International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 09 | Sep 2019 www.irjet.net p-ISSN: 2395-0072

Comparative study on Embedded Feature Selection Techniques for Internet Traffic Classification

M. Anitha[1]

[1]*Research Scholar, School of Computer Science, Engineering and Applications, Bharathidasan University, Trichy*

Abstract - The Internet has become the backbone of human communication. Now a day's every electronic gadget is built to communicate over the Internet. Internet traffic is heterogeneous and consists of traffic flows from a variety of applications. In the current scenario online services such as email, social networks, multimedia communication, and https traffic have become an essential need for human beings. Many new applications emerge every day and are unique and have their own requirements with the respect to network criteria's. To identify the traffic based on the application in a large network, the major part is traffic classification and it is useful to provide quality of service, lawful interception and intrusion detection. A number of limitations have been exhibited by older methods such as port-based and payload based classification. Hence Machine learning techniques are used by the research community to analyses the flow statistics for detecting network applications. The embedded based approach is used here for traffic identification and classification process which includes the classification algorithms such as Naives Bayes algorithm, Random forest and Support Vector Machine (SVM). After that implement deep learning algorithm named as Multilayer Perceptron (MLP) applied on network traffic datasets. Finally analyze the performance of the system in terms of error metrics and experimental results shows that the proposed deep learning approach can be outperformance than the existing machine learning techniques.

***Key Words*: Internet traffic, Classification, Machine learning algorithm, Deep learning, Embedded approach**

1. INTRODUCTION

A computer network is a digital telecommunications network which allows nodes to share resources. In computer networks, computing devices exchange data each other using connections data links between nodes. These data links are established over cable media such as wires or optic cables, or wireless media such as WI-FI. Network computer devices that originate, route and terminate the data are called network nodes. Nodes are generally identified by network addresses, and can include hosts such as personal computers, phones, and servers, as well as networking hardware such as routers and switches. Two such devices can be said to be networked together when one device is able to exchange information with the other device, whether or not they have a direct connection to each other. In most cases, application specific communication protocols are layers (i.e. carried as payload) over other more general communications protocols. This formidable collection of information technology requires skilled network management to keep it all running reliably. Computer networks support an enormous number of applications and services such as access to the World Wide Web, digital video, digital audio, shared use of application and storage servers, printers, and fax machines, and use of email and instant messaging applications as well as many others.

Computer networks differ in the transmission medium used to carry their signals, communications protocols to organize network traffic, the network's size, topology, traffic control mechanism and organizational intent. The best-known computer network is the Internet. Traffic classification is an automated process which categorizes computer network traffic according to various parameters (for example, based on port number or protocol) into a number of traffic classes. Each resulting traffic class can be treated differently in order to differentiate the service implied for the data generator or consumer. Packets are classified to be differently processed by the network scheduler. Upon classifying a traffic flow using a particular protocol, a predetermined policy can be applied to it and other flows to either guarantee a certain quality (as with VoIP or media streaming se Computer communication links that do not support packets, such as traditional point-to-point telecommunication links, simply transmit data as a bit stream. However, the overwhelming majority of computer networks carry their data in packets. A network packet is a formatted unit of data (a list of bits or bytes, usually a few tens of bytes to a few kilobytes long) carried by a packet-switched network. Packets are sent through the network to their destination. Once the packets arrive they are reassembled into their original message.

2. Related Work

Traffic classification through the network consists of flows from different applications. Some of the applications are sensitive to delay whereas others are insensitive to delay. So that the requirements vary for each and every applications. So traffic has to be classified. Traffic classification is the process of identifying and classifying the protocols or applications

available in the network. Traffic classification is beneficiary for network management and providing QOS support, network security. With traffic classification we can identify the insights of traffic.

