

Multilayer Feedforward Neural Network for Internet Traffic Classification

N. Manju^{1*}, B. S. Harish^{2*}, N. Nagadarshan¹

¹ Department of Information Science and Engineering, Sri Jayachamarajendra College of Engineering, Mysuru (India)

² Department of Information Science and Engineering, JSS Science and Technology University, Mysuru (India)

Received 1 April 2019 | Accepted 4 November 2019 | Published 25 November 2019



ABSTRACT

Recently, the efficient internet traffic classification has gained attention in order to improve service quality in IP networks. But the problem with the existing solutions is to handle the imbalanced dataset which has high uneven distribution of flows between the classes. In this paper, we propose a multilayer feedforward neural network architecture to handle the high imbalanced dataset. In the proposed model, we used a variation of multilayer perceptron with 4 hidden layers (called as mountain mirror networks) which does the feature transformation effectively. To check the efficacy of the proposed model, we used Cambridge dataset which consists of 248 features spread across 10 classes. Experimentation is carried out for two variants of the same dataset which is a standard one and a derived subset. The proposed model achieved an accuracy of 99.08% for highly imbalanced dataset (standard).

KEYWORDS

Classification, Feature Transformation, Feedforward Network, Internet Traffic.

DOI: 10.9781/ijimai.2019.11.002

I. INTRODUCTION

RECENTLY, the internet traffic classification has been frequently used in order to improve service quality in enterprise networks, developing new service packets, sharing the bandwidth source and performing traffic analysis effectively. There are various methods of internet traffic classification such as port based, payload based and flow statistical based approach. Port based classification is very simple but produces inaccurate results because of dynamic assignment of protocols [1], [2]. Payload based method have been proposed and presented by many researchers, which analyzes packet content to identify the internet traffic. Payload method is popular but has few limitations like lawful inspection, encrypted data cannot be handled and so on. To overcome these issues, a flow statistical based approach is used for IP traffic classification [3], [4]. The above said classification problems can be solved using machine learning approaches.

Machine learning consists of supervised learning and unsupervised learning. In unsupervised learning, class labels are unknown and a common technique of this type is called as clustering. In clustering, identical features are grouped as cluster. On the other hand, the classification can be performed using supervised learning. Supervised learning tries to figure out classes from the training data, and test data have been measured by studying the learning data [5]. The most used data discriminators in flow based method is given in [6] for the assessment of the classification task.

The paper is organized as follows: Section II presents the literature, section III depicts the proposed model, section IV discusses the results and section V presents the conclusion with future work.

II. RELATED WORK

A lot of network applications run on the Internet and generate a large amount of traffic every day. It is a daunting task to manage the huge and mixture traffic. Network traffic classification is more and more important along with the development of Internet.

The basic variants of machine learning approaches are supervised and unsupervised learning. Unsupervised method contains only the desired input and there are no output labels. A common unsupervised method is known as clustering [7], [8]. The supervised method is applied to analyze the learning data. Supervised learning method has both desired inputs and desired outputs. One of the initial important works in this area is the study in [9] which classifies the network traffic by applying the supervised naive bayes method on the flow statistical features. Classification of internet traffic is done by analyzing a packet via the capturing tool and determining the 248 features in [10]. Later, traffic classification is done using support vector machine with an accuracy of 90% on various types of internet traffic datasets [11]. Bayesian neural networks is applied for Cambridge dataset and achieved a class wise accuracy using 2441 flows for 7 classes ranging from 91% to 99.9% of accuracy [12]. Various classification algorithms are compared and evaluated in [13]. Such classification algorithms are: Naïve Bayes using both Naïve Bayes Descritization (NBD) and Kernel density estimation (NBK), C4.5 decision tree, Bayesian Network (BayesNet) and Naïve Bayes Tree (NBTree). Experimentation result shows that the C4.5 identifies the network flows faster than any other

* Corresponding author.

E-mail addresses: manjun007@sjce.ac.in (N. Manju), bsharish@jssstuniv.in (B. S. Harish).

four algorithms with respect to computational performance. Further, other studies reported using the machine learning algorithms Support Vector Machine (SVM) and Decision Tree. The result shows that; decision tree is computationally efficient compared to SVM as seen in [14]. Identification of Voice over Internet Protocol (VoIP) using machine learning approaches such as C5.0, AdaBoost and Genetic Programming is presented in [15]. Result shows that, C5.0 performs better compared to other two algorithms.

