz	-1,28E-15	4,30E-01	3,46E-01	-8,53E-01	1,99E+00
Balears	3,70E-02	1,12E+00	9,20E-01	-2,33E+00	2,83E+00
Barcelona	-1,31E-15	3,62E-01	2,98E-01	-6,84E-01	1,62E+00
Burgos	-2,11E-15	5,57E-01	4,48E-01	-1,43E+00	2,93E+00
Cáceres	3,48E-14	6,14E-01	4,77E-01	-1,38E+00	2,27E+00
Cádiz	1,10E-02	4,81E-01	3,72E-01	-1,86E+00	1,87E+00
Cantabria	1,00E-03	5,94E-01	4,44E-01	-1,31E+00	2,64E+00
Castellón	1,04E-02	7,84E-01	5,94E-01	-1,68E+00	2,69E+00
Ciudad Real	-5,52E-14	4,83E-01	3,64E-01	-9,65E-01	2,63E+00
Córdoba	-1,92E-15	3,21E-01	2,53E-01	-7,14E-01	1,70E+00
Coruña (A)	-1,13E-14	4,13E-01	3,32E-01	-8,47E-01	2,51E+00
Cuenca	1,28E-02	7,25E-01	5,44E-01	-1,47E+00	2,58E+00
Girona	4,26E-02	3,63E+00	2,93E+00	-6,20E+00	2,66E+00
Granada	-2,61E-15	4,51E-01	3,58E-01	-8,75E-01	2,23E+00
Guadalajara	6,55E-03	4,50E-01	3,66E-01	-7,54E-01	1,99E+00
Guipúzcoa	2,19E-02	9,24E-01	7,42E-01	-1,35E+00	1,80E+00
Huelva	1,58E-02	6,55E-01	5,25E-01	-1,29E+00	2,48E+00
Huesca	1,73E-02	9,17E-01	6,69E-01	-1,74E+00	2,29E+00
Jaén	3,01E-13	3,27E-01	2,65E-01	-5,16E-01	1,97E+00
León	-5,04E-14	5,40E-01	4,15E-01	-1,36E+00	2,59E+00
Lleida	-2,92E-15	7,98E-01	6,94E-01	-1,48E+00	1,21E+00
Lugo	-1,42E-15	4,61E-01	3,59E-01	-1,13E+00	2,02E+00
Madrid	-1,31E-15	2,93E-01	2,45E-01	-4,94E-01	1,47E+00
Málaga	2,58E-02	6,66E-01	5,81E-01	-1,33E+00	3,38E+00
Murcia	3,57E-13	4,44E-01	3,69E-01	-7,06E-01	1,73E+00
Navarra	2,16E-02	9,57E-01	7,69E-01	-1,41E+00	3,19E+00
Ourense	-1,55E-15	4,52E-01	3,41E-01	-1,06E+00	2,35E+00
Palencia	1,06E-02	5,60E-01	4,35E-01	-1,09E+00	1,57E+00
Pontevedra	-1,02E-15	4,93E-01	3,72E-01	-1,33E+00	2,23E+00
Rioja (La)	-5,18E-16	4,60E-01	3,75E-01	-9,08E-01	2,79E+00
Salamanca	6,41E-03	5,65E-01	4,47E-01	-1,47E+00	1,91E+00
Segovia	3,26E-03	6,05E-01	4,59E-01	-1,63E+00	2,23E+00
Sevilla	-7,77E-16	2,75E-01	2,17E-01	-7,33E-01	1,16E+00
Soria	1,21E-02	5,82E-01	4,62E-01	-1,24E+00	1,54E+00
Tarragona	2,66E-02	9,92E-01	7,81E-01	-2,32E+00	3,16E+00
Teruel	1,06E-02	7,68E-01	6,02E-01	-1,78E+00	2,08E+00
Toledo	6,18E-03	4,19E-01	3,36E-01	-6,47E-01	1,82E+00
Valencia	3,33E-16	3,85E-01	3,13E-01	-5,98E-01	1,98E+00
Valladolid	-2,78E-16	3,56E-01	2,95E-01	-6,88E-01	1,61E+00
Vizcaya	5,18E-16	2,76E-01	2,39E-01	-4,08E-01	1,44E+00
Zamora	1,39E-14	6,40E-01	4,76E-01	-1,99E+00	2,50E+00
Zaragoza	-1,33E-15	3,72E-01	3,11E-01	-5,76E-01	1,42E+00

APPENDIX TABLE III. SELECTED SARIMA MODELS FOR FORECASTING PER CAPITA PACKS

Province	ME	RMSE	MAE	MPE	MASE
Alava	4,57E+02	1,76E-14	3,78E-01	3,19E-01	1,35E+00
Albacete	3,11E+02	9,69E-03	4,86E-01	3,85E-01	1,69E+00
Alicante	3,87E+02	2,58E-02	7,44E-01	6,36E-01	2,97E+00
Almería	2,73E+02	8,99E-03	6,42E-01	5,21E-01	3,44E+00
Asturias	2,95E+02	-1,17E-15	4,12E-01	3,26E-01	2,29E+00
Ávila	3,20E+02	8,21E-03	9,38E-01	7,44E-01	2,82E+00
Badajoz	2,91E+02	-1,28E-15	4,30E-01	3,46E-01	1,99E+00
Balears	5,11E+02	3,70E-02	1,12E+00	9,20E-01	2,83E+00
Barcelona	2,66E+02	-1,31E-15	3,62E-01	2,98E-01	1,62E+00
Burgos	4,78E+02	-2,11E-15	5,57E-01	4,48E-01	2,93E+00
Cáceres	3,13E+02	3,48E-14	6,14E-01	4,77E-01	2,27E+00
Cádiz	2,29E+02	1,10E-02	4,81E-01	3,72E-01	1,87E+00
Cantabria	2,98E+02	1,00E-03	5,94E-01	4,44E-01	2,64E+00
Castellón	3,30E+02	1,04E-02	7,84E-01	5,94E-01	2,69E+00
Ciudad Real	2,92E+02	-5,52E-14	4,83E-01	3,64E-01	2,63E+00
Córdoba	3,02E+02	-1,92E-15	3,21E-01	2,53E-01	1,70E+00
Coruña (A)	2,67E+02	-1,13E-14	4,13E-01	3,32E-01	2,51E+00
Cuenca	4,12E+02	1,28E-02	7,25E-01	5,44E-01	2,58E+00
Girona	6,02E+02	4,26E-02	3,63E+00	2,93E+00	2,66E+00
Granada	2,84E+02	-2,61E-15	4,51E-01	3,58E-01	2,23E+00
Guadalajara	2,96E+02	6,55E-03	4,50E-01	3,66E-01	1,99E+00
Guipúzcoa	4,44E+02	2,19E-02	9,24E-01	7,42E-01	1,80E+00
Huelva	3,33E+02	1,58E-02	6,55E-01	5,25E-01	2,48E+00
Huesca	3,50E+02	1,73E-02	9,17E-01	6,69E-01	2,29E+00
Jaén	3,63E+02	3,01E-13	3,27E-01	2,65E-01	1,97E+00
León	2,72E+02	-5,04E-14	5,40E-01	4,15E-01	2,59E+00
Lleida	5,40E+02	-2,92E-15	7,98E-01	6,94E-01	1,21E+00
Lugo	3,12E+02	-1,42E-15	4,61E-01	3,59E-01	2,02E+00
Madrid	2,61E+02	-1,31E-15	2,93E-01	2,45E-01	1,47E+00
Málaga	3,27E+02	2,58E-02	6,66E-01	5,81E-01	3,38E+00
Murcia	2,55E+02	3,57E-13	4,44E-01	3,69E-01	1,73E+00
Navarra	4,87E+02	2,16E-02	9,57E-01	7,69E-01	3,19E+00
Ourense	2,16E+02	-1,55E-15	4,52E-01	3,41E-01	2,35E+00
Palencia	4,47E+02	1,06E-02	5,60E-01	4,35E-01	1,57E+00
Pontevedra	2,29E+02	-1,02E-15	4,93E-01	3,72E-01	2,23E+00
Rioja (La)	4,12E+02	-5,18E-16	4,60E-01	3,75E-01	2,79E+00
Salamanca	4,52E+02	6,41E-03	5,65E-01	4,47E-01	1,91E+00
Segovia	3,46E+02	3,26E-03	6,05E-01	4,59E-01	2,23E+00
Sevilla	2,55E+02	-7,77E-16	2,75E-01	2,17E-01	1,16E+00
Soria	3,95E+02	1,21E-02	5,82E-01	4,62E-01	1,54E+00
Tarragona	4,21E+02	2,66E-02	9,92E-01	7,81E-01	3,16E+00
Teruel	3,73E+02	1,06E-02	7,68E-01	6,02E-01	2,08E+00
Toledo	3,07E+02	6,18E-03	4,19E-01	3,36E-01	1,82E+00
Valencia	2,92E+02	3,33E-16	3,85E-01	3,13E-01	1,98E+00
Valladolid	3,98E+02	-2,78E-16	3,56E-01	2,95E-01	1,61E+00
Vizcaya	2,73E+02	5,18E-16	2,76E-01	2,39E-01	1,44E+00
Zamora	3,05E+02	1,39E-14	6,40E-01	4,76E-01	2,50E+00
Zaragoza	3,26E+02	-1,33E-15	3,72E-01	3,11E-01	1,42E+00

Acknowledgment

The authors are thanked for the support of Antonio Golpe, full professor at the University of Huelva and professor of Big Data Analysis in the Master's in Business Intelligence at Universidad Internacional de La Rioja (UNIR).

References

[1] I. Papanicolas,L.R. Woskie and A.K. Jha, "Health care spending in the United States and other high-income countries". *Jama*, vol. 319, no. 10, pp. 1024-1039, 2018, doi: 10.1001/jama.2018.1150.

[2] P. Cadahia, A. Golpe, J.M. Martín-Álvarez and E. Asensio, "Measuring anomalies in cigarette sales using official data from Spanish provinces: Are the anomalies detected by the Empty Pack Surveys (EPSs) used by Transnational Tobacco Companies (TTCs) the only anomalies?". *Tobacco Induced Diseases*, vol. 19, no. 98, 2021, doi: 10.18332/tid/143321.

[3] A. Almeida, A. Golpe and J.M. Martín-Álvarez, "A spatial analysis of the Spanish tobacco consumption distribution: Are there any consumption clusters?". *Public Health*, vol. 186, 2020, doi: 10.1016/j.puhe.2020.06.040.

[4] J.M. Martín-Álvarez, A. Golpe, J. Iglesias and R. Ingelmo, "Price and income elasticities of demand for cigarette consumption: what is the association of price and economic activity with cigarette consumption in Spain from 1957 to 2016?", *Public Health*, vol. 185, 275-282, 2020, doi: 10.1016/j.puhe.2020.05.059.

[5] J.M. Martín-Álvarez, A. Almeida, A. Galiano, A. and A. Golpe, "Asymmetric behavior of tobacco consumption in Spain across the business cycle: a long-term regional analysis", *International Journal of Health Economics and Management*, vol. 20, 391-421, 2020, doi: 10.1007/s10754-020-09286-y.

[6] J.M. Martín-Álvarez., A. Almeida, A. Golpe, and J.C. Vides, "The influence of cigarette price on the cigarette consumption in Spain: a Logarithmic Mean Divisia Index analysis from 1957 to 2018", *Revista Espanola de Salud Publica*, vol. 95, 2021, e202102026.

[7] A. Almeida, A. Golpe, J. Iglesias and J.M. Martín-Álvarez, "The price elasticity of cigarettes: new evidence from Spanish regions, 2002–2016", *Nicotine and Tobacco Research*, vol. 23, no. 1, 48-56, 2021, doi: 10.1093/ntr/ntaa131.

[8] R. Fu, A. Kundu, N. Mitsakakis, T. Elton-Marshall, W. Wang, S. Hill and M.O. Chaiton, "Machine learning applications in tobacco research: a scoping review". *Tobacco Control,* vol. 32, no. 1, pp. 99-109, 2023, doi: 10.1136/tobaccocontrol-2020-056438.

[9] K.P., Murphy, "Machine learning: a probabilistic perspective". *MIT press*, 2012.

[10] A.L. Beam and I.S. Kohane, "Big data and machine learning in health care". *Jama*, vol. 319, no. 13, pp. 1317-1318, 2018, doi: 10.1001/jama.2017.18391.

[11] T. Hastie, R. Tibshirani and J.H. Friedman, "The elements of statistical learning: data mining, inference, and prediction". *Springer New York*, vol. 2, pp. 1-758, 2009, doi: 10.1007/978-0-387-21606-5.

[12] J.M. Reps, P.R. Rijnbeek and P.B. Ryan, "Supplementing claims data analysis using self-reported data to develop a probabilistic phenotype model for current smoking status". *Journal of Biomedical Informatics*, vol. 97, pp. 103264, 2019, doi: 10.1016/j.jbi.2019.103264.

[13] P. Mamoshina, K. Kochetov, F. Cortese, A. Kovalchuk, A. Aliper, E. Putin and A. Zhavoronkov, "Blood biochemistry analysis to detect smoking status and quantify accelerated aging in smokers". *Scientific Reports*, vol. 9, no. 1, pp. 1-10, 2019, doi: 10.1038/s41598-018-35704-w.

[14] S. Huda, J. Yearwood and R. Borland, "Cluster based rule discovery model for enhancement of government's tobacco control strategy". *4th International Conference on Network and System Security IEEE*, pp. 383-390, 2010, doi:10.1109/NSS.2010.14.

[15] N. Kim, D.E. McCarthy, W.Y. Loh, J.W. Cook, M.E. Piper, T.R. Schlam and T.B. Baker, "Predictors of adherence to nicotine replacement therapy: Machine learning evidence that perceived need predicts medication use". *Drug and Alcohol Dependence*, vol. 205, pp. 107668, 2019, doi: 10.1016/j.drugalcdep.2019.107668

[16] A. Dumortier, E. Beckjord, S. Shiffman and E. Sejdić, "Classifying smoking urges via machine learning". *Computer Methods and Programs in Biomedicine*, vol. 137, pp. 203-213, 2016, doi: 10.1016/j.cmpb.2016.09.016.

[17] L.N. Coughlin, A.N. Tegge, C.E. Sheffer and W.K. Bickel, "A machine-learning approach to predicting smoking cessation treatment outcomes". *Nicotine and Tobacco Research*, vol. 22, no. 3, pp. 415-422, 2020, doi: 10.1093/ntr/nty259.

[18] K. Davagdorj, J.S. Lee, K.H. Park and K.H. Ryu, "A machine-learning approach for predicting success in smoking cessation intervention". *10th International Conference on Awareness Science and Technology IEEE*, pp. 1-6, 2019, doi: 10.1109/ICAwST.2019.8923252.

[19] A. Singh and H. Katyan, "Classification of nicotine-dependent users in India: a decision-tree approach". *Journal of Public Health*, vol. 27, no. 4, pp. 453-459, 2019, doi: 10.1007/s10389-018-0973-x.

[20] L. Clancy, S. Gallus, J. Leung and C.O. Egbe, "Tobacco and COVID-19: Understanding the science and policy implications". *Tobacco Induced Diseases*, vol. 18., 2020, doi: 10.18332/tid/131035.

[21] Y. Saloojee and A. Mathee, "COVID-19 and a temporary ban on tobacco sales in South Africa: impact on smoking cessation". *Tobacco Control*, vol. 31, no. 2, pp. 207-210, 2022, doi: 10.1136/tobaccocontrol-2020-056293.

[22] B.P. Lee, J.L. Dodge, A. Leventhal, N.A. Terrault, "Retail alcohol and tobacco sales during COVID-19". Annals of internal medicine, vol. 174, no. 7, pp. 1027-1029, 2021, doi: 10.7326/M20-7271.

[23] D. Yach, "Tobacco use patterns in five countries during the COVID-19 lockdown". *Nicotine & Tobacco Research*, vol. 22, no. 9, pp. 1671-1672, 2020, doi: 10.1093/ntr/ntaa097.