NeerajNamdev, et.al,…[1] implemented supervised learning based on attributes of a class i.e. in this we choose samples on the basis of attributes collected by the whole data. The machine learning is provided with a collection of sample instances, pre-classified into classes. The output of the learning process is a classification model that is constructed by examining generalizing from providing instances. In classification approaches mainly have two phases (steps), training and testing. Learning phase that examine the provided data (called the training dataset) and constructs (builds) a classification model. And the model that has been built in the training phase is used to classify new unseen instances, in this paper we discuss the some well-known supervised machine learning techniques and discuss also about issues related to different techniques. Bayes Net approach generally known as Belief Network. It is a Probabilistic model which uses the graph model to represent the set of random variables and their conditional dependencies. Bayes Net uses the concept of directed acyclic graph (DAG) to represent the set, in which each node represent a variable and edges among the nodes represent the relative dependencies between random variables and these relative dependencies in the graph are calculated by well-known statistical and computational methods. There are two phases of bayes net approach first phase is learning of network structure, in which uses various types of search algorithm like hill climbing, tabu search etc. for identified a good network structure and second is estimate probabilistic table for each random variable

TaimurBakhshi, et.al,…[2] analyzed Traffic classification utilizing flow measurement enables operators to perform essential network management. Flow accounting methods such as NetFlow are, however, considered in adequate for classification requiring additional packet-level information, host behavior analysis, and specialized hardware limiting their practical adoption. This paper aims to overcome these challenges by proposing two-phased machine learning classification mechanism with NetFlow as input. The individual flow classes are derived per application through K-means and are further used to train a C5.0 decision tree classifier. As part of validation, the initial unsupervised phase used flow records of fifteen popular Internet applications that were collected and independently subjected to fuzzy means clustering to determine unique flow classes generated per application

Yang Hong, et.al,…[3] implemented the technique for accurate and timely traffic classification is a key to providing Quality of Service (QoS), application-level visibility, and security monitoring for network operations and management. A class of traffic classification techniques have emerged that apply machine learning technology to predict the application class of a traffic flow based on the statistical properties of flow-features. In this paper, we propose a novel iterative-tuning scheme to increase the training speed of the classification algorithm using Support Vector Machine (SVM) learning. Meanwhile we derive the equations to obtain SVM parameters by conducting theoretical analysis of iterative-tuning SVM. Traffic classification is carried out using flow-level information extracted from NetFlow data. Performance evaluation demonstrates that the proposed iterative-tuning SVM exhibits a training speed that is two to ten times faster than eight other previously proposed SVM techniques found in the literature, while maintaining comparable classification accuracy as those eight SVM techniques. In the presence of millions of flows and Terabytes of data in the network, faster training speeds is essential to making SVM techniques a viable option for realworld deployment of traffic classification modules. In addition, network operators and cloud service providers can apply network traffic classification to address a range of issues including semi-real-time security monitoring and traffic engineering.

PietroDucange, et.al,…[4] discussed a preliminary study on the application of multi-objective evolutionary fuzzy classifiers for approaching the Internet traffic classification problem. In particular, we have concentrated our efforts on the possibility of generating interpretable classification models, also characterized by a good accuracy level. Indeed, we have experimented a state-of-the-art algorithm, namely PAES-RCS, for generating sets of fuzzy rule-based systems for classifying the Internet traffic flows extracted from two real-world networks. We have first performed a cross validation, considering the data extracted from each network as an independent classification dataset. Then, we have carried out a cross-network validation, where one network has been used for training the classifiers and the other network has been adopted for evaluating the generalization capability of the trained models. We have shown that the results achieved on the cross validation are promising. Indeed, the obtained fuzzy rule-based classifiers are characterized by very interesting trade-offs between their accuracy, expressed in terms of classification rate and also in terms of per class true positive and false positive rates, and their interpretability, expressed in terms of number of rules and total number of antecedents in each rules. As regards the cross-network evaluation, even though we have observed a substantial decrease of the accuracies, the generated classification models still maintain acceptable levels of classification rate, true positive rate and false positive rate.

LizhiPeng, et.al,…[5] set out to study the effectiveness of the early stage statistical features of Internet traffics. We try to answer the above mentioned question using the mutual information analysis and experimentalmethods. Trafficdatasetsand tenmachinelearningclassifiersareappliedforourexperiments.Weusetheapplication

layerpayloadsizesastheoriginalpacketlevelfeatures,and5statisticsasthestatistical features. Firstly, the mutual information of each feature and the traffic type label is computed to evaluate its effectiveness preliminary. Then we build 6 feature sets covering the pure original feature set, the pure statistical feature set and the hybrid featureset,andthenallselectedclassifiersareappliedonthesefeaturesetstovalidate the effectiveness of selected features