Artificial Immune System (AIS) is proposed and used to classify the malicious packets in Intrusion Detection System [16]. Inspired by this approach, AIS based algorithms are proposed to classify the flow of internet traffic and the performance of the proposed method is evaluated with and without using kernel function [17]. The size of the dataset used is 1000 flows. The range of the accuracy achieved is between 81.8% and 91% without using kernel function. Using linear kernel, the performance achieved was 92.3% of accuracy. Other two algorithms such as Naïve Bayes and SVM achieved an accuracy of 82.2% and 44.10% respectively [18]. Imbalance nature of the dataset used for the experimentation affects the algorithm proposed for the few classes as stated in [19]. Data Gravitation based Classification (DGC) model is proposed in [20] and 12 UCI datasets were used for the experimentation. DGC achieves better accuracy. However, it achieved less accuracy for imbalanced data sets. Hence, DGC suffers in using the unbalanced datasets. To address this problem, Imbalanced DGC (IDGC) model is proposed and Amplified Gravitation Coefficient (AGC) is used to compute the gravitation [21]. Based on the results of the proposed models, it can be concluded that most of the standard machine learning classification models are not much effective with imbalanced internet traffic data [22]. To enhance the performance of imbalance classification, Under-sampling DGC is proposed in [23]. The proposed method generates a set of training groups for re-balancing. Generated new training set are selected and employed to DGC model to get better classification accuracy. Later, voting policy is used to obtain the classification model. Therefore, it gives promising result for the imbalanced dataset. The proposed method is applied on 22 less imbalanced and 22 high imbalanced datasets. Class Oriented Feature Selection (COFS) is proposed to improve the recall and precision for minority classes maintaining overall high accuracy in [24]. Cost sensitive learning is studied for multiclass imbalance problem in [25] and misclassification cost with respect to dependent class is investigated. Weighted Cost Matrix (WCM) is proposed and compared with Flow rate based Cost Matrix (FCM). The obtained result gives better accuracy for the proposed method.

A. Motivation

The work is carried out for two reasons such as, a) To build multi layer neural network and b) to propose a model which can handle the class imbalance problem. Authors in [26] has proposed Extreme Learning Machine (ELM) which is used to classify the internet traffic. Extreme Learning Machine is a fast learner neural network model which randomly takes hidden layer weights. It was proposed for the training of single hidden layer feedforward neural networks. The weight and threshold values in traditional single hidden layer feedforward neural networks should be renewed by gradient-based learning algorithms. To provide a good solution, this situation may cause both an extension of the learning process and the error may be encountered in a local point. The change can be made in the momentum value for the problem of errors being encountered in a local point. However the change will not make a contribution to the learning process. The weights and thresholds in a single hidden layer feedforward network do not affect the performance of the network.

As the application of neural networks is increasing day to day, we exploited the application of multilayer feed-forward neural network

to solve the problem of Internet traffic classification. The proposed network is used towards the basic deep neural networks and thus we used multiple layers in our architecture. The main objective of using the architecture of multi layer feedforward neural network is because of the capability of handling large amount of datasets. Multilayer feedforward has the ability to detect complex nonlinear relationships between dependent and independent variables. Further, Multilayer feedforward exhibits the various advantages like adaptability in learning, non-linearity property, mapping between input and output, robustness in accommodating the noise. Using machine learning with feedforward neural network with multiple layers and with other parameters would be able to construct the deep learning architecture in near future.

III. PROPOSED METHODOLOGY

In this model, we are proposing a multi layer feedforward neural network for internet traffic classification. The main objective of using this network is that, it has high learning rate on non linear data [27]. It is quite obvious that, all the attributes of the dataset description are not same. Hence, it is highly difficult to classify without normalizing the data. Therefore, we first normalize the data before sending it to the proposed model. Normalizing is done using the eq. (1):

$$f(x) = (x - \mu) / \sigma \quad (1)$$

Where,

x is the column vector (features).

μ is the mean of x .

σ is the standard deviation of x .

After normalizing the features, the new features will be in the range of $[-1, 1]$. This helps us to classify the data without having bias towards the high biased features.

The proposed methodology uses multilayer feedforward neural network with six layers. Out of six layers, four are hidden layers, one is input layer, and one is output layer. Finally we use the softmax activation function for classification.