[24] P. Driezen, K.A. Kasza, S. Gravely, M.E. Thompson, G.T. Fong, K.M. Cummings and A. Hyland, "Was COVID-19 associated with increased cigarette purchasing, consumption, and smoking at home among US smokers in early 2020? Findings from the US arm of the International Tobacco Control (ITC) Four Country Smoking and Vaping Survey". *Addictive Behaviors*, vol. 129, pp. 107276, 2022, doi: 10.1016/j.addbeh.2022.107276.

[25] S. Asare, A. Majmundar, F. Islami, P. Bandi, S. Fedewa, L.J. Westmaas and N. Nargis, "Changes in cigarette sales in the United States during the COVID-19 pandemic". *Annals of Internal Medicine*, vol. 175, no. 1, pp. 141-143, 2022, doi: 10.7326/M21-3350.

[26] J. Kim and S. Lee, "Impact of the COVID-19 pandemic on tobacco sales and national smoking cessation services in Korea". *International Journal of Environmental Research and Public Health*, vol. 19, no. 9, pp. 5000, 2022, doi: 10.3390/ijerph19095000.

[27] I.B., Ahluwalia, M. Myers and J.E. Cohen, "COVID-19 pandemic: an opportunity for tobacco use cessation". *The Lancet Public Health*, vol. 5, no. 11, pp. e577, 2020, doi: 10.1016/S2468-2667(20)30236-X.

[28] M. Hefler and C.E. Gartner, "The tobacco industry in the time of COVID-19: time to shut it down?". *Tobacco Control*, vol. 29, no. 3, pp. 245-246, 2020, doi: 10.1136/tobaccocontrol-2020-055807.

[29] T.K. Burki, "Tobacco industry capitalises on the COVID-19 pandemic". *The Lancet Respiratory Medicine*, vol. 9, no. 10, pp. 1097-1098, 2021, doi: 10.1016/S2213-2600(21)00361-1.

[30] R. Álvarez, N. Vicente, L. Polo, P. Ríos, P. Ferrández, A.M. Furió, O Monteagudo, R. Dalmau, J. Doncel, S. Justo, J. Rey, C. González and C. Gómez-Chacón, "Tobacco use in Spain during COVID-19 lockdown: an evaluation through social media". *Revista Espanola de Salud Publica*, vol. 95, 2021, PMID: 33724261.

[31] E.P. Esplá, C.C. Faus, A.J. Baldó, I.B. Enrique and E.C. Vives, "COVID-19 and smoking: an opportunity to quit". *Archivos de Bronconeumologia*, vol. 57, no. 12, pp. 784, 2021, doi: 10.1016/j.arbr.2021.10.009.

[32] J.M. Suelves, B. Gomez-Zuniga and M. Armayones, "Changes in smoking behaviour due to the COVID-19 pandemic in Spain". *Tobacco Prevention & Cessation*, vol. 7(Supplement), no. 55, 2021, doi: 10.18332/tpc/143664.

[33] A. Estévez-Danta, L. Bijlsma, R. Capela, R. Cela, A. Celma, F. Hernández and J.B. Quintana, "Use of illicit drugs, alcohol and tobacco in Spain and Portugal during the COVID-19 crisis in 2020 as measured by wastewater-based epidemiology". *Science of the Total Environment*, vol. 836, pp. 155697, 2022, doi:10.1016/j.scitotenv.2022.155697.

[34] C. Martínez-Cao, L. de La Fuente-Tomas, I. Menéndez-Miranda, A. Velasco, P. Zurrón-Madera, L. García-Álvarez and J. Bobes, "Factors associated with alcohol and tobacco consumption as a coping strategy to deal with the coronavirus disease (COVID-19) pandemic and lockdown in Spain". *Addictive Behaviors*, vol. 121, pp. 107003, 2021, doi: 10.1016/j.addbeh.2021.107003.

[35] J.C. Vázquez, J. C. and D. Redolar-Ripoll, "COVID-19 outbreak impact in

Spain: A role for tobacco smoking?". *Tobacco Induced Diseases*, vol. 18, no. 30, 2020, doi: 10.18332/tid/120005.

[36] J.C. Vázquez and D. Redolar-Ripoll,"Epidemiological data from the COVID-19 outbreak in Spain for the promotion of tobacco smoking cessation policies". *Tobacco Use Insights*, vol. 13, pp. 1179173X20924028, 2020, doi: 10.1177/1179173X20924028.

[37] O. Parra, C. Suárez and L. Martínez, "Análisis comparativo de las técnicas de series de tiempo ARIMA y ANFIS para pronosticar tráfico Wimax", *Ingeniería*, vol. 12, no. 2, 73-79, 2007, doi: 10.14483/23448393.2166.

[38] R.A. Yaffee, and M. McGee, "An introduction to time series analysis and forecasting: with applications of SAS® and SPSS®". *Elsevier*, 2000.

[39] E.P. Box George, M. Jenkins Gwilym, C. Reinsel Gregory and M. Ljung Greta, "Time series analysis: forecasting and control". *San Francisco: Holden Bay*, 1979.

[40] M.S.M. Kasihmuddin, M.A. Mansor, S. A. Alzaeemi and S. Sathasivam, "Satisfiability logic analysis via radial basis function neural network with artificial bee colony algorithm". *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 6, 2021, doi: 10.9781/ijimai.2020.06.002.

[41] A. Gupta, K. Ghanshala and R.C. Joshi, "Machine Learning Classifier Approach with Gaussian Process, Ensemble boosted Trees, SVM, and Linear Regression for 5G Signal Coverage Mapping". *International Journal of Interactive Multimedia and Artificial Intelligence*, Vol. 6, No 6, 2021, doi: 10.9781/ijimai.2021.03.004.

[42] R Core Team, "R: A language and environment for statistical computing". *R Foundation for Statistical Computing, Vienna, Austria*, 2021, https://www.R-project.org/

[43] RStudio Team. "RStudio: Integrated Development for R". *RStudio, PBC, Boston, MA URL*, 2020, http://www.rstudio.com/

[44] H. Wickham, M. Averick, J. Bryan, W. Chang, L.D. McGowan, R. François, G. Grolemund, A. Hayes, L. Henry, J. Hester, M. Kuhn, T.L. Pedersen, E. Miller, S.M. Bache, K. Müller, J. Ooms, D. Robinson D, D.P. Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo, H. Yutani, "Welcome to the tidyverse." *Journal of Open Source Software*, vol. 4, no. 43, pp. 1686, 2019, doi: 10.21105/joss.01686.

[45] R.J. Hyndman, Y. Khandakar, "Automatic time series forecasting: the forecast package for R". *Journal of Statistical Software*, vol. 27, no. 3, pp. 1–22, 2008, doi: 10.18637/jss.v027.i03.

[46] R. Hyndman, G. Athanasopoulos, C. Bergmeir, G. Caceres, L. Chhay, M. O'Hara-Wild, F. Petropoulos, S. Razbash, E. Wang, F. Yasmeen, "forecast: Forecasting functions for time series and linear models". *R package version 8.16*, 2022, https://pkg.robjhyndman.com/forecast/

[47] R. Gomajee, H. Torregrossa, C. Bolze, M. Melchior and F.E.K. Lesueur, "Decrease in cross-border tobacco purchases despite intensification of antitobacco policies in France". *Tobacco Control*, vol. 30, no. 4, pp. 428-433, 2021, doi: 10.1136/tobaccocontrol-2019-055540.

[48] A. Golpe, J.M. Martín-Álvarez, A. Galiano and E. Asensio, "Effect of IQOS introduction on Philip Morris International cigarette sales in Spain: a Logarithmic Mean Divisa Index decomposition approach", *Gaceta Sanitaria*, vol. 36, 293-300, 2022, doi: 10.1016/j.gaceta.2021.12.007.

[49] P. Cadahia, A. Golpe, J.M. Martín-Álvarez, E. Asensio, "The importance of price, income, and affordability in the demand for cigarettes in Spain", *Addicta: The Turkish Journal on Addictions*, vol. 9, no. 3, 241-251, 2022, doi: 10.5152/ADDICTA.2022.22054.

Andoni Andueza

Andoni Andueza is a Graduated in Business Administration by University and Msc in Business Intelligence by Universidad Internacional de La Rioja (UNIR). His master's thesis focused on the analysis of time series. He has developed different dashboard for any companies for business analysis. He has worked in University of Mondragon (MU) as People Analyst. Currently, he is People Analyst at ULMA.

Miguel Ángel Del Arco-Osuna

Miguel Ángel Del Arco-Osuna is a Telecommunication Engineer by University of Seville (US) and Graduated in Business Administration by Distance Learning University (UNED). He is currently professor in Quantitative Methods for Economics and Business at Universidad Internacional de La Rioja (UNIR) and PhD Candidate in Economics.

Bernat Fornés

Bernat Fornés is a Graduated in Business Administration by University of Vic (UVic-UCC) and Msc in Business Intelligence by Universidad Internacional de La Rioja (UNIR). For the Graduated in Business Administration, his final project was related to the analysis of different areas of Fintech. His master's thesis focused on the analysis of time series. He has developed different dashboard for any companies for business analysis. Currently, he is treasurer for a local council.

Rubén González Crespo

Dr. Rubén González Crespo has a PhD in Computer Science Engineering. Currently he is Vice Chancellor of Academic Affairs and Faculty from UNIR and Global Director of Engineering Schools from PROEDUCA Group. He is advisory board member for the Ministry of Education at Colombia and evaluator from the National Agency for Quality Evaluation and Accreditation of Spain (ANECA). He is member from different committees at ISO Organization. Finally, He has published more than 200 papers in indexed journals and congresses.

Juan Manuel Martín-Álvarez

Dr. Juan Manuel Martín-Álvarez has a Phd in Economics with focus in quantitative analysis for decision making. He has extensive experience as a teacher in public and private universities in the areas of Accounting, Finance, Statistics and Econometrics. He is Associate Professor in Quantitative Methods for Economics and Business at Universidad Internacional de La Rioja (UNIR). Currently, he is head of the Msc in Business Intelligence at Universidad Internacional de La Rioja. Finally, he has published more than 10 Health Economics papers with special focus on tobacco use in indexed journals.

Blockchain Based Cloud Management Architecture for Maximum Availability

Alberto Arias Maestro[1]*, Oscar Sanjuan Martinez[1], Ankur M. Teredesai[2], Vicente García-Díaz[3]

[1] Universidad Internacional de La Rioja, Logroño (Spain)
[2] University of Washington, Tacoma, WA (USA)
[3] University of Oviedo, Oviedo (Spain)

Received 10 May 2022 | Accepted 20 January 2023 | Early Access 1 February 2023

Abstract

Contemporary cloud application and Edge computing orchestration systems rely on controller/worker design patterns to allocate, distribute, and manage resources. Standard solutions like Apache Mesos, Docker Swarm, and Kubernetes can span multiple zones at data centers, multiple global regions, and even consumer point of presence locations. Previous research has concluded that random network partitions cannot be avoided in these scenarios, leaving system designers to choose between consistency and availability, as defined by the CAP theorem. Controller/worker architectures guarantee configuration consistency via the employment of redundant storage systems, in most cases coordinated via consensus algorithms such as Paxos or Raft. These algorithms ensure information consistency against network failures while decreasing availability as network regions increase. Mainstream blockchain technology provides a solution to this compromise while decentralizing control via a fully distributed architecture coordinated through Byzantine-resistant consensus algorithms. This research proposes a blockchain-based decentralized architecture for cloud resource management systems. We analyze and compare the characteristics of the proposed architecture concerning the consistency, availability, and partition resistance of architectures that rely on Paxos/Raft distributed data stores. Our research demonstrates that the proposed blockchain-based decentralized architecture noticeably increases the system availability, including cases of network partitioning, without a significant impact on configuration consistency.

Keywords

Blockchain, Cloud Computing, Distributed Systems, Paxos, Raft.

DOI: 10.9781/ijimai.2023.02.002

I. Introduction

Contemporary cloud application and Edge computing management systems rely on centralized architectures to distribute and manage application configuration across the network [1]. The most prevalent implementations are designed around the controller/worker pattern, in which a controller node receives one or more requests and then communicates with worker nodes to execute them. In this architecture, the controller and worker constantly run a loop to ensure that the controller has an up-to-date view of the system and that the worker receives the latest scheduled configuration. The controller/worker pattern allows system designers to simplify the scheduling and allocation of resources by assuming that a consistent global state view is available to the controller nodes.

Intrinsic to the centralized architecture design is the requirement to implement strong security measures. It only requires the security compromise of the controller nodes in the system to take control of the entire network. It is common for system designers to isolate controller nodes from the application data plane [2] to restrict orchestrated application access to the control plane, further increasing the deployment complexity across network boundaries [3]-[4].

* Corresponding author.
E-mail address: alberto.arias@gmail.com

However, as these systems' topological complexity and scale increase, many questions arise, such as latency, reliability, and load balancing (Fig. 1). In most cases, as a single point of failure, the controller is replicated and strategically placed to minimize the impact of hardware failures [5]. As the number of zones increases, so does the number of controller replicas, thus increasing the system's overall fragility.

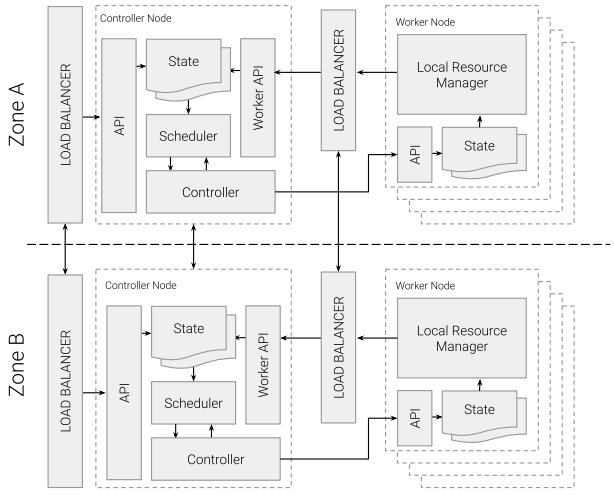


Fig. 1. Typical redundant Controller/Worker architecture.

System designers rely on data storage solutions to guarantee configuration consistency across controller nodes, therefore inheriting their underlying consistency and availability characteristics. System availability against machine failures is addressed through controller active redundancy [6] across regions or zones. However, this topology makes network partitions more likely, forcing system designers to choose between consistency and availability. According to the CAP theorem [7], any distributed data store can provide only two of three guarantees (Fig. 2): consistency, availability, or partition resistance.

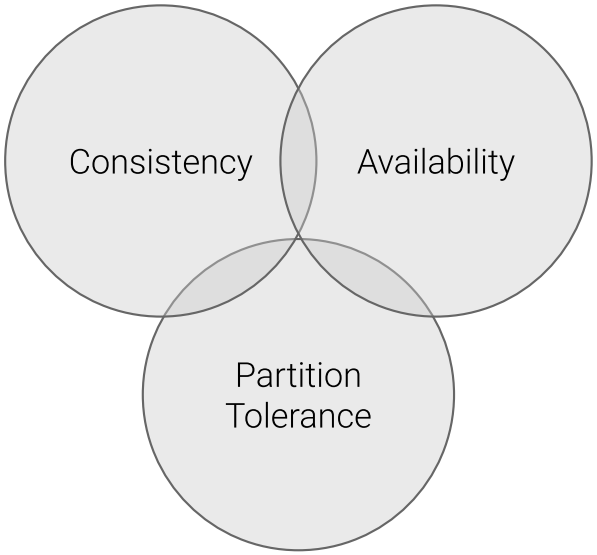


Fig. 2. CAP Theorem lemma [4].

Because system designers cannot prevent network failures [8], the compromise between consistency and availability is typically addressed by implementing consensus schemes such as the Paxos and Raft algorithms [9]. Architectures based on these algorithms require that most control nodes are available, and those worker nodes can connect to one of those nodes to ensure access to the most recent view of the system. If a controller network connection is interrupted, it can no longer perform its designated function.

However, modern cloud-based applications are typically designed to satisfy the need for scale, availability, and globally distributed access. These applications are designed to be resilient against transient failures and do not require absolute consistency of the control plane data to ensure availability across fragile global environments. Instead, those applications can benefit from increased availability of the underlying control system to ensure the triggering of actions when failures or scaling events occur.