3.Traffic classification using classification techniques

The evolution of the Internet into a large complex service-based network has posed tremendous challenges for network monitoring and control in terms of how to collect the large amount of data in addition to the accurate classification of new emerging applications such as peer to peer, video streaming and online gaming. These applications consume bandwidth and affect the performance of the network especially in a limited bandwidth networks such as university campuses causing performance deterioration of mission critical applications. Traffic Classification is a method of categorizing the computer network traffic based on various features observed passively in the traffic into a number of traffic classes. Due to the rapid increase of different Internet application behaviors', raised the need to disguise the applications for filtering, accounting, advertising, network designing etc. Many traditional methods like port based, packets based and some alternate methods based on machine learning approaches have been used for the classification process. Some of the classification techniques are described below:

3.1 RANDOM FOREST

Random Forest is essentially an ensemble of un-pruned classification trees. It gives excellent performance on a number of practical problems, largely because it is not sensitive to noise in the data set, and it is not subject to over-fitting. It works fast, and generally exhibits a substantial performance improvement over many other tree-based algorithms. Random forests are built by combining the predictions of several trees, each of which is trained in isolation. Unlike in boosting where the base models are trained and combined using a sophisticated weighting scheme, typically the trees are trained independently and the predictions of the trees are combined through averaging. There are three main choices to be made when constructing a random tree. These are

- The method for splitting the leafs.
- The type of predictor to use in each leaf.
- The method for injecting randomness into the trees.

In Brieman's early work each individual tree is given an equal vote and later version of Random Forest allows weighted and unweighted voting. The technique on which Random Forest ensemble is formed can be considered over following parameters: i) Base Classifier: It describes the base classifier used in the Random Forest ensemble. Base classifier can be decision tree, Random tree, or extremely randomized tree.

ii) Split Measure: If base classifier of Random Forest is decision tree, then which split measure is found at each node of the tree to perform the splitting. To perform splitting Gini index, Info gain etc are used.

iii) Number of Passes: For building Random Forest classifier, if single pass is sufficient or multiple passes through data are needed

iv) Combine Strategy: In Random Forest ensemble, all the base classifiers generated are used for classification. At the time of classification, how the results of individual base classifiers are combined is decided by the combine strategy.

v) Number of attributes used for base classifier generation: This parameter gives the number of how many attributes are to be used which is randomly selected from the original set of attributes at each node of the base decision tree. Filter and Wrapper these are main techniques used for feature selection and extraction. Each tree of Random Forest is grown, are described as follows: Suppose training data size containing N number of records, then N records are sampled at random but with replacement, from the original data, this is known as bootstrap sample along with M number of attributes. This sample will be used for the training set for growing the tree. If there are N input variables, a number n << N is selected such that at each node, n variables are selected at random out of N and the best split on these m attributes is used to split the node. The value of m is held constant during forest growing. The decision tree is grown to the largest extent possible. A tree forms "inbag" dataset by sampling with replacement member from the training set. It is checked whether sample data is correctly classified or not using out of bag error with the help of out of bag data which is normally one third of the "inbag" data. Prediction is made by aggregating (majority vote for classification or averaging for regression) the predictions of the ensemble

The overall process of Random forest algorithm is shown in fig 2.

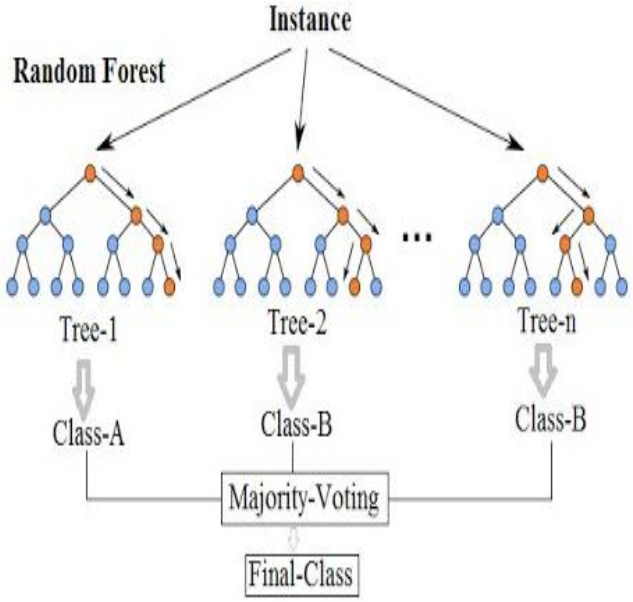


FIG 2: RANDOM FOREST ALGORITHM

3.2 NAIVES BAYES

The Naive Bayesian classifier is based on Bayes' theorem with independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods. Algorithm Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c). Naive Bayes classifier assume that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \text{ ---------- Eqn(1)}$$