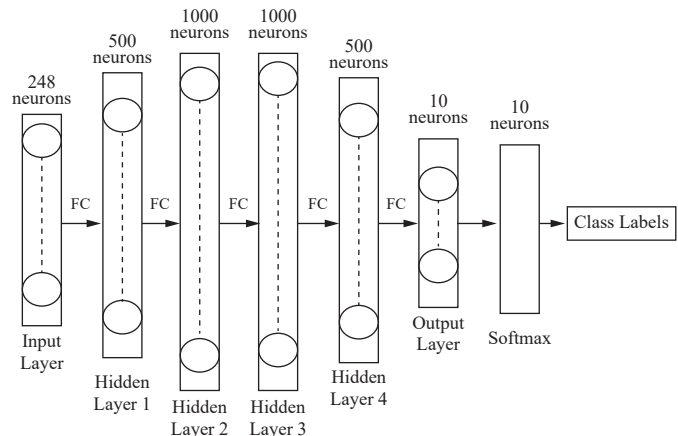


Fig.1. Proposed Multi Layer Feedforward Neural Network Model (FC refers to Fully Connected).

As we can see from Fig.1, the input layer consists of 248 neurons because of 248 features. The first hidden layer consists of 500 hidden neurons, the second and third hidden layers consist of 1000 hidden neurons each. Fourth and the final hidden layer consist of 500 hidden neurons. The final output layer consists of 10 neurons as there are 10 classes. At last we use softmax as the loss function.

Assuming the preprocessed features are encoded, we first extract the features and make it 1000 dimensions by decoding the features. After the model understands the data deeply, it is again transformed to the lower dimensions to get only the required features by encoding. This process can be understood by the loss functions. In this procedure, all the hidden layers can be viewed as feature transformational units.

The first layer as mentioned earlier is known as the input layer whose number of neurons are always equal to the number of attributes in the dataset. The dataset consists 248 attributes.

The next layer is the first hidden layer. In this layer, all the neurons of the previous layers are connected to all the neurons of the next layer through dot product of the weights (eq. (2)).

$$f(x) = wx + b \quad (2)$$

where,

w is the weights of the current layer

x is the output of the previous layer (here it the input features)

b is the bias.

Initially weights are initialized with normal Gaussian values. This type of weight initialization is used across various neural network descendants like Convolution Neural Network, Recurrent Neural Network, etc. The next layer i.e., the first hidden layer consists of 500 neurons. To achieve better performance, the number of hidden layers in multi layer perceptron should be as twice the number of neurons which takes the advantage mentioned in [28]. Therefore, we double and round off the number of input features to get 500 neurons. Further, we apply eq. (2) and we pass it to the activation function or kernel. The authors in [26] used wavelet kernel as their activation function. The proposed model uses Rectified Linear Unit (ReLU shown in eq. (3)) as the activation function as it is good in capturing data with moderate and large number of features [29]. Further it also helps in faster convergence and it is by far simpler than wavelet kernel.

$$f(x) = \max(0, X) \quad (3)$$

where,

x is the input to the ReLU function

Fig. 2 shows the graph of the ReLU activation function, which squashes every negative input to zero and keeps every positive input as the same.

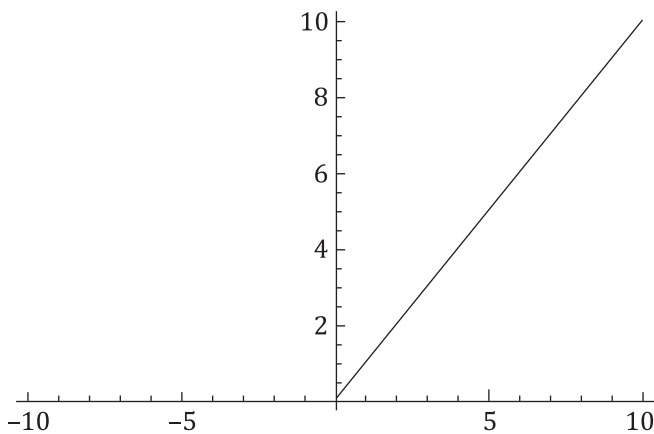


Fig. 2. Graph of the ReLU activation function.

The next layer, i.e., second hidden layer consists of 1000 neurons as it is still needed to extract the features from the data that are going to double the number of neurons from the previous layer. In particular, this layer helps in understanding the data much deeper. In this work,

ReLU is used as the activation function. The following layer is the same as previous layer, which also contains 1000 neurons and uses ReLU as the activation function.