Mainstream blockchain technology can provide an alternative solution by decentralizing the control plane with a fully distributed architecture that relies on the eventual consistency achieved through Byzantine-resistant consensus algorithms [10] and strong security enforced via defined cryptographic rules. Because of distributed consensus, all nodes in the network can validate the order of cryptographically signed system management transactions to reach an agreement on which application configuration blocks to add to the blockchain, including scenarios where parallel chains evolve during a network partition event. In this scenario, the control plane is available if any node in the network is reachable, functioning, and capable of recording transactions onto the longest known blockchain, maximizing availability in place of consistency. In contrast, controller/worker architecture's availability depends on having access to a controller node, even if the underlying storage systems were configured to use eventual consistent consensus schemes.

The main contributions of our work are (a) a design for a decentralized hybrid control/worker node, (b) a set of transaction validation rules for system state information, and (c) three theorems (maximum availability, eventual consistency, partition primacy) from the properties of the proposed system. The remainder of this paper is organized as follows: Section II explores related work and existing research. Section III describes the proposed integrated architecture for hybrid control/work nodes, the structure of the proposed blockchain, and validation logic. Our results are discussed in Section IV, and conclusions and suggestions for future research are presented in Section V.

II. Related Work

State-of-the-art application management technologies simplify automation via declarative configuration, where state updates are propagated over time in what is known as intent-record consistency. This means that the system will eventually reflect the most current configuration as scheduled by a central controller. The system records any requests submitted to be later processed by the controller nodes.

Examples of systems based on controller/worker architecture include Cloud Foundry [11], Apache Mesos [12], Docker Swarm [13], and Kubernetes [14]. As previously stated, the architecture of these systems prioritizes intent-record consistency and availability through controller replication [15].

Apache Mesos, Docker Swarm, and Kubernetes store configuration state in Etcd, a key-value store, using the Raft consensus algorithm to ensure consistency and partition resistance. Essentially, the controller returns the confirmation to the client only when a quorum of storage nodes coordinated by an algorithmically selected leader acknowledges the request. Reads are linearizable, implying that once a write is completed, all later read should return the value of that write or the value of the last write. Alternatively, Cloud Foundry utilizes MySQL, a relational database that relies on the Paxos algorithm. However, in practical terms, the only difference between Paxos and Raft is the leader's election mechanism [16].

For example, when a user submits an intent request, the desired configuration change is first stored in either Etcd or MySQL. Depending on the system, the transaction is then confirmed to the user, who reasonably expects the request to be distributed and committed. Once the configuration change is committed, the controller can execute the scheduling algorithm and communicate the changes to the affected worker nodes to achieve a consistent global state that matches the user's intentions [17]. These mechanisms, in aggregate, provide intent-record state consistency that guarantees high statistical availability and good network partition resistance if the controllers can connect to storage nodes and the storage nodes can achieve a quorum (Table I).

TABLE I. Partition Resistance Examples of Paxos/Raft

Servers	Quorum	Failure Tolerance[1]
1	1	0
2	2	0
3	2	1
4	3	1
5	3	2
6	4	2
7	4	3
8	5	4

[1] Server failure or networked partitioned

In the case of network partition across data center zones or regions (Table III), nodes placed in a partition outside the quorum cannot be managed or provide system updates, leading to potential outages. For example, an application might not be able to react to an autoscaling event, or a node failure cannot be redeployed. Raft and Paxos drive consistency and availability (up to a few failed nodes proportional to the number of replicas) from the point of view of an external consumer with equal access to all the replicas. In most cases, replicas are collocated with the nodes. If a replica loses network access, collocated nodes cannot be operated.

To date, little practical research has been performed to weaken the criteria for replica consistency to improve the partition tolerance, availability, and performance of cloud systems owing to the non-monotonic nature of the system configuration. Non-monotonicity occurs when a new configuration change request alters the previous configuration state request [18]. Consequently, request ordering determines the global state of the system. However, because of the characteristics of the eventual system consistency described previously, a system of rules that disambiguates potentially conflicting configuration requests can provide acceptable levels of consistency. For the most part, system operators prioritize their focus on the system's final state and, in most cases, can infer the consequences of intermediate states during configuration changes.

A well-defined set of transaction ordering rules implemented as a cryptographic protocol and persisting results on a blockchain presents an opportunity to leverage mainstream consensus algorithms to solve the challenges presented by the CAP theorem.

The feasibility of implementing blockchain technology control systems has been demonstrated using a multi-tier architecture to record and distribute configurations across multiple control nodes [16]. Existing implementations leverage smart contracts to substitute access control and preserve the sequence of change requests. However, deployment control is still delegated to a traditional controller/worker cluster architecture. While a fully distributed blockchain across all regions can yield similar results concerning global availability, it still depends on the availability of the local cluster controller to ensure all nodes can be operated, therefore not impacting the CAP properties of the system or the security of the control nodes.

Concerning blockchain performance, previous work has determined that the throughput characteristics of three-tier control systems utilizing a general-purpose blockchain as a record store yield good results [19].

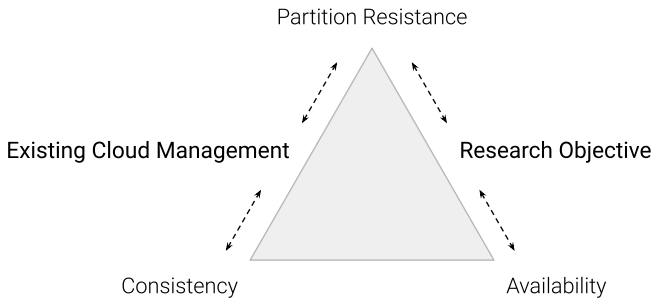


Fig. 3. Focus on Availability and Partition Resistance.

The objective of this research (Fig. 3) is to evaluate the implementation of a highly available and partition resistant [18] cloud management system offering a solution that utilizes a purpose-built blockchain to store system state to record configuration efficiently and in a verifiable and permanent manner [20].

This research evolves previous approaches by integrating control and work nodes into a single hybrid component and using Byzantine resistance consensus algorithms to coordinate the blockchain's agreement, termination, and validity.

Existing blockchains implementation like Ethereum, Cardano, Solana, Hyperledger, or any other general-purpose blockchain with support for smart contracts can be used to manage and execute purpose-built smart contracts containing the logic of configuration disambiguation, scheduling, and access control. However, using existing blockchains will require a network of Oracles capable of performing active functions, including failure detection. In addition, to ensure the same level of availability, it would require every node running the software to also operate as a general-purpose blockchain node alongside the required Oracles. We decided against this approach due to the runtime, management, and overhead. Although outside the scope of this research, we consider implementing the solution using general-purpose smart contract blockchains worth studying for Web3 applications that rely on both traditional stacks and smart contracts. Future research will evaluate and compare the overhead costs of running a general-purpose vs. purpose-built blockchain to be deployed to each node.

III. Proposed Framework

This proposal is structured into three sections. First, we cover the architecture of the hybrid controller/worker node and its connectivity to other nodes. The second section describes how the system state configuration is encoded into the blockchain structure, followed by global ordering rules that ensure transaction validity to be applied by participating nodes.

A. System State Blockchain

In blockchain-centric systems, a natural pattern is decentralizing control and replacing authority with Byzantine-resistant consensus patterns [21]. Applying this pattern to the cloud management space may seem unintuitive at first glance, yet this solution addresses the primary goals of this research.

This yields a highly available peer-to-peer architecture [22] of compute nodes collectively converging into a state that matches the sequence of intents stored in the blockchain. While the primary function of nodes is to host workloads, nodes maintain a full copy of the blockchain and participate in the consensus process as both block creators and validators.

Nodes are connected to other nodes using a peer-to-peer (P2P) gossip protocol. When a node is added to the network, the initial discovery of peer nodes is performed using dynamic DNS. Once a node is connected to other nodes, it can receive a list of other known nodes and blocks. In addition, the node can validate the list of known nodes obtained with the configuration stored in the blockchain. There may be additional security warranties when adding a node to the network depending on the consensus algorithm, for example, client certificate authentication in Proof of Authority schemes.

The nodes receive direct connections from users. Users submit new transactions and inspect the state of the node and the last known system state, according to the longest chain stored by that node. Additionally, nodes are assigned the responsibility of communicating with external services, for example, updating a DNS entry or configuring a new load balancer.

We incorporate the existing Kubernetes architecture elements for the proposed solution, such as the Kubelet component, which provides the actuation of state configuration changes by communicating with the node host operating system. As such, the components of a hybrid node include (Fig. 4):

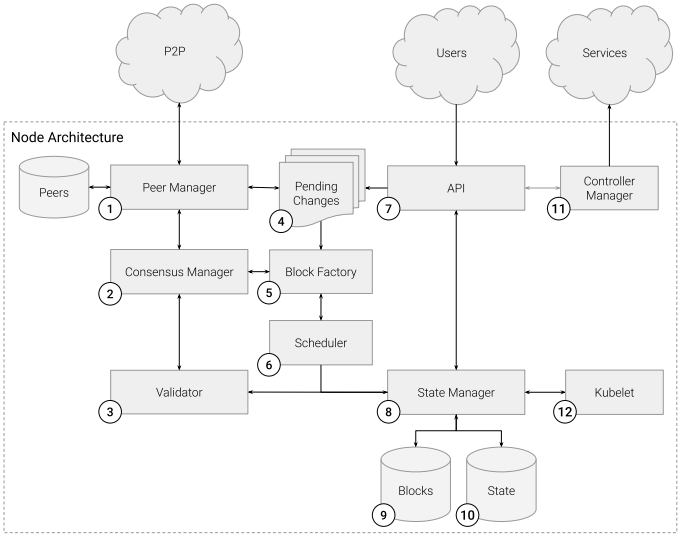


Fig. 4. Decentralized blockchain/worker node architecture.

1. The peer manager is responsible for maintaining a list of known peers. It creates and maintains TCP connections and receives new connections from peers. The peer manager communicates with other nodes via the P2P gossip protocol.
2. The consensus manager is dedicated to applying consensus rules to maintain the longest valid chain known by the node by determining which blocks should be added to the chain or even discarding dead-end chains. The consensus manager is integrated very closely with the peer manager, such that it can adapt the node chain to new information, including blocks and alternative chains. In addition, a node, depending on the consensus algorithm, may be selected for mining a new block. The consensus manager communicates the new block to the other peer nodes.
3. The validator is responsible for analyzing the contents of a block and ensuring that all new transactions are valid. Transaction order, transaction inputs and outputs, locking script execution, and other block rules are related to the consensus algorithm.
4. Pending changes comprise a list of known pending transactions. Each block maintains a list of pending transactions. When a new block is received or minted, the transactions in the block are removed from the pending list.
5. The block factory is responsible for mining a new block based on the inputs of the pending change list. It communicates with the consensus manager, ensuring that the block is valid by verifying with the validator. Any invalid transactions are reported until a block is valid and ready to be communicated.
6. The scheduler is a component that watches for newly created resources with no assigned nodes.
7. API is the front end of the contents of the state of the cluster and transaction management. Users connect to the node via an API to interact with the cluster without directly operating a node.
8. State Manager maintains the databases and indexes required to store and operate the cluster.
9. Blocks are key-value pair databases indexing every block and transaction of the blockchain by its hash value.
10. State is a document-oriented database with content resulting from executing all transactions in the blockchain.
11. The controller manager is responsible for maintaining the configuration and state of the services external to the cluster.
12. Kubelet is part of the Kubernetes architecture. It is responsible for connecting to the Docker runtime and ensuring that all pods and containers run according to the cluster state determined by the blockchain.

B. Blockchain Structure

Transaction data is stored in blocks organized into a linear sequence (Fig. 5). New transactions are added to the blocks, and blocks are added at the end of the blockchain. Each block indirectly contains the hash of each transaction calculated by adding every transaction to a Merkle tree and storing the root. Additionally, every block includes the hash of the previous block's header. In essence, every time a block is added to the chain, the harder it is to change or remove previous blocks, and every transaction in the blockchain is irreversible and final.

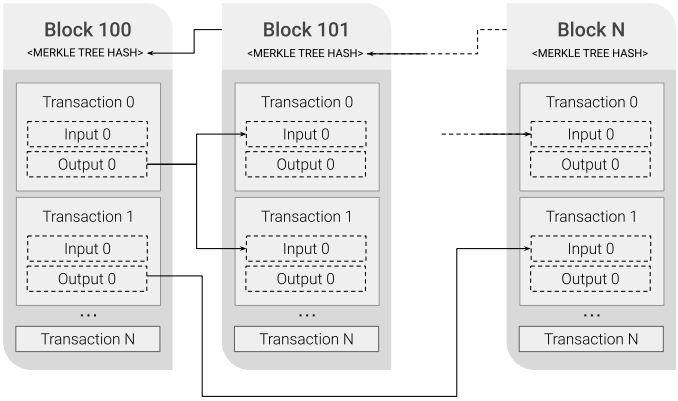


Fig. 5. Blockchain structure.

Block transactions contain each of the changes in the system's configuration submitted by users. The structure and content are like those utilized in cryptocurrency ledgers, with differences compared to the input and output of the transactions referring to hierarchical resource definitions. For example, to create a new resource within a folder or namespace, the transaction input must meet two conditions: a reference to the most recent transaction with a parent resource as an output and a script or data satisfying the requirements of the input transaction script.

When a user submits a transaction, it is possible to refer to a parent resource directly or indirectly to satisfy the validation requirements.

The input of direct access transactions refers to the most recent transaction with the targeted resource as the output. It is the user's responsibility to identify the latest transaction and produce the input script data that meets the requirements of the script securing the resource.

The input of indirect access transactions refers to a transaction used to create or update a hierarchical resource. For example, starting a new deployment refers to the output of the transaction used to create or update the namespace where it would be contained.

C. Transaction Validation

All nodes check every transaction during the block forming process. A block is constructed by assembling ordered transactions from the pending transaction list. The selection of transactions to be included in a block is critical for the system design. Transaction order is performed using both topological and canonical rules.

Topological ordering by ordering transactions according to their positions in the resource hierarchy. Topological ordering ensures that sequential transactions that depend on a previous transaction that manages a parent resource are evaluated to maximize transaction validity. For example, a resource cannot be created until a parent resource is created.

Canonical ordering is performed when two resources have equivalent inputs and outputs in a resource hierarchy. When this occurs, transactions are ordered by transaction ID, calculated as the SHA256 of the transaction data. Canonical ordering ensures that the output is unique and deterministic given the same set of transactions. In other words, given the same set of unordered transactions, the result after ordering would be the same regardless of who performs the ordering or when the operation is performed.

Additionally, when two chains evolve independently due to a network partition event, nodes that adopt a new longer chain would move transactions on the shorter chain to the pending changes list and be evaluated accordingly.

IV. Results and Discussion

The proposed architecture provides the foundation for a fully distributed configuration management system that stores the global configuration in a blockchain structure and is distributed across all the nodes in the network. This architecture solution offers improved network-partitioning resistance and availability.

Network partitioning occurs when a group of nodes is isolated and cannot communicate with the remaining nodes in the network. This is a common scenario when those nodes are not in the same data center or the data center is partitioned into two or more availability zones. Note that in the proposed architecture, when a network partition occurs, there is a risk that transactions submitted to the partition with the shortest chain will become invalid once the network connectivity is restored. The transactions are appended to the Pending Changes list (Fig. 6).

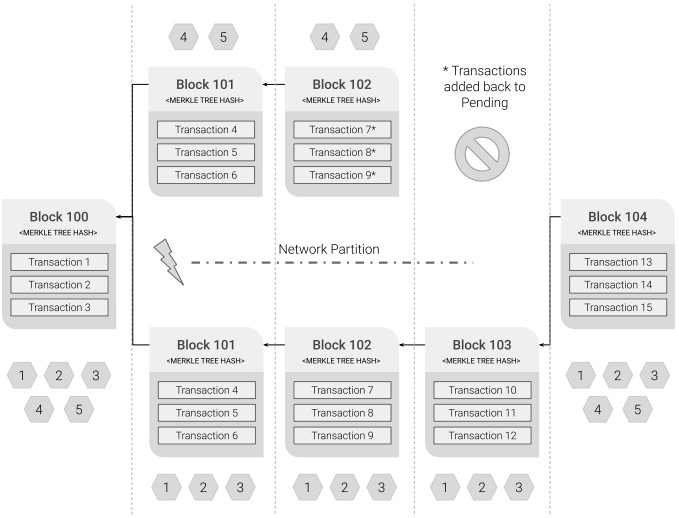


Fig. 6. Chain resolution after Network Partition.