$P(c|X) = P(x_1|c) * P(x_2|c)*....*P(x_n|c)*P(c)$

Where

- P(c|x) is the posterior probability of class (c, target) given predictor(x, attributes).
- P(c) is the prior probability of class.
- P(x|c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor

Naive Bayes technique is a supervised machine learning technique. Flow contents such as port numbers, flow length and time between consecutive flows are used to train the classifier. Moreover, to train the classifier 248 full-flow based features were used. The chosen traffic for application was categorized into different groups such as database, mail services, games and multimedia, www, p2p, bulk data transfer and attack

3.3 SUPPORT VECTOR MACHINE

Support vector machine analyzes the data, define the decision boundaries and uses the kernels for computation which are performed in input space. The input data are two sets of vectors of size m each. Then every data represented as a vector is classified in a particular class. Now the task is to find a margin between two classes that is far from any document. The distance defines the margin of the classifier, maximizing the margin reduces indecisive decisions. SVM also supports classification and regression which are useful for statistical learning theory and it helps recognizing the factors precisely, that needs to be taken

into account, to understand it successfully. It solves an optimization problem of finding the maximum margin hyperplane between the classes. This is basically required to avoid over fitting. Basically it is a linear classifier separating the classes which can be separated with the help of linear decision surfaces called hyperplanes. Foe classes having binary features SVM draws a line between the classes and for classes having multiple features hyper planes are drawn. However it can be used for classifying the non linear data also by transforming the feature space into the higher dimensional space so that nonlinear data in higher dimensional can be separated easily by a hyperplane. This transformation is made easy with the help of Kernel-trick. With the help of kernels it is not necessary to calculate all the dimensions when transform and calculation of hyperplane can be done in the same lower dimensional feature space. Kernels are not used only for this purpose but also for making the calculation easier in case of many features. Various kernels are used by machine learning approaches e.g. RBF(Radial Basis Function), Linear kernel, Poly kernel etc. The SVM algorithm is stated as follows:

Input : Training data Xi Labels Yi

Output: Sum of weight vector. α array, b and SV

Initialize α_i = 0, $f_i = -Y_i$

Compute $b_{high}, I_{high}, b_{low}, I_{low}$

Update α_{ihigh} and α_{ilow}

Repeat

Update f_i

Compute: $b_{high}, I_{high}, b_{low}, I_{low}$

Update α_{ihigh} and α_{ilow}

Until $b_{low} \leq b_{up}+2\tau$

Update the threshold b

Store the new α_1 and α_2 values

Update weight vector w if SVM is linear

Support Vector Machine (SVM), based on statistical learning theory, is known as one of the best machine learning algorithms for classification purpose and has been successfully applied to many classification problems such as image recognition, text categorization, medical diagnosis, remote sensing, and motion classification. SVM method is selected as classification algorithm due to its ability for simultaneously minimizing the empirical classification error and maximizing the geometric margin classification space. These properties reduce the structural risk of over-learning with limited samples.

3.4 MULTI LAYER PERCEPTRON

A multilayer perceptron (MLP) is a class of feed forward artificial neural network. An MLP consists of at least three layers of nodes. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a learning technique called back propagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable. Multilayer perceptrons are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer. A perceptron is a linear classifier; that is, it is an algorithm that classifies input by separating two categories with a straight line. In WEKA tool, choose classify option and pick the functions options to perform Multilayer perceptron based on class attribute. Input is typically a feature vector x multiplied by weights wand added to a bias

b: y = w * x + b.

The basic layout is shown in fig 4.