In the third hidden layer, we tend to project it to the lower dimensional to reduce the number of neurons and hence it contains 500 neurons. It also uses ReLU as the activation function. The main objective of doing this is to extract only the relevant features.

The last layer is the output which contains 10 neurons as there are 10 classes. After the output layer we have the softmax layer which is used as the loss function and is given by:

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \quad (4)$$

where,

y_i is the score of class j

e is the standard mathematical constant ≈ 2.718

The softmax function is chosen among other loss functions because the range of the softmax is [0, 1] which does not require more space and the sum of all the scores is equal 1. Hence it is a probabilistic score and losses can be predicted relatively.

The back propagation algorithm is used to back propagate through the network updating the weights through adam optimizer [30]. It is used for optimization which automatically adapts and updates its parameters. This model is executed for 1000 epochs with a batch size of 2486. The increase in number of epochs leads to the over fitting problem. Therefore epoch was set to 1000.

IV. EXPERIMENTAL SECTION

A. Experimental Setup

The experiment was carried on a machine with the operating system ubuntu version 16.04 with 12 GB RAM and the secondary storage capacity of 1TB. The processor is Intel Core i7-7700HQ CPU at 2.80GHz \times 8. Implementation is done using a 64 bit machine.

B. Dataset

The dataset used is a publicly available Cambridge dataset which consists of 248 features. The total number of classes are 10. Two variants of the same dataset are used. One standard dataset is highly imbalanced and the other is less imbalanced, which is a derived dataset. In the standard dataset, flows are more biased towards WWW class and least biased towards multimedia class. The dataset is divided in the ratio of 70% for training and 30% for testing. Table I shows the characteristics of datasets used for experimentation.

C. Results and Discussions

In the following section, we discuss the results obtained on the dataset01 which is the high imbalanced dataset. The main objective of this experimentation is to overcome the problem of imbalance dataset which is one of the major challenges in the research. The proposed model avoids the bias result and gives the better result.

As we can see from Table II, which contains the confusion matrix for imbalanced dataset, almost all the classes have been classified correctly which lead to an accuracy of 99.08%, which is around 3% greater than the accuracy achieved in [26]. Hence the proposed model achieves a greater result. We can also see that the number of misclassification samples of each class is far less than the number of samples correctly classified samples.

TABLE I. CHARACTERISTICS OF DATASETS USED FOR EXPERIMENTATION

Class No.	Class Name	Standard Dataset (Dataset01)	Flow distribution Ratio in %	Derived Dataset (Dataset02)	Flow distribution Ratio in %
		Flow Count		Flow Count	
1	ATTACK	122	00.49	1793	07.50
2	DATABASE	238	00.95	2648	11.08
3	FTP-CONTROL	149	00.59	3000	12.55
4	FTP-DATA	1319	05.30	3000	12.55
5	TFP-PASV	43	00.17	2688	11.24
6	MAIL	4146	16.67	3000	12.55
7	MULTIMEDIA	87	00.34	576	02.41
8	P2P	339	01.36	2094	08.76
9	SERVICES	206	00.82	2099	08.78
10	WWW	18211	73.25	3000	12.55
Total Number of Flows		24860	~100.00	23898	~100.00

TABLE II. CONFUSION MATRIX OF 10 CLASSES FOR DATASET01

		Actual									
		Class	1	2	3	4	5	6	7	8	9
Predicted	1	30	0	0	0	0	0	3	1	0	14
	2	0	68	0	0	1	0	0	0	0	3
	3	0	0	42	0	0	5	0	0	0	0
	4	0	0	0	369	0	1	0	1	0	1
	5	0	0	0	0	7	0	0	3	0	2
	6	2	1	0	0	0	1195	0	0	0	10
	7	2	0	1	0	1	0	17	0	0	0
	8	0	0	0	0	0	0	2	81	0	2
	9	0	0	0	0	0	0	0	0	71	0
	10	0	2	0	0	2	3	0	5	0	5510

TABLE III. TABLE SHOWING, RECALL, PRECISION AND F₁ SCORE OF ALL THE CLASSES FOR DATASET01

Class	Recall	Precision	F ₁ Score
1	0.88	0.62	0.73
2	0.95	0.94	0.95
3	0.97	0.89	0.93
4	1.00	0.99	0.99
5	0.63	0.58	0.60
6	0.99	0.98	0.99
7	0.77	0.80	0.79
8	0.89	0.95	0.92
9	1.00	1.00	1.00
10	0.99	0.99	0.99
Average	0.91	0.87	0.89

The table III shows that we have compelling precision and recall values. As we can see the average precision is of 0.87, which is quite high. The average recall value is 0.91 that is also very high, which shows low false-negative rate. The F₁ score of 0.89 shows a good result for a dataset01 shown in Table I. The Precision, Recall and F₁ score show that the model is successful in handling even the less balanced data.