So far, we have discussed the core components and behaviors of the system. From the analysis conducted throughout this study, we can deduce that the system meets the following propositions:

Proposition 1: Any node can accept a transaction.

Proposition 2: A single node can add a block to the chain.

Proposition 3: Nodes do not require connection to other nodes to accept transactions.

Proposition 4: A group of nodes (more than one node), where each node can connect to others, will generate a chain faster than a group with fewer nodes.

Proposition 5: A node will always accept the longest chain available.

Therefore, we can formulate the following three theorems by the principle of mathematical logic.

1. Theorem: Maximum Availability

If a node is available, the system is available.

Proof of Theorem 1. P1 ∧ P2 ∧ P3 ⟹ T1. If any node can accept a transaction (Proposition 1), and a single node can add a block to the chain (Proposition 2), and nodes do not require the connection to other nodes to accept transactions (Proposition 3), then if a node is available, the system is available.

2. Theorem: Eventual Consistency

A transaction can only be considered irreversibly committed when it is part of a block in the longest chain, and is part of the current chain for most of the nodes in the network.

Proof of Theorem 2. P3 ∧ P4 ⟹T2. If nodes do not require the connection to other nodes to accept transactions (Proposition 3), and a group of nodes (more than one node), where each node can connect to others, will generate a chain faster than a group with fewer nodes (Proposition 4), then a transaction can only be considered irreversibly committed when it is part of a block that is in the longest chain, and it is part of the current chain for most of the nodes in the network.

3. Theorem: Partition Primacy

A network partition with the majority of nodes generates the longest chain with irreversibly committed transactions.

Proof of Theorem 3. P4 ∧ P5 ⟹T3. If a group of nodes (more than one node), where each node can connect to others, will generate a chain faster than a group with fewer nodes (Proposition 4), and a node will always accept the longest chain available (Proposition 5), then a network partition with the majority of nodes generates the longest chain with irreversibly committed transactions.

4. Examples

Traditional Paxos/Raft-based systems are available if most replica nodes are available to achieve quorum and maintain the configuration store consistency (Table II). When there are three zones, both systems are reliable when one fault occurs. However, the differences are revealed when two Paxos/Raft replicas fail, preventing the system from achieving a quorum and leading to system failure. Note that in this proposal (Table III), only users who can access a partition with available nodes will be able to submit transactions.

TABLE II. Availability Examples of Paxos/Raft

Zones/Replicas	Replica Faults	Partitions	Paxos/Raft
3 / 3	1	0	Available
3 / 3	2	0	Fault
3 / 3	0	2	Fault
9 / 9	4	0	Available
9 / 9	5	0	Fault
9 / 9	0	3	Fault
3 / 3	1	0	Available
3 / 3	2	0	Fault

Additionally, as stated in the Partition Primacy and Eventual Consistency theorems, only nodes in the largest partition will be able to confirm transactions irreversibly.

TABLE III. Availability Examples of the Proposed Solution

Zones/Replicas	Replica Faults	Partitions	Proposed
3 / 300	50 / 50 / 50	0	Available
3 / 300	100 / 0 / 0	0	Available[1]
3 / 300	100 / 100 / 100	0	Fault
3 / 300	100 / 0 / 0	1	Available[1]
3 / 300	50 / 50 / 50	1	Available[2]
3 / 300	50 / 50 / 50	2	Available[2]
3 / 300	50 / 50 / 50	3	Available[2]

[1] Not accessible from failed partitions.
[2] Transactions cannot be considered irreversible until restored.

In the Paxos/Raft system, when the number of zones is expanded to nine, and thus, the number of replicas, the statistical availability increases dramatically. However, in cases where multiple network partitions occur, the system can become unavailable because of the inability of replicas to talk to each other and thus prevent a quorum, even with no replica failures. As stated in the theorem of maximum availability, our proposal becomes unavailable only when all the nodes fail.

V. Conclusions and Future Research

In the proposed decentralized architecture, the system is available as long as the nodes are accessible to the user. However, the intent-record consistency is compromised and replaced with eventual consistency. In essence, a user querying a different node that received the change might obtain a response that does not include the most recent change, that is, until that change is broadcast through the network and adopted in a block that is part of the longest computed chain. This scenario, we believe, is an acceptable compromise.

Integrating the blockchain node capabilities, scheduler, and container management agent reduces management overhead by reducing the number of software components to be deployed and managed. Since our proposal does not allow the execution of general-purpose smart contracts, the security surface is reduced, and configuration management operations costs stay constant.

Minting an additional block to the blockchain is perhaps the most critical operation [23] to meet the desired consistency and performance requirements. In future research, we will analyze different algorithms that can potentially be used to ensure that blocks are minted, validated, and added to the blockchain throughout the network while minimizing the trust required. In essence, these algorithms enable the capability to achieve consensus on which blocks to add to the chain based on rules that ensure fairness and security for all participants. Examples of these algorithms are as follows:

1. Proof of Work (PoW) is a consensus algorithm based on demonstrable computational effort across a fixed time window, forcing each party to upfront a total energy/computational cost proportional to their weight on the consensus effort.
2. Proof of Space (PoS) is a consensus algorithm based on demonstrable storage capacity requiring every participant to pre-compute and store an established function output. Participants must be able to prove knowledge of that output at any time, ensuring a commitment to integrity by upfronting the storage cost.
3. Proof of Authority (PoA) is a consensus mechanism based on the proven identity of the participants. This algorithm requires establishing a level of trust across the participants.
4. Proof of Stake (PoS) is a consensus algorithm based on demonstrable funds requiring all participants to deposit a monetary amount in an escrow account controlled by a cryptographic protocol.

It should be noted that the proposed architecture integrates data and control planes, thereby forcing the re-evaluation of existing security threat models. Future research should compare current architectures to secure control and application data.

References

[1] M. Schwarzkopf, A. Konwinski, M. Abd-El-Malek, and J. Wilkes, "Omega, Flexible, scalable schedulers for large compute clusters", in Proceedings from the European Conference on Computer Systems, Prague, Czech Republic, 2013, pp. 351-364, doi: 10.1145/2465351.2465386.

[2] G. Dasher, I. Envid, and B. Calder, "Architectures for Protecting Cloud Data Planes", Google, Mountain View, CA, USA, 2022. Accessed: Nov. 15, 2022. [Online]. Available: https://arxiv.org/abs/2201.13010, doi: 0.48550/arXiv.2201.13010.

[3] A. Kumar, S. Avinash Kumar, V. Dutt, A. Dubey, S. Narang, "A Hybrid Secure Cloud Platform Maintenance Based on Improved Attribute-Based Encryption Strategies", International Journal of Interactive Multimedia and Artificial Intelligence, In Press, pp. 1-8, 2021, doi: 10.9781/ijimai.2021.11.004.

[4] G. Zhang, X. Chen, L. Zhang, B. Feng, X. Guo, J. Liang, Y. Zhang, "STAIBT: Blockchain and CP-ABE Empowered Secure and Trusted Agricultural IoT Blockchain Terminal", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 7, no. 5, pp. 66-75, 2022, doi: 10.9781/ijimai.2022.07.004.

[5] A. Berenberg, and B. Calder, "Deployment Archetypes for Cloud Applications", ACM Computing Surveys, vol. 55, no. 3, pp. 1-48, 2022, doi:10.48550/arXiv.2105.00560.

[6] S. Sebastio, R. Ghosh, and T. Mukherjee, "An availability analysis approach for deployment configurations of containers", IEEE Transactions on Services Computing, vol. 14, no. 1, pp. 16-29, 2018, doi:10.1109/TSC.2017.2788442.

[7] E. Brewer, "Spanner, truetime and the cap theorem", Google, Mountain View, CA, USA, 2022. Accessed: Nov. 15, 2022. [Online]. Available: https://research.google/pubs/pub45855.

[8] P. Bailis, and K. Kingsbury, "The network is reliable: An informal survey of real-world communications failures", Queue, vol. 12, no. 7, pp. 20-32, 2014, doi:10.1145/2639988.2655736.

[9] L. Lamport, "The part-time parliament", ACM Transactions on Computer System, vol. 16, no. 2, pp 133-169, 1998, doi:10.1145/3335772.3335939.

[10] V. Gramoli, "From blockchain consensus back to Byzantine consensus", Future Generation Computer Systems, vol. 107, no. C, pp. 760-769, 2020, doi:10.1016/j.future.2017.09.023.

[11] D. Bernstein, "Cloud Foundry Aims to Become the OpenStack of PaaS", in IEEE Cloud Computing, vol. 1, no. 2, pp. 57-60, 2014, doi:10.1109/MCC.2014.32.

[12] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. Joseph, R. Katz, S. Shenker, and I. Stoica, "Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center", 8th USENIX Symposium on Networked Systems Design and Implementation, vol. 11, pp. 22-22, 2011.

[13] N. Naik, "Building a virtual system of systems using docker swarm in multiple clouds", IEEE International Symposium on Systems Engineering (ISSE), pp. 1-3, 2016, doi: 10.1109/SysEng.2016.7753148.

[14] B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, "Borg, omega, and Kubernetes", Communications of the ACM, vol. 59, no. 5, pp. 50-57, 2016, doi:10.1145/2890784.

[15] S. Davidson, "Optimism and consistency in partitioned distributed database systems", ACM Transactions on Database Systems (TODS), vol. 9, no. 3, pp. 456-481, 1984, doi:10.1145/1270.1499.

[16] H. Howard, M. Schwarzkopf, A. Madhavapeddy, and J. Crowcroft, "Raft refloated: Do we have consensus?", ACM SIGOPS Operating Systems Review, vol 49, no. 1, pp. 12-21, 2015, doi:10.1145/2723872.2723876.

[17] B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, "Borg, Omega, and Kubernetes: Lessons learned from three container-management systems over a decade", Queue, vol. 14, no. 1, pp. 70–93, 2016, doi:10.1145/2898442.2898444.

[18] J. Hellerstein, and P. Alvaro, "Keeping CALM: when distributed consistency is easy", Communications of the ACM, vol. 63, no. 9, pp. 72-81, 2020, doi: 10.48550/arXiv.1901.01930.

[19] J. Yang, J. Dai, H. B. Gooi, H. Nguyen and A. Paudel, "A Proof-of-Authority Blockchain Based Distributed Control System for Islanded Microgrids", IEEE Transactions on Industrial Informatics, vol. 18, no. 11, pp. 8287-8297, 2022, doi:10.1109/TII.2022.3142755.

[20] P. K. Sharma, M. Chen and J. H. Park, "A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT", IEEE Access, vol. 6, pp. 115-124, 2018, doi: 10.1109/ACCESS.2017.2757955.

[21] G. Nguyen, and K. Kim, "A survey about consensus algorithms used in blockchain", Journal of Information processing systems, vol. 14, no. 1, pp. 101-128, 2018, doi:10.3745/JIPS.01.0024.

[22] I. Weber, V. Gramoli, A. Ponomarev, M. Staples, R. Holz, A. Tran, and P. Rimba, "On availability for blockchain-based systems", IEEE 36th Symposium on Reliable Distributed Systems (SRDS), Hong Kong, China, pp. 64-73, 2017, doi:10.1109/SRDS.2017.15.

[23] G. Carrara, L. Burle, D. Medeiros, C. Vinicius, and D. Mattos, "Consistency, availability, and partition tolerance in blockchain: a survey on the consensus mechanism over peer-to-peer networking", Annals of Telecommunications, vol. 75, no. 3, pp. 163-174, 2020.

Alberto Arias Maestro

Alberto Arias Maestro is a PhD student at Universidad Internacional de La Rioja. He has 20+ years of experience in large-scale software system design and development. Previously, he led teams at Google Cloud for open-source tech and multi-cloud interoperability. He founded ElasticBox, a multi-cloud app management startup. Prior to ElasticBox, he served as VP of Architecture at DynamicOps and led the design of a private cloud platform used by Fortune 500 firms. He holds a Master's in computer science from Pontifical University of Salamanca.

Oscar Sanjuan Martinez

Dr. Oscar Sanjuan Martinez is a professor in the Department of Computer Science at the Universidad Internacional de La Rioja, and he is currently a VP of Engineering at Lumen. Before joining Lumen, he was an interim associate professor at the University of Oviedo and Director of the R + D + I Office at the Pontifical University of Salamanca. He also led the Software Engineering Department at the Pontifical University of Salamanca as a Director for the Madrid Campus and Author Biography as a professor in the Department of Languages, Computer Systems, and Software Engineering. His current research interests include cloud computing, intelligent agents, and blockchain systems.

Ankur Teredesai

Prof. Ankur Teredesai is a full professor of computer science and systems at the School of Engineering & Technology, University of Washington. Today, healthcare technology solutions are complex, with an increasing emphasis on AI-driven software. Dr. Teredesai's research on AI regulation of solutions for the personalization of decisions in healthcare has widespread application. He is an invited member of a global industrial and governmental partnership, pushing the boundaries of innovation and policy in this field. Prof. Teredesai has published 100+ papers on machine learning, and his work has been deployed across various industries (advertising, recommendation systems, and global health systems). This work has been recognized in popular press as well as in academic citations. Since 2009, his research contributions have advanced our understanding of the risk and utilization of chronic conditions such as diabetes and heart failure. In 2015, after years of collaborative and applied research on large clinical and claims datasets, Prof. Teredesai founded KenSci, a spin-off at the University of Washington, which was acquired in 2021. Prof. Teredesai served as the Information Officer for ACM SIGKDD (Special Interest Group in Knowledge Discovery and Data Mining) from 2006 to 2018 and as the past general chair of KDD 2019. He is currently an associate editor for ACM SIGKDD Explorations and serves several program committees of conferences on AI and machine learning. Prof. Teredesai is an active advocate and mentor for non-traditional female students to pursue computing careers.

Vicente García-Díaz

Dr. Vicente García-Díaz is an associate professor in the Department of Computer Science at the University of Oviedo, Spain. He is a software engineer with a Ph.D. in computer science. He has a master's in occupational risk prevention and qualifies as a university expert in blockchain application development. He is part of the editorial and advisory boards of several indexed journals and conferences and has been the editor of several special issues in books and indexed journals. He has supervised 100+ academic projects and has published 100+ research papers in journals, conferences, and books. His teaching interests are primarily the design and analysis of algorithms and the design of domain-specific languages. His current research interests include decision-support systems, health informatics, and e-learning. Engineer, Ph.D. in Computer Science. He has a master's in occupational risk prevention and qualifies as a university expert in blockchain.

An Efficient Probabilistic Methodology to Evaluate Web Sources as Data Source for Warehousing

Hariom Sharan Sinha[1], Saket Kumar Choudhary[2], Vijender Kumar Solanki[3]*

[1] Department of Computer Science & Engineering, Adamas University, Barasat, Kolkata, West Bengal (India)
[2] Department of Computer Science & Engineering, GITAM University, Bengaluru, Karnataka (India)
[3] Department of Computer Science & Engineering, CMR Institute of Technology, Hyderabad, Telangana (India)

Received 23 May 2022 | Accepted 1 February 2023 | Early Access 24 February 2023

Abstract

Internet is the largest source of data and the requirement of data analytics have fueled the data warehouse to switch from structured conventional Data Warehouse to complex Web Data Warehouse. The dynamic and complex nature of web poses various types of complexities during synthesis of web data into a conventional warehouse. Multi-Criteria-Decision Making (MCDM) is a prominent mechanism to select the best data for storing into the data-warehouse. In this article, a method, based on the probabilistic analysis of SAW and TOPSIS methods, has been proposed to select web data sources as data sources for web data warehouse. This method deals more efficiently with the dynamic and complex nature of web. Here, the result of the selection employs the analysis of both the methods (SAW and TOPSIS) to evaluate the probability of selection of respective score (1-9) for each feature. With these probability values, the probability of selection of the next web sources has been be determined. Moreover, using the same probability values, mean score and standard deviation of the scores of respective features of selected web sources have been deduced, which are further used to fix the standard score of each feature for selection of web sources. The standard score is a parameter of the proposed Mean-Standard-Deviation (MSD) method to check the suitability of web sources individually, whereas others do the same on comparative basis. The proposed method cuts down the cost of the repetitive comparison operation, once after computation of the Standard score using Mean and Standard deviation of each individual feature. Here, the respective value of the standard score of each feature is only compared with the score of each respective feature of the next web sources, so it reduces the cost of computation and selects the web sources faster as well.