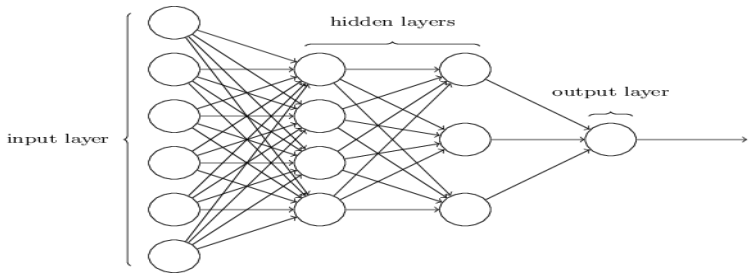


FIG 3: MLP FRAMEWORK

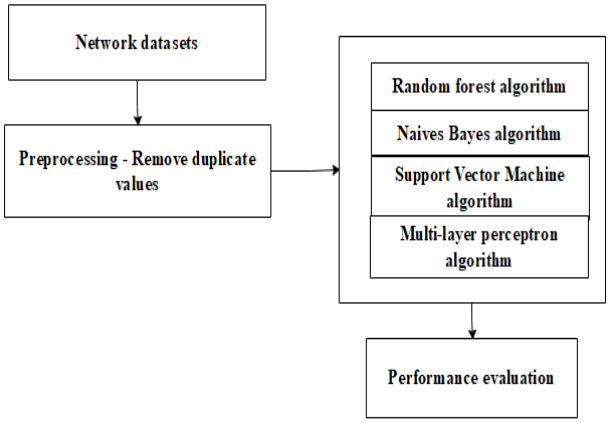


FIG 4: PROPOSED FRAMEWORK

Fig 4 provide framework for the proposed system include steps such as datasets acquisition, preprocessing and classification. In preprocessing, remove duplicate values and perform classification algorithms such as Random forest, SVM, Naives Bayes and MLP algorithm. Finally compare the classification algorithm in terms of error rates in performance evaluation

4.Experimental Results

We can upload the datasets for 1075 records and collect the samples from UCI repository database. And using 23 attributes for predicting traffic in internet. The dataset contains the attributes such as id, node, utilized, packet, full bandwidth, average delay time per second,Percentage_Of_Lost_Pcaket_Rate, Percentage_Of_Lost_Byte_Rate,Packet, of length, lost bandwidth, Packet, Packet_transmitted, Packet_received, 10-Run-AVG-Drop-Rate, 10-Run-AVG-Bandwith-Use, 10-Run-Delay, Node, Flood, class. These attributes can perform classification using tool named as WEKA for WINDOWS OS with any configuration. We can evaluate the performance of each algorithm and compare the performance based on MSE, RMSE, RAE, RRSE and shown in table and performance graph. The classification results can be shown in following figures.

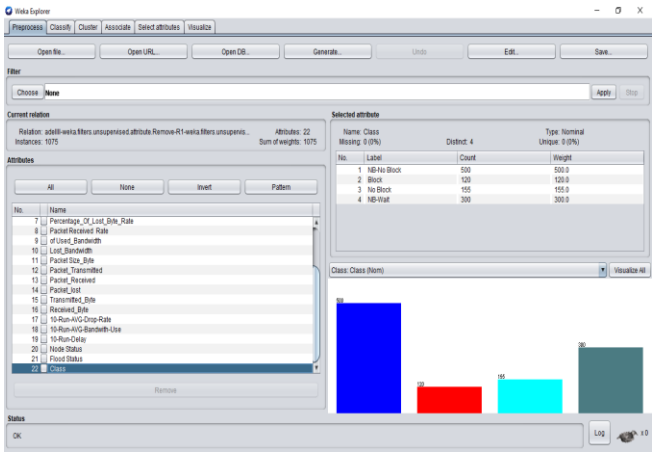


FIG 5: UPLOAD THE DATASET

In this figure, user can upload the datasets with 23 attributes with multiple classes