Further, we discuss the results achieved using the dataset02 which is a less imbalanced (derived) dataset.

TABLE IV. CONFUSION MATRIX OF 10 CLASSES FOR DATASET02

		Actual									
		Class	1	2	3	4	5	6	7	8	9
Predicted	1	472	3	2	2	0	7	4	9	1	22
	2	0	780	1	0	2	0	0	1	0	0
	3	1	0	890	0	0	12	1	0	1	0
	4	1	0	0	907	1	6	0	0	0	1
	5	1	1	1	1	774	0	0	4	0	2
	6	4	1	5	2	1	908	0	0	1	5
	7	9	0	0	3	2	3	160	2	0	0
	8	12	5	0	0	7	7	8	593	0	2
	9	0	3	1	0	0	5	0	1	637	1
	10	6	4	3	0	4	0	0	0	2	852

As we can see from Table IV, the confusion matrix for dataset02, almost all the classes have been classified correctly, which leads to an accuracy of 97.25%, which is again much greater than existing in [26].

TABLE V. RECALL, PRECISION AND F₁ SCORE OF ALL THE CLASSES FOR DATASET02

Class	Recall	Precision	F ₁ Score
1	0.93	0.90	0.91
2	0.97	0.99	0.98
3	0.98	0.98	0.98
4	0.99	0.99	0.99
5	0.97	0.98	0.98
6	0.95	0.97	0.96
7	0.92	0.89	0.90
8	0.97	0.93	0.95
9	0.99	0.98	0.98
10	0.96	0.97	0.97
Average	0.96	0.95	0.96

Table V presents the result of Precision, Recall and F₁ Score. From the results, we can observe that the average Precision is of 0.95, which is quite high for imbalanced dataset. The average Recall value of 0.96 is also very high, which shows low false-negative rate. The F₁ score of 0.96 shows a good result for a highly balanced dataset.

TABLE VI. ACCURACY COMPARISON TABLE

Methodology	Activation Function	Number of hidden layers	Accuracy in %
ELM [26]	Sigmoid	1	79.16
ELM [26]	Tangent Sigmoid	1	92.11
ELM [26]	Radial basis	1	87.30
ELM [26]	Wavelet	1	96.57
Standard dataset (Dataset01)	ReLU	4	99.08
Derived dataset (Dataset02)	ReLU	4	97.25

Table VI shows that ELM neural network with Sigmoid kernel and using one hidden layer achieved an accuracy of 79.16% and ELM with Tangent Sigmoid kernel achieved 92.11% of accuracy. ELM with Radial basis attained 87.30% accuracy and ELM with Wavelet kernel achieved 96.57% of accuracy. Further, the proposed model outperformed with an accuracy of 99.08% for dataset01 and 97.25% accuracy for dataset02, which is more than the accuracy reported in [26].

The increase in accuracy is due to the design architecture of neural network model discussed in section III. The model first extracts the number of features and projects it to the higher dimensions. Later, the model understood the data deeply and it is transformed to the lower dimensions to get only the required features. Hence it is a fact that the proposed classifier has successfully solved the problem of internet traffic classification for both less imbalanced and highly imbalanced dataset. Even though the model is intended for high imbalanced dataset, table IV, V and VI justifies that it also performed well for a less imbalanced dataset.

V. CONCLUSION

Currently, internet traffic classification is becoming popular due to growth in the internet usage and utility applications. However, network management is a key challenge to the service providers. Efficient network management leads to provide quality of service, ensure security and high computing performance of the network. Another issue with the internet traffic is about handling imbalance dataset. Hence there is a need of addressing the existing problem, i.e internet data that comes with imbalance in particular. Thus, to solve this issues, we proposed a multi layer feedforward neural network model with four hidden layers (mountain mirror network). The effective transformation of the dimensions of the features helps in accurate classification and thereby increasing the accuracy. In future, we can also try with various kernels to compare the results achieved. Further, we are also intending to use dimensionality reduction technique to reduce the higher dimensional internet traffic data to lower dimensions. To test the scalability of our proposed method, we also plan to work on datasets of massive traffic in real networks.