Keywords

Mean-Standard-Deviation (MSD) Method, Multi-Criteria Decision Method (MCDM), Probabilistic Method, Standard Deviation of Score, Web Source.

DOI: 10.9781/ijimai.2023.02.012

I. Introduction

The evolution of Internet as well as the ease to share and to fetch the data from web have made web a magnificent platform of sources for information. Web is an independent platform to get and provide nearly all the types of information. At the mean time the requirement of data analytics for decision support system has obligated the data warehouse to deal with web data rather than traditional data, because the data from local data sources has turned insufficient for decision support systems. Despite being the data easily and publicly available on web, the web data cannot be queried and manipulated efficiently for data analytics as done in traditional Data Warehouse. So, the efficient way to use the data for data analytics is to exploit the warehouse technique rather than directly access the data. The Data warehouse main obligation is to collect information from various data sources to create repository and make integrated information available for Decision Support Systems. However, the exponential rising of web sources and complexity as well as dynamic nature of web have posed new challenges for data warehouse to deal with web-data from various and independent web sources [18], [22], [26], [27].

To find the suitable data to systematically incorporate it into a warehouse is an anticipating approach for data analytics. In order to collect the data for data warehouse, finding the relevant data on web is just as to find out needle in a haystack because of so many web sources [25], [26]. Besides, the dynamic nature of web data has made the situation more complicated and complex. So, the very first task for web warehouse is to find the relevant web sources as the data source for it. Thus, there is the requirement of evaluation of the relevancy and compatibility of the web sources. Various features must be entertained during evaluation of web sources. Zhu et al. have classified the features into three categories viz. web sources stability, web data quality and contextual issues of web data [25].

According to Zooknic statistics (http://www.zooknic.com/Domains/counts.html) on 15 December 2009, the total number of worldwide registered domains was 111,889,734, and these 111 million (around) websites are owned by government, private or individual organization and agencies, which causes complex (structured, semi-structured, unstructured) natured data designed in different (heterogeneous) styles. Besides so large number of websites, web sources are dynamic, the web data is updated frequently as well as even millions of new

* Corresponding author.
E-mail address: spesinfo@yahoo.com

web sources and web pages are being added every day on Internet. So already available sources may change or even disappear.

Another challenge is the quality of web-data because the web techniques are so opened and independent that web masters can fire whatever data they like on web. A big amount of data on the web is not properly examined, retrospected and percolated as done in conventional publications. Wrong, inconsistent, incomplete or vague data are easily available on web and even correct data are not properly presented. So, the quality of data on web is maverick. Third challenge is the context of data should fulfill the requirements of the user, because the availability of data on web is with the intention of browsing usually rather than for warehousing and analysis. So, the web-data must fulfill the requirements of web-warehousing, as relevancy of web data for analysis, easily extraction of necessitated data, all-important metadata (data definition, data format and derivation rules) etc. Probably these requirements may not be fulfilled.

Therefore, the designer of web-warehouse must build a set of features to evaluate the web sources to select the most suitable sources as data source for warehousing. In this article we will look into these challenges and discuss the methods for relevant web sources selection for warehousing. Firstly, a set of selection features is formulated and then evaluation of the web sources has been performed using Multi-Criteria Decision Method (MCDM) approach, (especially SAW and TOPSIS methods) with respect to these features. Again statistical and probabilistic analysis of the selected web sources has been done with respect to the score of the corresponding features. Then mean and standard deviation of the score of the corresponding features have been evaluated. Now using mean and standard deviation the relevancy of web source can be computed without any further relative comparison of web sources. Here only a fixed number of comparisons as the number of features and one more with the threshold value are required. So the computational complexity of the proposed method becomes constant.

The rest of the paper is organized as follows: Section II presents related work on web source selection as data source for warehousing and various approaches including MCDM (SAW and TOPSIS specially) which is based on evaluation of web sources. Section III explains the complexity during web sources selection and set of features for web source evaluation. Section IV explicates SAW and TOPSIS methods of MCDM, for selection of web sources as data source for warehouse. In Section V the proposed work has been explained. This section consists of three parts viz. statistical analysis of SAW and TOPSIS, Probability of selection of new web sources and Mean-Standard-Deviation (MSD) method based on mean and standard deviation. Section VI analyses the experimental setup and results of SAW, TOPSIS and MSD Methods and at last, Section VII presents the conclusion.

II. Related Work

During incorporating the data from web into warehouses, the dynamic and complex nature of web [2], [3], [6], [8], [9], [14] poses various challenges. Different approaches have been developed to overcome the challenges during warehousing the web data [2], [6], [17], [25], [29]. Doan et al. have explained XML technologies to extract, incorporate, store, query and analyze web data as well as their application to data warehouse [6]. Boussaid et al. haves proposed a UML (Unified Modeling Language) and XML model of warehousing along with the attributes of XML [2]. Hao Fan used HDM (Hypergraph Data Model) for warehousing the web data [8].

Another approach is comparative analysis of web sources to select the best one. In order to select the web source as data source for warehouse, quality of data available on web source is an important criteria of source selection. To define the quality of data, multiple features of web sources are entertained [20], [25]. So, the selection of a web source is multi features selection task [16]. To deal with the multi features selection problem [20], [25], Multi criterion Decision Making (MCDM) [13], [23], [33], [34] methods have been employed. With this method, on the basis of score and weight of features, the comparative analysis has been done. Having multiple criteria of decision, the MCDM approach is applicable in various real problems besides ranking of web sources [25] like [4] and many other problems. Le et al. [11] proposed a dynamic approach of web data warehousing using object oriented methodology to design the logical level for apprehending and presenting basic semantics of web sources and user requirements in a flexible and sensible way.

Moreover, Dong et al. have proposed a marginalism approach to select the web source. The marginalism approach is based on the marginalism principle of economics [12]. It restricts the selection of a new source till the marginal benefit is more than the marginal cost of integration. The marginal benefit is here the difference between benefit after and before the new source integration. Similarly marginal cost is the difference between the cost after and before the integration [7].

III. Features to Evaluate Web Sources

The evaluation features of web sources have been roughly classified into three major categories viz. web sources stability, web data quality and contextual issues of web data [18], [22], [25], [28].

A. Web Source Stability

This selection features can be further subcategorized into availability, durability, accessibility, and refreshing rate.

- Availability defines whether the specific site is up and in running mode, its response time and also reachability of the pages through the links.
- Durability defines the time period by which the data is made available on the website. Historical data may or may not be available on the website. So, the volatile data must be extracted and warehoused for the purpose of availability [20], [25].
- Accessibility checks whether the data has been accessed without breaching any authenticity norms (registration or password) during the automatic extraction for warehousing [20, 25].
- Refresh rate defines the timeliness by which the data is made available on the website, at the meantime fast refresh rate means volatile data is overwritten quickly, so must be extracted with the same rate to make it available for data analytics [20], [25], [28].

B. Web Data Quality

This selection feature can be further split into Origination, Objectivity, Accurateness, Completeness, and Metadata. Origination usually refers to data lineage, i.e. origin of the data. Objectivity concerns with deficiency of biasness in the data. Accurateness concerns with the accuracy of web data, i.e. error free data. Completeness concerns with the coverage, whereas Metadata concerns with the derivation rules and interpretation of web data [20], [25], [28], [30].

C. Contextual Issues of Web Data

This feature can be further split into three sub-categories viz. Relevancy, Timeliness, Layout. Relevancy is the most important feature to select the web source, as how much the specific data is relevant for data analytics. Timeliness concerns with how timely the data is made available on the website. Layout defines different formats of data presentation like XML, HTML, pdf, docs, pictures, audio, video or any other representation [20], [25].

IV. Evaluation and Selection of Web Source Using MCDM Methods (SAW and TOPSIS)

Zhu et al. [25] proposed four approaches to select the Web sources in compensatory methods viz. Simple Additive Weighing (SAW), Analytic Hierarchy Process (AHP), Data Envelopment Analysis (DEA) and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS). Here SAW and AHP come under the scoring group, DEA under the concordance group while TOPSIS comes under the compromising group [13]. This section presents SAW and TOPSIS methods to statically analyze the source selection.

A. Simple Additive Weighing (SAW) Method

In this method for every feature of the web sources, some weight has been provided with the constraint that the sum of the weights of all the features must be 1. For example, four web sources *WS*1, *WS*2, *WS*3 and *WS*4, and twelve features have been assumed as shown in the following Table I.

TABLE I. Weights of Quality Features

Feature Symbol	Features	Weight
F1	Availability	0.07
F2	Durability	0.08
F3	Accessibility	0.09
F4	Refreshing Rates	0.07
F5	Origination	0.10
F6	Objectivity	0.07
F7	Accurateness	0.11
F8	Completeness	0.06
F9	Metadata	0.08
F10	Relevancy	0.10
F11	Timeliness	0.08
F12	Layout	0.09

In the SAW method, no standard scale has been defined for rating i.e. for giving a score, so it is defined by a decision maker. In this example, the minimum and maximum scale for score have been taken 1 and 9 respectively. Table II shows the performance score of the different web sources with respect to each feature.

TABLE II. Scores of the Different Web Sources With Regard to Each Feature

FS	WS1	WS2	WS3	WS4
F1	8	9	7	4
F2	6	1	9	8
F3	8	3	6	1
F4	4	3	2	2
F5	4	1	6	5
F6	7	7	6	1
F7	1	3	5	6
F8	4	8	7	9
F9	5	3	1	6
F10	8	4	1	3
F11	2	3	4	5
F12	5	8	5	4

Then

$$SAWi = \sum_{j=1}^{N} c_{ij} w_j; i = 1,2,\dots M \quad (1)$$

Where SAW_i: the SAW score of i^{th} web source; M: number of web sources; N: number of features; C_{ij}: score of i^{th} source in j^{th} feature; w_j: weight of j^{th} feature [4], [13], [15], [21]. Applying formula given in Eq. (1) to Table II, we find ranking score as SAW(W S1) = 5.09, SAW(W S2) = 4.19, SAW(W S3) = 4.83 and SAW(W S4) = 4.91. Here, Web Source WS1 is the best source for warehousing.

B. Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) Method

This method was formulated by Hwang and Yoon as mentioned in the research article of Zhu et al. [25]. The fundamental approach of this method is to get an alternate solution in multi-dimensional computational area, such as; the solution is nearest to the ideal solution and farthest to the negative solution. The multi-dimensional computational area is defined by taking set of features as dimensions. Here the ideal solution is the positive extreme solution with a set of possible best synthetically scores with regard to each feature. Similarly, the negative ideal solution is the negative extreme solution with a set of possible worst scores. These two (ideal and negative ideal) solutions in computing area, are two points with extreme values as dimensions. This method has five steps to evaluate the best source [4], [12], [13], [21], [25]. They, with explanations taking the aforementioned example, are as follows:

1. Normalize the decision matrix.

$$y_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{M} x_{ij}^2}} \quad (2)$$

Where X_{ij} is the performance score of i^{th} Web Source in terms of j^{th} feature; M is the number of Web Sources. The values of the result are shown in Table III.

TABLE III. Normalized Decision Matrix

FS	WS1	WS2	WS3	WS4
F1	0.5521	0.6211	0.4830	0.2760
F2	0.4447	0.0741	0.6671	0.5930
F3	0.7628	0.2860	0.5721	0.0953
F4	0.6963	0.5222	0.3482	0.3482
F5	0.4529	0.1132	0.6794	0.5661
F6	0.6025	0.6025	0.5164	0.0861
F7	0.1187	0.3560	0.5934	0.7121
F8	0.2760	0.5521	0.4830	0.6211
F9	0.5394	0.3560	0.1187	0.7121
F10	0.8433	0.4216	0.1054	0.3162
F11	0.2722	0.4082	0.5443	0.6804
F12	0.3581	0.5729	0.3581	0.6445

2. Construct the weighted normalized decision Matrix.

$$WY = w_i y_{ij} \quad (3)$$

Where w_j is the weight of j^{th} feature (refer to Table I). The values of resultant matrix are shown in the Table IV.

TABLE IV. Weighted Normalized Matrix

FS	WS1	WS2	WS3	WS4
F1	0.0386	0.0435	0.0338	0.0193
F2	0.0356	0.0059	0.0534	0.0474
F3	0.0686	0.0257	0.0515	0.0086
F4	0.0487	0.0366	0.0244	0.0244
F5	0.0453	0.0113	0.0679	0.0566
F6	0.0422	0.0422	0.0361	0.0060
F7	0.0131	0.0392	0.0653	0.0783
F8	0.0166	0.0331	0.0290	0.0373
F9	0.0475	0.0285	0.0095	0.0570
F10	0.0843	0.0422	0.0105	0.0316
F11	0.0218	0.0327	0.0435	0.0544
F12	0.0322	0.0516	0.0311	0.0580

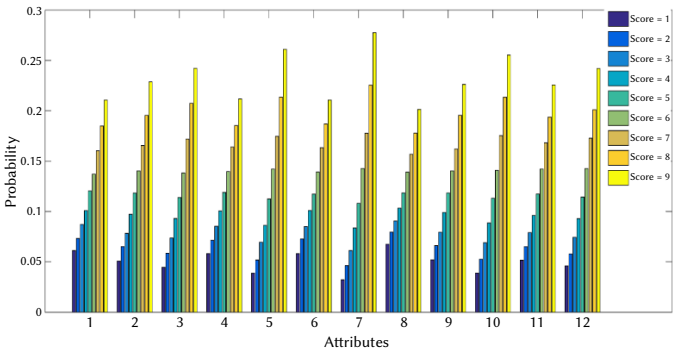


Fig. 1. Probability of selection with respective scores of each feature: SAW method.

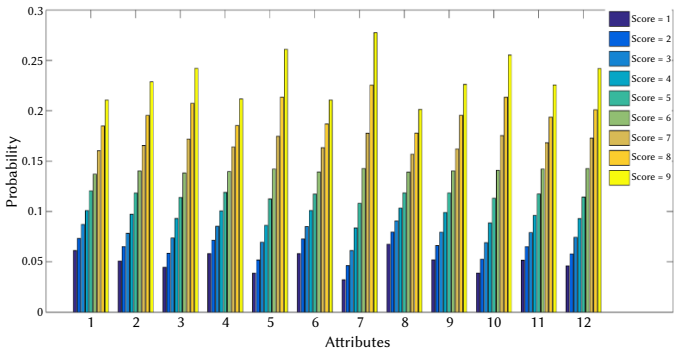


Fig. 2. Probability of selection with respective scores of each feature: TOPSIS method.

TABLE V. Probability of Selection With Respective Scores of Each Feature: SAW Method

Score	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
1	0.0193	0.0144	0.0090	0.0186	0.0059	0.0192	0.0035	0.0258	0.0137	0.0059	0.0136	0.0091
2	0.0278	0.0205	0.0149	0.0285	0.0113	0.0276	0.0077	0.0366	0.0207	0.0102	0.0211	0.0150
3	0.0408	0.0328	0.0247	0.0406	0.0202	0.0406	0.0151	0.0496	0.0334	0.0202	0.0334	0.0265
4	0.0579	0.0504	0.0424	0.0587	0.0347	0.0589	0.0292	0.0690	0.0502	0.0343	0.0497	0.0435
5	0.0820	0.0749	0.0667	0.0837	0.0582	0.0814	0.0522	0.0882	0.0769	0.0597	0.0770	0.0681
6	0.1142	0.1102	0.1040	0.1134	0.0964	0.1160	0.0914	0.1193	0.1097	0.0982	0.1100	0.1053
7	0.1589	0.1591	0.1595	0.1582	0.1547	0.1582	0.1525	0.1527	0.1591	0.1590	0.1602	0.1565
8	0.2126	0.2262	0.2355	0.2131	0.2461	0.2141	0.2494	0.2026	0.2256	0.2413	0.2216	0.2350
9	0.2864	0.3115	0.3434	0.2853	0.3726	0.2841	0.3989	0.2562	0.3108	0.3711	0.3134	0.3411

3. Fix the positive extreme and negative extreme solutions.

   Positive extreme solution: $PESj = max(w_j y_{ij})$ (4)

   Negative extreme solution: $NESj = min(w_j y_{ij})$ (5)

   where i = 1, 2, ...