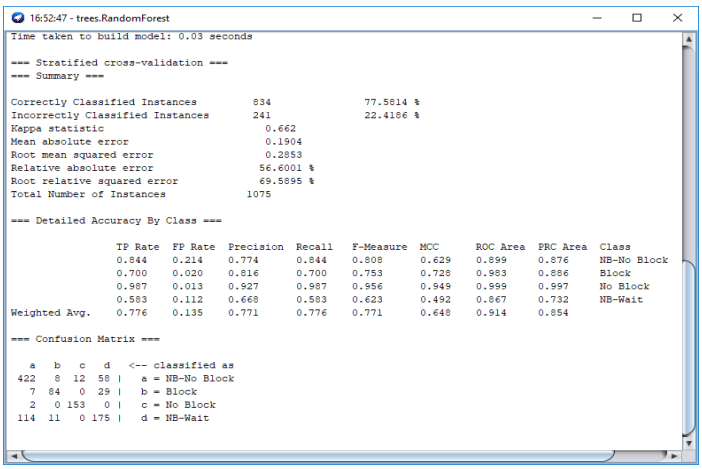


FIG 6: RANDOM FOREST ALGORITHM

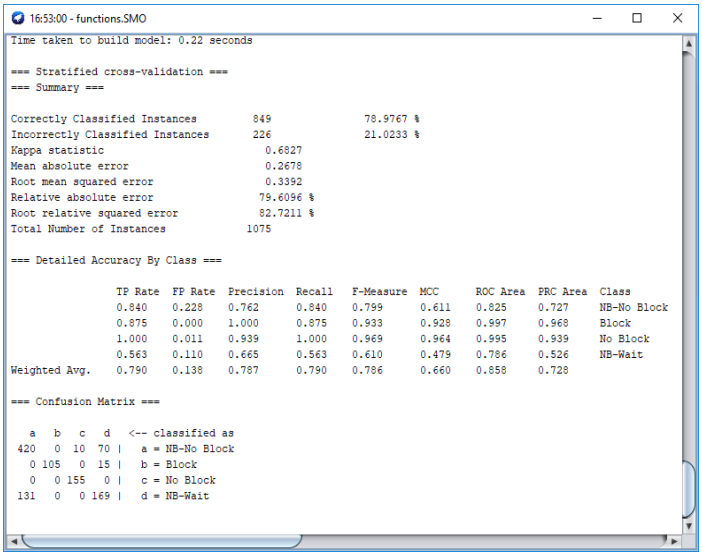


FIG 7: SVM ALGORITHM

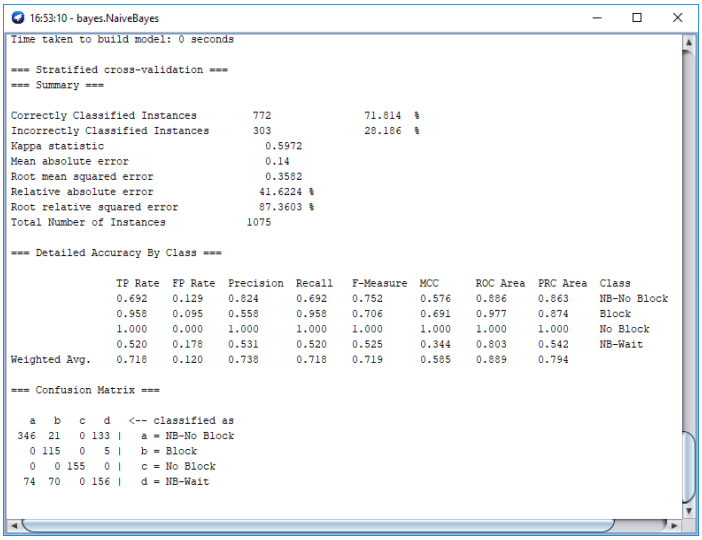
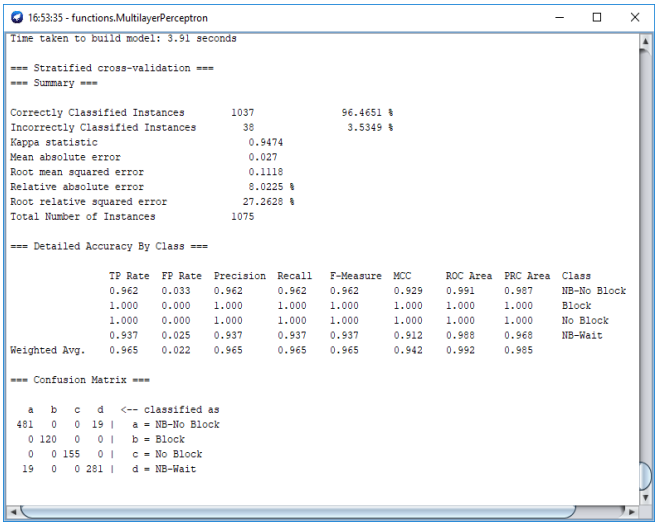