REFERENCES

- [1] Kim, H., Claffy, K. C., Fomenkov, M., Barman, D., Faloutsos, M., and Lee, K., "Internet traffic classification demystified: myths, caveats, and the best practices," Proceedings of the ACM CoNEXT conference, ACM, 2008.
- [2] Callado, A., Kamienski, C., Szabó, G., Gero, B. P., Kelner, J., Fernandes, S., and Sadok, D., "A survey on internet traffic identification," IEEE Communications Surveys & Tutorials, vol. 11, no. 3, pp. 37-52, 2009.
- [3] Finsterbusch, M., Richter, C., Rocha, E., Muller, J. A., and Hanssgen, K., "A survey of payload-based traffic classification approaches," IEEE Communications Surveys & Tutorials, vol. 16, no. 2, pp. 1135-1156, 2014.
- [4] Ducange, P., Mannarà, G., Marcelloni, F., Pecori, R., and Vecchio, M., "A novel approach for internet traffic classification based on multi-objective evolutionary fuzzy classifiers," IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1-6, July, 2017.
- [5] Namdev, N., Agrawal, S., and Silkari, S., "Recent advancement in machine learning based internet traffic classification," Procedia Computer Science, vol. 60, pp.784-791, 2015.
- [6] Moore, A., Zuev, D., and Crogan, M., "Discriminators for use in flow-based classification," 2013.
- [7] J. Erman, A. Mahanti, M. and Arlitt, C. Williamson, "Identifying and discriminating between web and peer-to-peer traffic in the network core," Proceedings of the 16th International Conference on World Wide Web, ACM, pp. 883–892, 2007.
- [8] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data, ACM, pp. 281–286, 2006.
- [9] A.W. Moore, and D. Zuev, "Internet traffic classification using bayesian analysis techniques," ACM SIGMETRICS Performance Evaluation Review, vol. 33, no. 1, ACM, pp. 50–60, 2005.
- [10] A. Moore, D. Zuev, and M. Crogan, "Discriminators for Use in Flow-Based Classification," Queen Mary and Westfield College, Department of Computer Science, 2005.
- [11] A. Este, F. Gringoli, and L. Salgarelli, "Support vector machines for TCP traffic classification," Computer Networks, vol. 53, no. 14, pp. 2476–2490, 2009.
- [12] T. Auld, A.W. Moore, and S.F. Gull, "Bayesian neural networks for internet traffic classification," IEEE Transaction on Neural Networks, vol. 18, no. 1, pp. 223–239, 2007.
- [13] N. Williams, S. Zander, and G.A. Armitage, "Preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification," ACM SIGCOMM Computer. Communication. Review, vol. 36, no. 5, pp. 5–16, 2006.
- [14] Valenti, S., Rossi, D., Dainotti, A., Pescapè, A., Finamore, A., and Mellia, M., "Reviewing traffic classification," In Data Traffic Monitoring and Analysis, pp. 123-147, Springer, Berlin, Heidelberg, 2013.
- [15] Alshammari, R., & Zincir-Heywood, A. N., "Identification of VoIP encrypted traffic using a machine learning approach," Journal of King Saud University-Computer and Information Sciences, vol. 27, no. 1, pp. 77-92, 2015.
- [16] Kim, J., Bentley, P. J., Aickelin, U., Greensmith, J., Tedesco, G., and Twycross, J., "Immune system approaches to intrusion detection—a review," Natural computing, vol. 6, no. 4, pp. 413-466, 2007.
- [17] Schmidt, B., Kountanis, D., and Al-Fuqaha, A., "Artificial immune system inspired algorithm for flow-based internet traffic classification," IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom), pp. 664-667, December, 2014.
- [18] Schmidt, B., Kountanis, D., and Al-Fuqaha, A. "A Biologically-Inspired Approach to Network Traffic Classification for Resource-Constrained Systems," In Proceedings of the 2014 IEEE/ACM International Symposium on Big Data Computing IEEE Computer Society, pp. 113-118, December, 2014.
- [19] Schmidt, B., Al-Fuqaha, A., Gupta, A., and Kountanis, D., "Optimizing an artificial immune system algorithm in support of flow-Based internet traffic classification," Applied Soft Computing, vol. 54, pp. 1-22, 2017.
- [20] Peng, L., Yang, B., Chen, Y., and Abraham, A., "Data gravitation based classification," Information Sciences, vol. 179, no. 6, pp. 809-819, 2009.
- [21] Peng, L., Zhang, H., Yang, B., and Chen, Y., "A new approach for imbalanced data classification based on data gravitation," Information Sciences, vol. 288, pp. 347-373, 2014.
- [22] Peng, L., Zhang, H., Chen, Y., and Yang, B., "Imbalanced traffic identification using an imbalanced data gravitation-based classification model," Computer Communications, vol. 102, pp. 177-189, 2017.
- [23] Peng, L., Yang, B., Chen, Y., and Zhou, X., "An under-sampling imbalanced learning of data gravitation based classification," IEEE 12th International Conference on Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), pp. 419-425, August, 2016.
- [24] Liu, Z., Wang, R., Tao, M., and Cai, X., "A class-oriented feature selection approach for multi-class imbalanced network traffic datasets based on local and global metrics fusion," Neurocomputing, vol. 168, pp. 365-381, 2015.
- [25] Zhen, L. I. U., and Qiong, L. I. U., "Studying cost-sensitive learning for multi-class imbalance in Internet traffic classification," The Journal of China Universities of Posts and Telecommunications, vol. 19, no. 6, pp. 63-72, 2012.
- [26] Ertam, F. and Avci, E., "A new approach for internet traffic classification: GA-WK-ELM," Measurement, vol. 95, pp.135-142, 2017.
- [27] Panchal, G., Ganatra, A., Kosta, Y. P., and Panchal, D., "Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers," International Journal of Computer Theory and Engineering, vol. 3, no. 2, pp. 332-337, 2011.
- [28] Panchal, G., Ganatra, A., Kosta, Y.P. and Panchal, D., "Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers," International Journal of Computer Theory and Engineering, vol. 3, no. 2, p.332, 2011.
- [29] Agarap, A. F., "Deep learning using rectified linear units (relu)," arXiv preprint arXiv:1803.08375, 2018.
- [30] Kingma, D. P., and Ba, J. "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.