   PES = (0.0435, 0.0534, 0.0686, 0.0487, 0.0679, 0.0422, 0.0783, 0.0373, 0.0570, 0.0843, 0.0544, 0.0580); and NES = (0.0193, 0.0059, 0.0086, 0.0244, 0.0113, 0.0060, 0.0131, 0.0166, 0.0095, 0.0105, 0.0218, 0.0322);

4. Determine the Euclidean distance of both virtual solutions.

   $$DPES_i = \sqrt{\sum_{j=1}^{N}(PES_j - w_j y_{ij})^2} \quad (6)$$

   $$DNES_i = \sqrt{\sum_{j=1}^{N}(w_j y_{ij} - NES_j)^2} \quad (7)$$

   In current example it takes the values (DP ESW S1 = 0.0858, DP ESW S2 = 0.1100, DP ESW S3 = 0.0987, DP ESW S4 = 0.0950) and (DNESW S1 = 0.1217, DNESW S2 = 0.0717, DNESW S3 = 0.1085, DNESW S4 = 0.1135).

5. Compute the relative closeness for the ideal solution.

   $$C_i = \frac{DNES_i}{DPES_i + DNES_i}; 0 \le C_i \le 1 \quad (8)$$

   and here the measure of relative closeness are found as: CW S1 = 0.5864, CW S1 = 0.3946, CW S1 = 0.5237, CW S1 = 0.5444.

Thus, Web Source WS1 is the best source for warehousing.

V. Proposed Work

In this article the proposed work consists of three parts. In first part statistical analysis of SAW and TOPSIS has been performed and the probability of selection with respective scores of each feature has been determined. In the second part, the probability of selection of a new web source has been determined using the probability of the respective scores of features. In the third part, improvised method (MSD method) has been proposed which is more efficient to handle the dynamic and complex behavior of the web.

A. Statistical Analysis of SAW and TOPSIS

In the statistical analysis, we have entertained all the twelve features [5], [10], [31], [32]. After execution of the Matlab implementation of both methods (SAW & TOPSIS) repeatedly around 105 times, we have selected 105 web sources, every time the best one out of 500 random sources. After that, the probability of selection of web sources respective to each score (1 to 9) for each feature has been determined using histogram methodology [19], [24]. The calculated value of the probabilities for both methods is shown in Table V and Table VI. The pictorial representation of the probability of selection with respective scores of each feature in both methods are illustrated in Fig. 1 and Fig. 2. As the figures show in both methods, as the score of the feature increases the probability of selection also increases irrespective of the weight.

Now, we are calculating the mean score and the standard deviation of score of each feature of the selected web sources for both the methods by employing the formulae:

Mean: $M_{score}(i) = \sum_{j=1}^{9} p(WS_i(j))WS_i(j)$ (9)

Std. Deviation: $DS_{score}(i) = \sqrt{\sum_{j=1}^{9} p(WS_i(j))(WS_i(j))^2}$ (10)

The values of mean score and standard deviation [19], [24] of score are shown in Table VII and Table VIII respectively. The pictorial representation of mean score and standard deviation of the score, as shown in the Fig. 3 and Fig. 4, illustrate that there is minor variation in the mean score of respective features of the selected web sources while there is a significant variation in the standard deviation of respective features of the selected web sources for both methods. As Fig. 4 shows the value of standard deviation in SAW method is greater than what we get in the TOPSIS method. It depicts that TOPSIS method is more efficient than SAW method while selecting the web sources.

TABLE VI. PROBABILITY OF SELECTION WITH RESPECTIVE SCORES OF EACH FEATURE: TOPSIS METHOD

Score	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
1	**0.0135**	0.0055	0.0014	0.0141	0.0002	0.0137	0.0000	0.0273	0.0052	0.0003	0.0054	0.0014
2	0.0257	0.0130	0.0055	0.0257	0.0017	0.0266	0.0003	0.0430	**0.0135**	0.0016	0.0140	0.0053
3	0.0451	0.0289	0.0158	0.0469	0.0077	0.0462	**0.0028**	0.0610	0.0290	0.0073	0.0298	0.0166
4	0.0731	0.0556	0.0393	0.0730	0.0245	0.0713	0.0138	0.0844	0.0560	0.0253	0.0565	**0.0384**
5	0.1002	0.0911	**0.0779**	0.1014	0.0587	0.1018	0.0425	0.1110	0.0916	0.0597	**0.0896**	0.0760
6	0.1389	0.1348	0.1269	0.1377	0.1158	**0.1370**	0.1007	0.1355	0.1332	0.1162	0.1354	0.1262
7	0.1734	**0.1819**	0.1866	0.1726	0.1891	0.1713	0.1865	**0.1599**	0.1837	0.1881	0.1815	0.1884
8	0.2027	0.2265	0.2452	**0.2038**	0.2666	0.2055	0.2849	0.1824	0.2262	0.2680	0.2277	0.2495
9	0.2274	0.2628	0.3014	0.2250	**0.3358**	0.2266	0.3686	0.1954	0.2616	**0.3353**	0.2602	0.2982

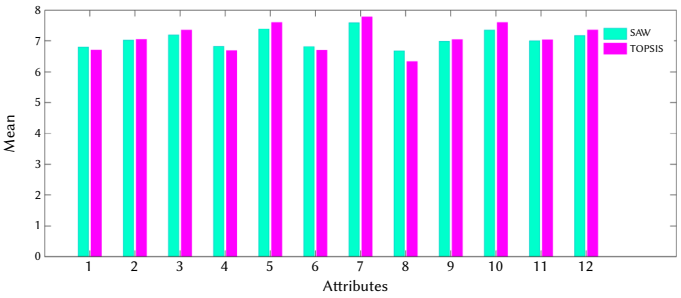


Fig. 3. Mean of the scores of features of selected resources.

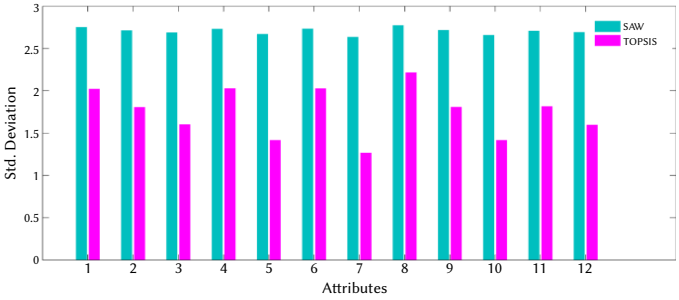


Fig. 4. Standard deviation of the scores of features of selected resources.

TABLE VII. MEAN SCORE OF SELECTED WEB SOURCES FEATURES

Feature	SAW Method	TOPSIS Method	Average
F1	6.9152	6.7088	6.8120
F2	7.1176	7.0549	7.0862
F3	7.3302	7.3479	7.3390
F4	6.9106	6.6940	6.8023
F5	7.5019	7.5914	7.5467
F6	6.9118	6.7043	6.8081
F7	7.6434	7.7822	7.7128
F8	6.6763	6.3394	6.5078
F9	7.1139	7.0502	7.0820
F10	7.4954	7.5920	7.5437
F11	7.1146	7.0426	7.0786
F12	7.3102	7.3509	7.3305

TABLE VIII. STANDARD DEVIATION OF SCORE OF SELECTED WEB SOURCES FEATURES

Feature	SAW Method	TOPSIS Method	Average
F1	2.0876	2.0190	2.0533
F2	1.9693	1.8050	1.8871
F3	1.8289	1.6014	1.1715
F4	2.0863	2.0272	2.0567
F5	1.7157	1.4148	1.5653
F6	2.0839	2.0265	2.0552
F7	1.5990	1.2630	1.4310
F8	2.2028	2.2141	2.2084
F9	1.9672	1.8063	1.8836
F10	1.9694	1.8136	1.8915
F11	1.8430	1.5952	1.7191
F12	1.3420	1.7920	1.8121

B. Probability of Selection of a New Web-Source

As there is a large number of web sources available on web as well as an exponential growth of web sources, an efficient methodology to select the web sources for web warehousing is required. One way is to calculate the probability of selection of each new web source to evaluate the relevance of specific web source. In order to calculate the probability of selection of a web source by any comparative method (like SAW and TOPSIS methods), the probability of respective score and weight of each feature are required and the formula to calculate the probability is as follows [1]:

$$p_s(WS) = \sum_{i=1}^{n} W_i p_i(WS_i(j)) \quad (11)$$

Where

$$\sum_{i=1}^{n} W_i = 1 \quad (12)$$

$$\sum_{i=1}^{m} p_i(WS_i(j)) = 1 \quad (13)$$

Here, n is the number of features of the web source (WS) and (1, 2... m) are the scores of the features. $p_i\,(WS_i\,(j))$ is the probability of selection of i^{th} feature having score value j. In this article n = 12 and m = 9.

A higher value of the probability shows the specific web source is more relevant, so it is recommendable to be used as data source for warehouse. Here the individual web source relevancy can be assessed by probability without any comparison.

For example, if a new web source (WS) has the score of all twelve features as {1, 7, 5, 8, 9, 6, 3, 7, 2, 9, 5, 4} and the weight of each feature is as mentioned in Table I, then its probability of getting selected (in TOPSIS method) is as follows:

Here, WS(1) = 1, WS(2) = 7, WS(3) = 5, WS(4) = 8, WS(5) = 9, WS(6) = 6, WS(7) = 3, WS(8) = 7, WS(9) = 2, WS(10) = 9, WS(11) = 5 and W S(12) = 4. The probability of selection of the respective feature with respect to the score in TOPSIS method is given in TABLE VI (highlighted in bold) which are:

p1 (WS (1)) = 0.0135, p2 (WS (2)) = 0.1819, p3 (WS (3)) = 0.0779, p4 (WS (4)) = 0.2038, p5 (WS (5)) = 0.3358, p6 (WS (6)) = 0.1370,

p7 (WS (7)) = 0.0028, p8 (WS (8)) = 0.1599, p9 (WS (9)) = 0.0135, p10 (WS (10)) = 0.3353, p11 (WS (11)) = 0.0896, p12 (WS (12)) = 0.0384.

Now, by applying the formula (11), the probability of selection of the web source (WS) is **0.1351**.

Similarly, the probability of selection in the TOPSIS method can also be calculated.

C. The MSD Method

The proposed MSD method is an enhancement of the already defined methods to handle the complex and dynamic nature of the web. It is based on probabilistic analysis of MCDM methods. The proposed method consists of two parts: (i) fixing the standard score *Sscore* of each feature for selection, and (ii) checking the suitability of the coming web sources. The steps of the method have been elaborated in the following algorithm.

Algorithm:

(i) Fixing the standard score:

1. Determine the mean score (Mscore(i)) and standard deviation of score (Sscore(i)) of i^{th} feature from selected web sources, where i = {1, 2, ..., m}.

2. Determine the standard score (Sscore(i)) for i^{th} feature using formula:

$$\text{Sscore}(i) = \text{Mscore}(i) - \text{SDscore}(i) \quad (14)$$

3. Set the Sscore(i) as the selection parameter for i^{th} feature.

(ii) Check the suitability:

1. Set the threshold value Th: (where Th ≤ m).

2. For each feature of a web source (WS) calculate:

if $\text{WS}(i) \geq \text{Sscore}(i)$

$\text{Suitability}(i) = 1;$

otherwise,

$\text{Suitability}(i) = 0;$

3. If $\sum_{i=1}^{m} \text{Suitability}\,(i) \geq \text{Th}$, then the web source is suitable to select, otherwise rejected.

Here the standard score of each feature is derived from the mean score and the standard deviation of the scores, using the probability values of the respective score of each features employing SAW and TOPSIS methods. Now the standard score of the respective feature is used as parameter to check the suitability of the web source with respect to that feature. If a new web source has the number of features (whose score is greater than the respective standard score) more than the threshold value, then the web source is selected otherwise rejected. Here the threshold value is only to check the number of features whose value is more than the standard score.

VI. Experimental Setup and Result Analysis

For the implementation of all the three methods (SAW, TOPSIS and MSD), we have used Matlab12a, Windows 8 (64 bit Operating System), Intel CITM) i3-4005U CPU @ 1.70 GHz. In order to determine the standard score (*Sscore*), we calculate the average of the mean scores and average of the standard deviation of the scores of each feature of the selected web sources employing SAW and TOPSIS methods, and results are shown in the TABLE VII and TABLE VIII respectively. Using the aforementioned algorithm of MSD method, the worthy web sources have been selected as shown in TABLE IX, here NFS stands for 'None Found Suitable'. For implementation and analysis, we have taken fourteen data-sets and each data-set consists of twenty randomly generated web sources with some score value for each feature. All these data-sets are given in the Appendix I.

The results and comparative analysis of all the three methods as shown in Table IX, show the effectiveness of the proposed MSD method while dealing with the complex and dynamic nature of web. The MSD method also shows improvisation during selection of web sources in comparison with SAW and TOPSIS methods in the following way:

- MSD method assures the suitability of web sources individually, whereas SAW and TOPSIS methods find the best one, on relative comparison basis.
- SAW and TOPSIS methods will select available single web source by default without any evaluation. However, the MSD method either selects or rejects depending on whether the threshold value is met or not.
- When the data-set consists of worthy web sources, the MSD method either agrees or disagrees with the SAW and TOPSIS methods due to the involvement of weight of features, as shown in Table IX for Data-sets 1, 2, 3, 4, 5, 6 and for Data-sets 7, 8, 11 respectively.
- SAW and TOPSIS methods usually select one web source (the best one) while the MSD method may select more than one suitable web sources in a single execution as shown in TABLE IX for Data-sets 3, 4, 5, 6, 10, 13 and 14. So it is effective to handle the dynamics of web.
- If all the new web sources are bad, both SAW and TOPSIS methods will select the best one from all the bad, but the MSD method will reject all of them as shown in in TABLE IX for Data-sets 9 and 12.

TABLE IX. Selected Web Sources Features

Data Set	SAW Method	TOPSIS Method	MSD Method
1	7	7	7
2	9	9	9
3	4	4	4, 8
4	6	6	6, 15
5	16	15	15, 16
6	5	5	5, 10, 20
7	4	3	14
8	17	17	15
9	16	16	NFS
10	10	8	6, 9, 12
11	11	16	13
12	10	8	NFS
13	15	17	11, 17, 19
14	9	10	9, 17

VII. Conclusion

In this article, statistical analysis has been performed on SAW and TOPSIS methods to study the behavior of both methods and also propose an efficient method based on this statistical study. In statistical analysis, the probability of the scores of each feature in both methods enforces that, as the value of the probabilities increases, the chance of selection increases. Using these probability values of the score of the features, the probability of selection of a new web source can be calculated by eq. (11). Furthermore, the mean of the score of a feature in both methods is almost the same but there is significant variation in standard deviation of the scores of the respective features. It shows the TOPSIS method is more effective than SAW to select the web sources. SAW and TOPSIS methods always yield the best one among all the available web sources on comparative basis without checking the quality of web sources, while the MSD method deals individually with each web source and assures its quality while selecting. So, if there is single web source, it is selected by default by both the methods because there is no other source to compare, but not in the MSD method.

Once after the computation of Mean Score and Standard Deviation Score, there is no further comparisons of feature score as in SAW and TOPSIS methods for selection of web-sources. So, the proposed method is more efficient in selection of web-sources where the data is updated frequently.

The proposed MSD method is only based on standard scores of each feature so gets rid from manually/randomly fixing weight of features as well as comparison among the web sources. Thus checking the quality of web sources individually in the MSD method makes it more efficient to deal with the dynamic and complex nature of web. Moreover, the computation cost in both methods is always higher than the MSD method due to the involvement of comparison operations to select the best one but it is linear for the proposed method and it will based on the number of evaluation features of the web-sources. If the number of the features are n then the computational cost is $O(n)$.