FIG 8: NAIVES BAYES ALGORITHM



```
16:53:35 - functions.MultilayerPerceptron
Time taken to build model: 3.91 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1037               96.4651 %
Incorrectly Classified Instances        38                3.5349 %
Kappa statistic                          0.9474
Mean absolute error                      0.027
Root mean squared error                  0.1118
Relative absolute error                  8.0225 %
Root relative squared error             27.2628 %
Total Number of Instances             1075

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.962    0.033     0.962      0.962     0.962     0.929     0.991     0.987     NB-No Block
                 1.000    0.000     1.000      1.000     1.000     1.000     1.000     1.000     Block
                 1.000    0.000     1.000      1.000     1.000     1.000     1.000     1.000     No Block
                 0.937    0.025     0.937      0.937     0.937     0.912     0.988     0.968     NB-Wait
Weighted Avg.    0.965    0.022     0.965      0.965     0.965     0.942     0.992     0.985

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 481   0   0  19 |   a = NB-No Block
   0 120   0   0 |   b = Block
   0   0 155   0 |   c = No Block
  19   0   0 281 |   d = NB-Wait
```

FIG 9: MLP ALGORITHM

MSE:

Mean Squared Error (MSE) is by far the most common measure of numerical model performance. It is simply the average of the squares of the differences between the predicted and actual values. It is a reasonably good measure of performance, though it could be argued that it overemphasizes the importance of larger errors. Many modeling procedures directly minimize the MSE.

RMSE:

The RMSE serves to aggregate the magnitudes of the errors in predictions into a single measure of predictive power. RMSE is a good measure of accuracy, but only to compare forecasting errors of different models for a particular variable and not between variables, as it is scale-dependent

RAE:

The relative absolute error in some data is the discrepancy between an exact value and some approximation to it. An approximation error can occur because:

1. The measurement of the data is not precise due to the instruments.
2. Approximations are used instead of the real data (e.g., 3.14 instead of π).

In the mathematical field of numerical analysis, the numerical stability of an algorithm indicates how the error is propagated by the algorithm.

RRSE:

The root relative squared error is relative to what it would have been if a simple predictor had been used. More specifically, this simple predictor is just the average of the actual values. Thus, the relative squared error takes the total squared error and normalizes it by dividing by the total squared error of the simple predictor. By taking the square root of the relative squared error one reduces the error to the same dimensions as the quantity being predicted.

TABLE 1: PERFORMANCE TABLE

Algorithms	MSE	RMSE	RAE	RRSE
MLP	0.03	0.11	8.03	27.26
Naives Bayes	0.14	0.35	41.62	87.36