N. Manju

Obtained his B.E in Computer Science and Engineering in the year 2005 and M.Tech in Computer Network Engineering in 2008 from Visvesvaraya Technological University, Belagavi, Karnataka, India. He is presently working as an Assistant Professor in the Department of Information Science & Engineering, Sri Jayachamarajendra College of Engineering, Mysuru, Karnataka, India. He is a Life Member of ISTE. His area of interest includes Machine Learning and Computational Intelligence.



B. S. Harish

Obtained his B.E in Electronics and Communication (2002), M.Tech in Networking and Internet Engineering (2004) from Visvesvaraya Technological University, Belagavi, Karnataka, India. He completed his Ph.D. in Computer Science (2011); thesis entitled "Classification of Large Text Data" from University of Mysore. He is presently working as an Associate Professor in the Department of Information Science & Engineering, JSS Science & Technology University, Mysuru. He was invited as a Visiting Researcher to DIBRIS - Department of Informatics, Bio Engineering, Robotics and System Engineering, University of Genova, Italy from May- July 2016. He delivered various technical talks in National and International Conferences. He has been invited as a resource person to deliver various technical talks on Data Mining, Image Processing, Pattern Recognition, Soft Computing. He is also serving and served as a reviewer for National, International Conferences and Journals. He has published more than 50 International reputed peer reviewed journals and conferences proceedings. He successfully executed AICTE-RPS project which was sanctioned by AICTE, Government of India. He also served as a secretary, CSI Mysore chapter. He is a Member of IEEE (93068688), Life Member of CSI (09872), Life Member of Institute of Engineers and Life Member of ISTE. His area of interest includes Machine Learning, Text Mining and Computational Intelligence.



N. Nagadarshan

Obtained his B.E in Information Science and Engineering in the year 2019 from Sri Jayachamarajendra College of Engineering. His area of interests are Deep Learning, Biometrics and also he is very much passionate about Spiking Neural Networks. He also interned at Indian Statistical Institute, Kolkata. He is currently working on Spiking Neural Networks at Indian Institute of Science, Bengaluru.