VIII. Future Work

A lot of enhancement is still required to design the effective web warehouse. Further research is needed to analyze the sensitiveness of the selected web sources when various critical factors are changed simultaneously. The MCDM approaches for selecting suitable web-data sources have a number of methods to evaluate the suitability. The proposed work is based on the aggregated study of SAW and TOPSIS methods. Various other MCDM methods like TOPSIS-COMET, COCOSO, and MABRAC [34] may be statistically investigated and incorporated with the proposed MSD method to improve the suitability of the selection of web-data sources for the data-warehouse storage. Moreover, the contents over the web sources change randomly and dynamically. So our focus in the future is to identify the updated relevant data over the selected web sources with minimum latency in order to update the web warehouse.

Appendix

Data Set: 01

Features	1	2	3	4	5	6	7	8	9	10	11	12
WS1	7	6	8	7	8	1	4	1	1	9	7	6
WS2	5	3	8	9	9	8	2	3	4	9	5	6
WS3	1	3	3	4	5	4	5	3	2	4	7	5
WS4	3	2	9	4	5	6	1	9	8	4	8	1
WS5	4	4	6	2	9	9	8	9	7	7	3	5
WS6	2	9	1	4	6	3	3	1	7	7	9	1
WS7	1	8	4	5	7	9	7	6	9	8	6	8
WS8	5	4	5	5	6	8	8	1	8	6	3	6
WS9	4	7	6	2	5	8	7	4	5	1	3	7
WS10	8	3	5	2	2	6	6	5	1	6	9	6
WS11	5	9	8	4	6	5	3	8	1	1	4	6
WS12	8	6	2	8	2	4	5	6	5	7	2	9
WS13	5	2	9	2	2	8	9	3	9	6	9	2
WS14	1	1	8	5	8	6	3	9	8	6	6	5
WS15	1	6	4	2	2	8	2	7	9	3	9	8
WS16	2	3	6	8	6	9	3	6	9	7	2	5
WS17	4	1	3	7	4	8	7	9	4	7	8	7
WS18	7	2	1	6	6	8	3	4	3	4	5	2
WS19	2	1	5	4	7	5	6	4	8	1	6	7
WS20	3	7	8	5	3	8	2	6	2	8	1	9

Data Set: 02

Features	1	2	3	4	5	6	7	8	9	10	11	12
WS1	7	6	5	7	8	4	3	2	7	6	6	1
WS2	1	6	3	5	1	3	7	1	8	4	9	1
WS3	4	9	7	6	7	6	3	6	5	6	6	2
WS4	5	2	9	1	6	4	9	1	1	4	5	5
WS5	7	8	6	3	7	2	5	7	1	2	6	1
WS6	2	4	3	1	9	9	4	2	2	1	9	1
WS7	5	9	9	8	2	5	6	3	8	2	5	1
WS8	1	8	1	1	1	5	6	9	4	4	4	6
WS9	8	7	6	5	8	3	8	2	8	8	9	1
WS10	4	2	9	2	5	2	8	8	2	4	9	6
WS11	3	1	5	6	1	8	2	1	7	6	9	6
WS12	1	2	9	6	1	3	1	3	2	4	7	1
WS13	1	5	3	7	2	5	3	9	8	8	6	3
WS14	9	1	5	3	2	5	6	3	5	7	7	4
WS15	2	8	9	9	9	9	4	2	9	2	4	2
WS16	9	9	2	8	3	3	9	8	4	4	8	1
WS17	4	4	8	1	6	3	1	6	9	6	2	5
WS18	9	5	2	3	8	3	9	8	4	4	8	1
WS19	4	1	4	9	3	6	7	9	1	1	9	7
WS20	1	3	8	1	7	4	6	1	9	6	9	8

Data Set: 03

Features	1	2	3	4	5	6	7	8	9	10	11	12
WS1	4	7	4	2	6	4	9	2	5	7	3	1
WS2	2	7	3	2	6	2	8	5	6	7	4	8
WS3	7	7	3	6	8	4	9	9	1	5	5	3
WS4	2	8	3	7	9	8	7	4	6	9	9	8
WS5	9	6	1	8	1	4	3	6	7	9	1	2
WS6	1	5	9	1	8	7	7	1	6	9	7	7
WS7	1	3	5	2	6	3	7	2	5	5	9	8
WS8	8	9	6	5	2	6	7	2	8	9	8	7
WS9	8	3	3	4	2	8	7	1	9	4	2	3
WS10	6	5	9	4	7	2	3	3	3	6	9	4
WS11	7	7	9	1	6	5	5	3	8	6	4	4
WS12	5	3	9	7	6	2	7	4	3	8	8	3
WS13	6	8	6	4	9	2	6	2	1	3	8	4
WS14	2	5	7	7	6	1	9	9	1	6	4	1
WS15	5	9	4	5	1	1	2	5	3	6	6	5
WS16	1	4	5	1	2	6	9	4	3	6	8	5
WS17	1	9	3	3	1	4	7	3	1	3	1	1
WS18	4	9	7	9	2	6	3	4	2	3	1	6
WS19	6	6	8	9	7	8	6	4	4	2	4	1
WS20	9	1	1	4	6	7	5	9	7	1	2	2

Data Set: 04

Features	1	2	3	4	5	6	7	8	9	10	11	12
WS1	3	9	7	1	9	5	9	3	8	8	5	6
WS2	7	1	6	6	2	1	3	4	4	5	5	9
WS3	9	3	2	3	5	3	9	3	3	1	7	1
WS4	8	5	1	5	9	2	5	2	4	9	9	2
WS5	6	3	4	6	6	1	6	6	4	4	9	4
WS6	6	5	5	1	6	9	9	8	8	7	9	7
WS7	9	8	2	1	5	1	7	2	1	2	6	9
WS8	4	5	8	3	6	8	3	9	7	8	9	1
WS9	3	9	6	5	9	5	1	7	8	9	2	2
WS10	6	2	1	6	4	2	1	5	3	8	4	4
WS11	3	4	9	3	3	5	8	9	6	7	1	7
WS12	6	3	5	9	4	6	4	6	3	4	5	5
WS13	1	6	1	8	3	1	4	2	6	7	7	7
WS14	9	1	1	5	5	9	5	8	1	6	2	2
WS15	2	1	9	9	6	6	9	8	9	6	4	7
WS16	6	3	8	6	5	5	4	3	6	4	1	6
WS17	4	1	6	3	4	8	6	3	4	7	3	1
WS18	7	2	1	8	8	3	1	8	1	2	5	9
WS19	1	4	3	5	3	8	4	5	8	1	5	9
WS20	1	1	5	3	4	3	2	1	6	6	7	2

Data Set: 05

Features	1	2	3	4	5	6	7	8	9	10	11	12
WS1	4	4	1	3	3	2	6	7	7	3	1	1
WS2	9	2	9	8	3	3	2	1	9	7	3	7
WS3	4	2	6	8	2	3	5	9	9	3	8	7
WS4	7	1	6	3	2	8	5	8	4	7	1	2
WS5	1	8	5	9	4	2	8	6	2	8	6	8
WS6	2	8	5	2	9	7	6	8	4	1	9	2
WS7	1	9	5	8	8	8	2	2	8	9	9	9
WS8	2	4	7	8	7	7	2	2	6	2	9	3
WS9	2	5	6	5	6	8	6	1	6	2	1	7
WS10	5	6	2	2	3	7	2	7	9	6	4	6
WS11	2	7	4	1	3	8	5	3	3	3	3	2
WS12	5	9	6	2	1	4	7	7	3	9	4	3
WS13	7	6	5	7	8	9	6	5	3	6	9	2
WS14	8	9	2	5	7	5	3	3	9	9	3	9
WS15	7	9	8	5	7	8	7	9	8	2	6	5
WS16	9	9	8	9	3	9	4	8	9	1	8	9
WS17	2	3	3	2	8	8	1	5	2	9	7	2
WS18	2	9	2	2	5	4	7	2	8	7	9	6
WS19	9	2	8	4	2	6	7	1	4	1	1	8
WS20	9	8	3	2	8	5	1	7	6	1	8	4

Data Set: 06

Features	1	2	3	4	5	6	7	8	9	10	11	12
WS1	8	8	1	8	8	2	3	2	4	7	6	2
WS2	3	4	8	9	6	7	3	3	8	8	2	6
WS3	4	6	5	2	9	5	5	8	9	8	2	7
WS4	4	9	9	5	6	6	5	5	4	9	1	3
WS5	7	8	8	7	8	4	6	5	7	8	7	8
WS6	9	8	2	1	2	4	6	8	9	5	6	2
WS7	2	9	4	1	8	7	2	9	9	5	8	3
WS8	7	2	1	4	9	9	4	1	1	1	1	6
WS9	5	7	4	2	3	8	4	4	1	5	3	8
WS10	7	7	4	9	9	6	1	5	6	6	1	8
WS11	8	8	3	9	7	9	3	7	1	1	6	6
WS12	3	7	9	9	7	6	9	9	2	4	3	1
WS13	9	3	5	9	7	7	9	7	1	3	7	9
WS14	1	1	7	7	8	9	8	3	8	6	2	8
WS15	6	2	6	7	5	3	2	9	2	4	5	4
WS16	9	3	8	4	7	1	5	9	9	4	1	8
WS17	2	8	3	1	5	8	1	5	7	1	1	7
WS18	4	9	7	4	9	4	7	7	8	3	5	3
WS19	3	5	4	5	2	7	8	6	4	3	7	2
WS20	6	9	1	5	8	9	9	9	3	7	6	8

Data Set: 07

Features	1	2	3	4	5	6	7	8	9	10	11	12
WS1	9	1	1	6	6	7	3	4	5	1	3	7
WS2	4	3	1	2	3	2	1	1	3	8	9	3
WS3	7	7	7	6	8	1	8	9	4	6	5	1
WS4	2	8	4	7	8	7	4	6	3	5	7	9
WS5	1	5	6	5	5	5	6	3	3	5	7	9
WS6	6	9	3	2	4	5	6	1	2	5	8	8
WS7	7	4	7	4	9	3	1	3	2	6	3	9
WS8	3	3	4	6	1	3	6	9	5	6	3	5
WS9	6	5	7	3	4	9	3	2	1	2	7	2
WS10	3	2	9	8	5	5	3	5	9	8	2	5
WS11	8	2	1	2	6	7	5	9	2	2	8	7
WS12	4	3	5	3	1	2	6	2	2	4	2	7
WS13	7	7	1	9	2	1	8	5	7	3	9	8
WS14	6	6	7	6	6	2	6	6	7	3	7	7
WS15	9	9	9	7	2	6	1	6	4	1	2	6
WS16	2	7	1	8	2	6	8	6	1	9	8	3
WS17	6	6	5	1	1	9	3	3	1	5	9	4
WS18	1	5	5	4	2	1	8	4	4	5	6	8
WS19	5	3	7	7	5	2	7	8	8	5	4	4
WS20	8	7	3	2	8	4	6	9	5	4	4	1

Data Set: 08

Features	1	2	3	4	5	6	7	8	9	10	11	12
WS1	8	2	2	6	1	1	4	7	9	9	8	3
WS2	2	3	3	1	9	3	6	1	1	4	3	5
WS3	5	6	5	4	6	5	9	1	9	5	9	9
WS4	3	4	3	9	5	3	8	8	9	8	9	6
WS5	6	6	5	9	9	9	8	4	9	2	9	1
WS6	5	3	8	3	7	2	1	6	7	9	1	4
WS7	3	9	6	3	5	3	8	7	7	3	6	8
WS8	5	7	1	8	4	5	4	9	1	7	4	4
WS9	1	7	2	4	4	6	3	1	2	5	1	7
WS10	6	4	4	8	2	8	1	2	8	6	9	7
WS11	8	6	9	5	5	6	7	2	8	7	7	3
WS12	7	5	6	5	9	8	8	8	1	3	1	9
WS13	5	9	2	4	2	6	1	2	7	5	7	2
WS14	7	1	4	4	7	5	3	5	3	2	3	8
WS15	8	5	3	5	8	8	9	7	3	9	9	4
WS16	5	7	4	7	9	4	1	9	8	6	2	7
WS17	9	9	8	7	6	1	9	6	4	8	6	4
WS18	6	6	5	4	6	9	4	3	6	5	9	5
WS19	8	6	3	5	7	7	8	5	1	8	4	8
WS20	5	6	6	4	3	6	3	5	7	5	8	2

Data Set: 09

Features	1	2	3	4	5	6	7	8	9	10	11	12
WS1	7	7	4	9	8	5	4	3	4	8	2	6
WS2	3	5	6	1	5	6	4	7	1	5	9	6
WS3	1	1	9	4	4	3	8	3	8	7	7	5
WS4	8	4	9	1	7	2	2	2	6	7	9	2
WS5	7	4	3	3	5	7	6	5	5	5	7	5
WS6	9	6	2	7	3	4	9	4	6	8	2	7
WS7	9	2	4	7	1	5	6	9	3	5	3	4
WS8	5	1	5	3	6	7	1	3	7	9	1	7
WS9	9	5	1	3	7	6	7	6	4	8	1	7
WS10	5	7	8	1	6	5	3	6	8	2	3	9
WS11	9	6	9	1	5	7	7	6	4	3	2	3
WS12	1	9	5	7	5	7	4	6	8	2	7	4
WS13	2	3	2	3	3	3	5	3	8	5	9	4
WS14	7	2	7	5	4	8	2	2	1	1	4	1
WS15	4	3	7	9	7	6	4	1	9	9	8	6
WS16	4	8	5	7	9	6	4	1	9	9	8	6
WS17	6	7	2	5	3	1	1	8	4	7	5	4
WS18	1	8	8	6	8	8	1	4	3	8	7	4
WS19	4	7	8	6	2	2	6	8	6	5	7	5
WS20	1	6	2	3	1	8	9	9	8	6	4	3

Data Set: 10

Features	1	2	3	4	5	6	7	8	9	10	11	12
WS1	8	8	9	3	2	8	8	2	6	1	7	1
WS2	3	3	5	2	8	8	7	3	4	1	8	6
WS3	2	7	4	8	5	1	6	2	4	6	5	8
WS4	7	4	6	7	2	8	4	3	1	2	1	3
WS5	9	3	8	2	1	4	2	2	5	5	6	6
WS6	9	8	8	8	3	6	4	5	6	6	9	2
WS7	2	5	7	4	1	1	8	6	9	3	5	8
WS8	2	4	9	3	9	8	2	4	7	9	5	8
WS9	5	6	2	8	8	5	1	5	8	7	9	7
WS10	9	2	9	9	3	3	8	9	5	4	7	9
WS11	2	7	9	3	2	5	4	2	6	3	5	5
WS12	7	3	8	9	7	8	2	8	6	9	1	7
WS13	5	3	3	1	3	7	8	3	3	4	5	7
WS14	8	5	3	6	1	8	1	6	2	1	7	4
WS15	6	8	5	8	4	9	3	5	3	2	1	9
WS16	4	4	9	2	4	6	8	1	3	6	1	8
WS17	5	6	5	9	5	4	3	5	3	5	6	3
WS18	9	2	7	6	7	7	7	6	2	3	4	6
WS19	9	6	2	1	9	3	7	4	8	4	4	5
WS20	6	3	6	1	7	4	3	3	5	2	5	9

Data Set: 11

Features	1	2	3	4	5	6	7	8	9	10	11	12
WS1	3	3	7	5	9	7	6	9	4	1	3	6
WS2	2	2	8	9	3	8	5	8	4	3	2	3
WS3	6	8	6	4	4	7	3	8	8	3	2	1
WS4	9	7	5	6	6	2	8	9	1	2	6	1
WS5	7	4	2	4	3	3	7	9	9	7	4	9
WS6	2	6	2	2	9	3	7	1	5	5	1	1
WS7	2	4	8	8	2	9	8	9	3	2	3	6
WS8	3	5	4	9	7	8	3	8	5	9	6	4
WS9	1	2	4	5	1	7	5	1	7	1	5	6
WS10	7	6	5	3	3	8	9	3	2	6	8	8
WS11	5	5	5	9	8	4	9	9	6	1	8	8
WS12	2	4	4	2	2	8	2	5	6	4	9	7
WS13	6	8	8	7	3	7	9	5	2	3	9	8
WS14	2	1	4	5	7	6	6	1	9	1	1	4
WS15	2	7	2	8	3	1	1	5	2	3	9	8
WS16	6	1	6	1	6	9	5	4	8	9	9	7
WS17	3	6	9	5	7	7	3	4	3	7	2	1
WS18	5	4	2	4	7	3	5	1	5	5	6	1
WS19	2	4	8	1	6	1	3	7	9	5	1	8
WS20	1	7	4	9	4	5	3	9	6	5	8	6