Random forest	0.19	0.29	56.6	66.6
SVM	0.26	0.34	79.6	82.72

The overall performance of the results is shown as graph

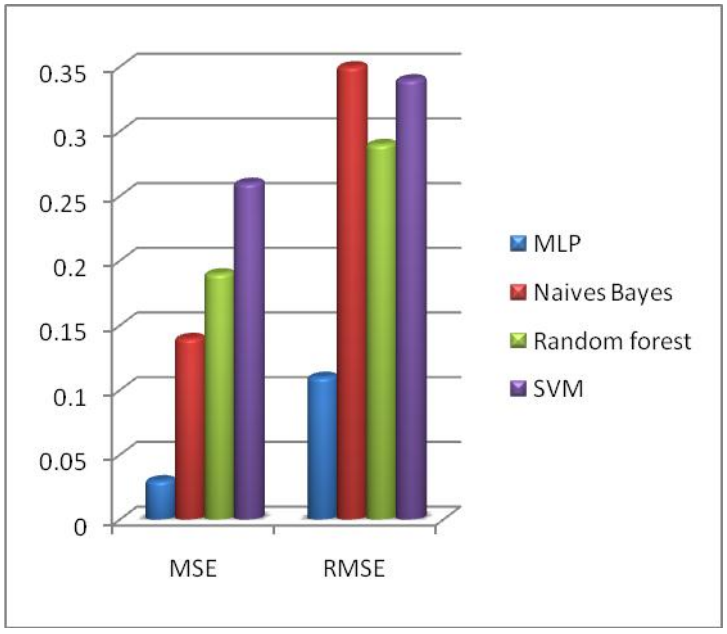


FIG 10 MSE AND RMSE GRAPH

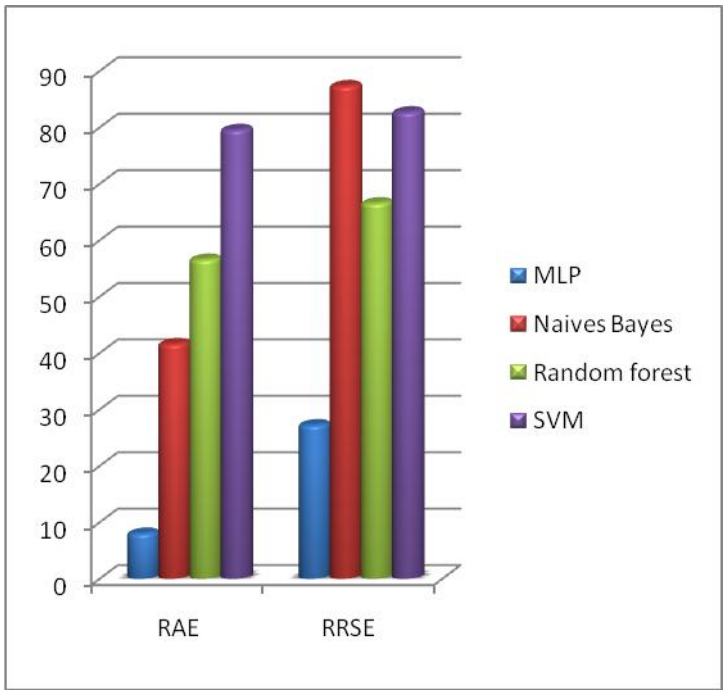


FIG 11 RAE AND RRSE GRAPH

From the above comparison in fig 10 and 11, MLP can be outperforms than the existing algorithms and provides reduce number of error rate values. In this paper, a novel approach based on MLP learning is proposed to classify the network traffic. The proposed used nearly 23 attributes for performance predictions. The experiment displays good performance of the proposed algorithm and was compared to similar approaches over the same dataset.By analyzing the experimental results, it is observed that the MLP algorithm turned out to be best classifier for network traffic performance prediction because it contains more accuracy and least error rate.

5. Conclusion

Traffic classification plays an important role in the network security as the applications and their behavior are changing day to day. Network Traffic classification has extensively researched in recent years and many techniques has been proposed including Flow-Based technique, Host-Based technique and Graph-Based technique. As a result there increased the need for accurate classification of the network flows. Here we have proposed deep learning model using Multi-layer perceptron with feature selection for the accurate classification of internet traffic. In this paper we have demonstrated the construction of a lightweight neural network capable of real-time network traffic classification. In the process, we have also provided greater insight into methodologies used by different classification schemes. We discussed potential procedures for both data processesing and optimization which are generalizable to other supervised machine learning methods. We also outlined a fast method of identifying key attributes in the neural network based on the connection weights. In future we can also use network traffic for cloud computing and Hadoop and big data for security purpose or identification of traffic.

REFERENCES

[1]Namdev, Neeraj, ShikhaAgrawal, and Sanjay Silkari. "Recent advancement in machine learning based internet traffic classification." Procedia Computer Science 60 (2015): 784-791.

[2] Bakhshi, Taimur, and BogdanGhita. "On internet traffic classification: A two-phased machine learning approach." Journal of Computer Networks and Communications 2016 (2016).

[3] Hong, Yang, et al. "Iterative-tuning support vector machine for network traffic classification." 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM).IEEE, 2015.

[4] Ducange, Pietro, et al. "A novel approach for internet traffic classification based on multi-objective evolutionary fuzzy classifiers." 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).IEEE, 2017.

[5] Peng, Lizhi, et al. "Effectiveness of statistical features for early stage internet traffic identification." International Journal of Parallel Programming 44.1 (2016): 181-197.

[6] Kurt, M. N., YÄ´slmaz, Y., & Wang, X. (2019). Real-time detection of hybrid and stealthy cyber-attacks in smart grid. IEEE Transactions on Information Forensics and Security, 14(2), 498-513.

[7] Sabar, N. R., Yi, X., & Song, A. (2018). A Bi-objective Hyper-Heuristic Support Vector Machines for Big Data Cyber-Security. IEEE ACCESS, 6.

[8] Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., & Zhu, M. (2018). HAST-IDS: learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access, 6, 1792-1806.

[9]Aditham, S., &Ranganathan, N. (2018). A system architecture for the detection of insider attacks in big data systems. IEEE Transactions on Dependable and Secure Computing, 15(6), 974-987.

[10] Zhang, T., & Zhu, Q. (2018). Distributed privacy-preserving collaborative intrusion detection systems for VANETs.IEEE Transactions on Signal and Information Processing over Networks, 4(1), 148-161.