Data Set: 12

Features	1	2	3	4	5	6	7	8	9	10	11	12
WS1	2	8	3	1	3	9	6	8	1	4	5	8
WS2	8	2	4	1	7	5	7	2	9	8	8	6
WS3	1	1	5	5	3	6	8	5	6	5	6	9
WS4	4	6	1	5	9	1	4	7	2	6	2	6
WS5	3	1	4	5	3	6	7	3	4	3	6	4
WS6	9	8	3	2	2	9	5	5	8	1	2	6
WS7	8	9	5	4	5	5	4	4	5	6	4	5
WS8	8	3	8	8	8	4	4	8	5	9	5	1
WS9	9	3	4	7	4	1	7	7	4	1	8	6
WS10	8	1	5	4	9	6	3	9	2	2	6	9
WS11	8	5	1	9	4	9	9	9	3	1	8	5
WS12	3	9	9	1	3	7	9	6	3	8	2	6
WS13	7	7	2	2	3	4	3	1	5	5	5	6
WS14	6	3	4	9	9	6	4	4	6	5	9	3
WS15	3	6	6	1	3	6	9	2	4	9	1	7
WS16	4	8	7	4	4	7	8	9	9	4	6	2
WS17	9	2	2	3	3	6	8	5	3	5	1	7
WS18	1	6	5	1	3	5	6	8	6	7	4	2
WS19	7	8	9	2	4	9	8	6	3	1	7	2
WS20	2	7	7	4	9	6	5	5	3	5	1	7

Data Set: 13

Features	1	2	3	4	5	6	7	8	9	10	11	12
WS1	8	2	2	6	1	1	4	7	9	9	8	3
WS2	2	3	3	1	9	3	6	1	1	4	3	5
WS3	5	6	5	4	6	5	9	1	9	5	9	9
WS4	3	4	3	9	5	3	8	8	9	8	9	6
WS5	6	6	5	9	9	9	8	4	9	2	9	1
WS6	5	3	8	3	7	2	1	6	7	9	1	4
WS7	3	9	6	3	5	3	8	7	7	3	6	8
WS8	5	7	1	8	4	5	4	9	1	7	4	4
WS9	1	7	2	4	4	6	3	1	2	5	1	7
WS10	6	4	4	8	2	8	1	2	8	6	9	7
WS11	8	6	9	5	5	6	7	2	8	7	7	3
WS12	7	5	6	5	9	8	8	8	1	3	1	9
WS13	5	9	2	4	2	6	1	2	7	5	7	2
WS14	7	1	4	4	7	5	3	5	3	2	3	8
WS15	8	5	3	5	8	8	9	7	3	9	9	4
WS16	5	7	4	7	9	4	1	9	8	6	2	7
WS17	9	9	8	7	6	1	9	6	4	8	6	4
WS18	6	6	5	4	6	9	4	3	6	9	5	9
WS19	8	6	3	5	7	7	8	5	1	8	4	8
WS20	5	6	6	4	3	6	3	5	7	5	8	2

Data Set: 14

Features	1	2	3	4	5	6	7	8	9	10	11	12
WS1	3	9	2	5	5	6	2	9	8	1	6	1
WS2	6	7	1	9	8	1	8	4	9	5	6	8
WS3	2	3	2	2	6	4	1	3	4	2	3	5
WS4	9	7	1	9	3	4	2	5	1	5	8	7
WS5	7	3	3	4	5	7	2	2	1	6	4	3
WS6	8	6	1	5	9	4	9	4	8	1	9	1
WS7	5	3	7	4	8	5	5	8	9	8	2	2
WS8	9	5	9	2	8	6	1	3	7	1	8	6
WS9	8	7	2	8	9	4	8	7	9	6	8	1
WS10	2	7	6	2	7	8	9	6	5	5	5	9
WS11	9	4	4	8	1	5	5	8	5	6	7	9
WS12	7	2	2	4	9	8	4	5	6	8	8	4
WS13	3	1	2	4	5	3	6	4	9	2	3	9
WS14	8	1	2	5	5	8	3	6	9	2	7	6
WS15	9	9	9	4	1	2	7	6	1	2	9	8
WS16	7	6	5	1	8	2	9	5	6	8	9	4
WS17	6	6	8	6	7	1	2	7	7	7	8	1
WS18	2	1	2	3	5	2	6	4	2	9	5	5
WS19	9	2	3	6	7	8	6	2	1	2	1	3
WS20	8	1	8	4	7	9	6	4	5	4	1	9

References

[1] S. I. Amari, H. Nagaoka, and D. Harda, "Methods of information geometry. Translation of mathematical monographs," *Oxford University Press*, 2000. ISBN: 978-1-4704-4605-**5** https://bookstore.ams.org/mmono-191

[2] O. Boussaid, J. Darmont, F. Bentayeb, and S. Loudcher, "Warehousing complex data with from the web," *International Journal of Web Engineering and Technology*, vol. 4, no. 4, pp. 408-433, 2008. doi: 10.1504/IJWET.2008.019942.

[3] O. Boussaid, A. Tanasescu, F. Bentayeb, and J. Darmont, "Integration and dimensional modeling approaches for complex data warehousing," *Journal of Global Optimization*, vol. 37, pp. 571-591, 2007. https://doi.org/10.1007/s10898-006-9064-6.

[4] T. Y. Chen, "Comparative analysis of SAW and TOPSIS based on interval valued fuzzy sets: Discussion on score functions and weights constraints," *Expert Systems with Applications*, vol. 39, pp. 1848–1861, 2012. doi: 10.1016/j.eswa.2011.08.065.

[5] J. L. Devore, "Probability and statistics for engineering and the sciences," *Cengage Learning*, 2012. ISBN: 978-8131518397.

[6] A. Doan, A. Halevy, and Z. Ives, "Principles of data integration," *Elsevier*, 2012. ISBN: 978-0-12-416044-6.

[7] X. L. Dong, B. Saha, and D. Srivastava, "Less is more: Selecting sources wisely for integration," *Proceeding of the VLDB Endowment*, vol. 6, pp. 37-48, 2012. doi: https://doi.org/10.14778/2535568.2448938 .

[8] H. Fan, "Investigating a heterogeneous data integration approach for data warehousing," *PhD Thesis*, School of Computer Science & Information Systems, Birkbeck College, University of London, 2005. Accessed: Jan. 15, 2023. [Online]. Available: https://www.dcs.bbk.ac.uk/site/assets/files/1025/haofan.pdf

[9] R. D. Hackathorn, "Web framing for the data warehouse," *Morgan Kaufmann Publishers Inc.*, San Francisco, CA, USA, 1999. ISBN: 978-1558605039.

[10] J. L. Johnson, "Probability and statistics for computer science," *Wiley*, 2008. ISBN: 978-0470383421.

[11] D. Le, J. Rahayu, and E. Pardede, "Dynamic approach for integrating web data warehouses," *Computational Science and Its Applications*, ICCSA-2006, Springer, 2006. ISBN: 0302-9743.

[12] A. Marshall, "Principles of Economics," Prometheus Books, 1890. Accessed: Jan. 15, 2023. [Online]. Available: https://eet.pixel-online.org/files/etranslation/original/Marshall,%20Principles%20of%20Economics.pdf

[13] B. H. Massam, "Massam. Multi-criteria decision making (mcdm) techniques in planning," *Progress in planning*, vol. 30, no. 1, pp. 1–84, 1988.

[14] A. Mehedintu, I. Buligiu, and C. Pirvu, "Web-enabled data warehouse and data webhouse," *Revista Informatica Economica nr*, vol. 1, no. 45, pp. 96-102, 2008. https://core.ac.uk/download/pdf/6612753.pdf

[15] A. Memariani, A. Amini, and A. Alinezhad, "Sensitivity analysis of simple additive weighting method (saw): the results of change in the weight of one attribute on the final ranking of alternatives," *Journal of Industrial Engineering*, vol. 4, pp. 13-18, 2009.

[16] F. Naumann, "Data fusion and data quality," 1998.

[17] J. M. Perez, R. Berlanga, M. J. Aramburu, and T. B. Pedersen, "Integrating data warehouses with web data: A survey," *IEEE Transactions on Knowledge and Engineering*, vol. 20, no. 7, pp. 940-955, 2008. doi: 10.1109TKDE.2007.190746.

[18] S. Rizzi, A. Abello, J. Lechtenborger, and J. Trujillo, "Research in data warehouse modeling and design: dead or alive?" *Proceedings of the 9th ACM international workshop on Data warehousing and OLAP* (DOLAP '06), pp. 3-10. *IEEE Computer Society*, 2006. doi: 10.1145/1183512.1183515.

[19] S. Ross, "Introduction to Probability Models," *Academic Press/Elsevier*, 2012. ISBN: 978-0-12-407948-9.

[20] R. Simanaviciene and L. Ustinovichius, "Quality-driven integration of heterogeneous information systems," *Informatik-Berichte*, vol. 117, pp. 1-21, 1999. https://www.vldb.org/conf/1999/P43.pdf

[21] R. Simanaviciene and L. Ustinovichius, "Sensitivity analysis for multiple criteria decision making methods: Topsis and saw," *Procedia Social and Behavioral Sciences*, vol. 2, pp. 7743-7744, 2010.

[22] X. Tan, D. C. Yen, and X. Fang, "Web warehousing: Web technology meets data warehousing," *Technology in Society*, vol. 25, no. 131-148, 2003.

[23] E. Triantaphyllou, B. Shu, S. Sanchez, and T. Ray, "Multi-criteria decision making: an operations research approach," *Encyclopedia of Electrical and Electronics Engineering*, vol. 15, pp. 175-186, 1998.

[24] K. S. Trivedi, "Probability and Statistics with Reliability, Queuing and Computer Science Applications," *Wiley*, 2013.

[25] Y. Zhu and A. Buchmann, "Evaluating and selecting web sources as external information resources of a data warehouse," *Proceedings of the 3rd International Conference on Web Information Systems Engineering (WISE202)*, pp. 140-160. IEEE Computer Society, 2002. doi: 10.1109/WISE.2002.1181652.

[26] G. Xu, "The Construction Site Management of Concrete Prefabricated Building by ISM-ANP Network Structure Model and BIM Under Big Data Text Mining," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, no. 4, pp. 138-145, 2020. doi: 10.9781/ijimai.2020.11.013

[27] S. Kumar, V. K. Solanki, S. K. Choudhary, A. Selamat and R. G. Crespo, "Comparative Study on Ant Colony Optimization (ACO) and K-Means Clustering Approaches for Jobs Scheduling and Energy Optimization Model in Internet of Things (IoT)," International Journal of Interactive Multimedia and Artificial Intelligence (Special Issues on Soft Computing), vol. 6, no. 1, pp. 107-116, 2020. doi: 10.9781/ijimai.2020.01.003.

[28] S. Zhang, L. Genga, H. Yan, H. Nie, X. Lu and U. Kaymak, "Towards Multi-perspective Conference Checking with Fuzzy Sets," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, no. 5, pp. 134-141, 2021. doi: 10.9781/ijimai.2021.02.013.

[29] Y. Wu, L. Zhang, G. Ding, T. Xue and F. Zhang, "Modeling of Performance Creative Evaluation Driven by Multimodal Affective Data," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, no. 7, pp. 90-100, 2021. doi: 10.9781/ijimai.2021.08.005.

[30] D. Burgos, "Ritual and Data Analytics: A Mixed-Methods Model to Process Personal Belief," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 7, no. 1, pp. 52-61, 2021. doi: 10.9781/ijimai.2021.07.002.

[31] S. K. Choudhary, K. Singh and V. K. Solanki, "Spiking Activity of LIF Neuron in Distributed Delay Framework," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 3, no. 7, pp. 70-76, 2016. doi: 10.9781/ijimai.2016.3710 .

[32] I. Lopez-Plata, C. Exposito-Izquierdo, E. Lalla-Ruiz, B. Melian-Batista, J. Marcos-Vega, "A Greedy Randomized Adaptive Search With Probabilistic Learning for solving the Uncapacitated Plant Cycle Location Problem," International Journal of Interactive Multimedia and Artificial Intelligence, 2022, (In Press), doi: 10.9781/ijimai.2022.04.003.

[33] N.S. Houari & N. Taghezout, "An Efficient Tool for the Experts' Recommendation Based on PROMETHEE II and Negotiation: Application to the Industrial Maintenance," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, no. 6, pp.67-77, 2021, doi: 10.9781/ijimai.2021.01.002.

[34] A. Baczkiewicz, B. Kizielewicz, A. Shekhovtsov, J. Watrobski & W. Salabun, "Methodical Aspects of MCDM Based E-Commerce Recommender System," Journal of Theoretical and Applied Electronic Commerce Research, vol. 192, pp. 4991-5002, 2021. doi: https://doi.org/10.1016/j.procs.2021.09.277.

Hariom Sharan Sinha

Hariom Sharan Sinha has obtained M.Tech, PhD (CSE) from JNU New Delhi. He is working as an associate professor in the Department of Computer Science and Engineering at Adamas University, Barasat, Kolkata, West Bengal, India. He has more than nine years of experience in teaching and eleven years of experience in research.

Saket Kumar Choudhary

Saket Kumar Choudhary is an assistant professor in CSE, GITAM University, Bengaluru, Karnatka, India. He has obtained his master degrees in Mathematics from the University of Allahabad, Allahabad, India in 2005, Master of Computer Application (MCA) from UPTU, Lucknow, India in 2010, Master of Technology (M.Tech) from Jawaharlal Nehru University, New Delhi, India in 2014. He is Ph.D (Computer Science and Technology) School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India in 2017. His research interest includes mathematical modeling and simulation, dynamical systems, computational neuroscience: modeling of single and coupled neurons, computer vision, digital image processing, machine learning and artificial intelligence.

Vijendra Kumar Solanki

Vijender Kumar Solanki, Ph.D., is an Associate Professor in Department of Computer Science & Engineering, CMR Institute of Technology (Autonomous), Hyderabad, TS, India. He has more than 15 years of academic experience in network security, IoT, Big Data, Smart City and IT. Prior to his current role, he was associated with Apeejay Institute of Technology, Greater Noida, UP, KSRCE (Autonomous) Institution, Tamilnadu, India & Institute of Technology & Science, Ghaziabad, UP, India. He has attended an orientation program at UGC-Academic Staff College, University of Kerala, Thiruvananthapuram, Kerala &Refresher course at Indian Institute of Information Technology, Allahabad, UP, India. He has authored or co-authored more than 75 research articles that are published in journals, books and conference proceedings. He has edited or co-edited 12 books in the area of Information Technology. He teaches graduate & post graduate level courses in IT at ITS. He received Ph.D in Computer Science and Engineering from Anna University, Chennai, India in 2017 and ME, MCA from Maharishi Dayanand University, Rohtak, Haryana, India in 2007 and 2004, respectively and a bachelor's degree in Science from JLN Government College, Faridabad Haryana, India in 2001.He is Editor in International Journal of Machine Learning and Networked Collaborative Engineering (IJMLNCE) ISSN 2581-3242, Associate Editor in International Journal of Information Retrieval Research (IJIRR), IGI-GLOBAL, USA,ISSN: 2155-6377 | E-ISSN: 2155-6385 also serving editorial board members with many reputed journals. He has guest edited many volumes, with IGI-Global, USA, InderScience & many more reputed publishers.

LA UNIVERSIDAD EN INTERNET

Rectorado
Avenida de la Paz, 137
26006 Logroño (La Rioja)
t (+34) 941 21 02 11

www.unir.net