Journal of Theoretical and Applied Information Technology
31st October 2016. Vol.92. No.2
© 2005 - 2016 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CLASSIFICATION OF VOIP AND NON-VOIP TRAFFIC USING MACHINE LEARNING APPROACHES

[1]GHAZI AL-NAYMAT, [2]MOUHAMMD AL-KASASSBEH, [3]NOSAIBA ABU-SAMHADANH, [4]SHERIF SAKR

[1] Princess Sumaya University for Technology - Jordan

[2,3]Mutah University, IT Department, Jordan

[4]University of New South Wales, CSE, Australia

E-mail: [1]g.naymat@psut.edu.jo, [2]mouhammd.alkasassbeh@mutah.edu.jo, [3]nosaibahamdan@gmail.com,[4]ssakr@cse.unsw.edu.au

ABSTRACT

Enhancing network services and security can be achieved by performing network traffic classification identifying applications, which is one of the primary components of network operations and management. The traditional transport-layer and port-based classification approaches have some limitations in achieving accurate identification. In this paper, a real test bed is used to collect first-hand traffic dataset from five different VoIP and Non-VoIP applications that are used by majority of Internet community, namely Skype, YouTube, Yahoo Messenger, GTalk and PayPal. The collected data encompasses new features that have never been used before. In addition, a classification step is performed using off-the-shelf machine learning techniques, specifically Random Forest J48, meta.AdaBoost (J48) and MultiLayer Perceptron to classify the traffic. Our experimental results show that using the new features can dramatically improve the true positive ratio by up to 98% and this is significant outcome towards providing accurate traffic classification.

Keywords: *Traffic classification, Application identification, Machine Learning, VOIP and Non-VOIP Application, CAPTCHA*

1. INTRODUCTION

Online services have, recently, become one of the essential needs for the majority of the Internet community. A large number of applications have been developed to fulfill the internet users' needs, such as chat, voice calls, emails, videos, file transfer and online payments, to mention but a few. Internet providers aim to analyze and classify the generated traffic from the online applications. The most prominent applications are Voice over Internet Protocol (VoIP) applications, such as Skype[1], Yahoo Messenger[2], and Google Talk (GTalk)[3]. Additionally, the most prominent Non-VoIP applications include YouTube[4] and PayPal[5]. These applications share some common features, such as user friendly interfaces, improved usability on smart devices such as smart phone and iPads, tablets, etc. In addition, most of them are available in a free of charge basic version. This has motivated us to incorporate them in our study.

VoIP technology allows users to communication with each other over the Internet protocol and that is better than the traditional telephone networks [13]. As for the non-VoIP applications, they are used for watching the preferred videos and many other features available on the YouTube site for example. In addition, people can manage different financial matters online by using PayPal.

The aforementioned examples of data communication and digital media applications generate millions of dollars of revenue for their providers each year [1]. Therefore, traffic classification is one of the most appropriate and essential network monitoring approaches In other words, it is a task that allows Internet Service Providers (ISP) to identify which application is generating the traffic data. Traffic classification leads to the implementation of Quality of Services

[1] http://www.skype.com/en/
[2] https://messenger.yahoo.com/
[3] http://google-talk.software.informer.com/
[4] https://www.youtube.com/
[5] https://www.paypal.com

(QoS) for enforcing Internet users to comply with the internet polices and for intrusion detections.

The rise of new applications and Internet services has made the network traffic very complex and diverse [1]. The security concerns, which are very important nowadays, have motivated researchers to conduct their research and provide advances in the area of traffic classification. However, due to the sensitivity and restrictions on the sharing or making the traffic data available for researchers, there is insufficient reproducible research in the domain of Internet traffic classification. This has also motivated us to generate an unprecedented traffic dataset that collected data from a real test-bed environment designed to capture four different features (Packet Length, Delta Time, Cumulative Byte and Relative Time) to be used, to the best of our knowledge, for the first time in traffic classification.

An open-source packet analyzer called Wireshark[6] is used to capture the traffic data. Then the desired features are selected from the collected data. A pre-processing step is applied on the collected data to prepare and transform it into the desired format. Four off-the-shelf Machine Learning (ML) classification techniques, specifically meta.AdaBoost (J48) [25], Random forest [24], J48 [26] and MultiLayer Perceptron (MLP) [27], are used to classify the traffic data. The aforementioned techniques are used from the well-known ML tool, WEKA [22].

It should be noted that traffic classification is a very important issue for different areas, the most important one is the Intrusion Detection Systems (IDS) as it shown in [30] [31] [32]. Through the traffic type, the IDS decide whether the traffic is normal or abnormal. Another area, is the network management problem, some applications need to have a high percentage of their utilization which means that knowing the type of traffic they deal with in advance would give them the chance to avoid any potential problems in terms of hardware or software [33][34]. Recently, smart proxies use the traffic status to distinguish between normal and unwanted users. For instance, some users use illegal applications to get over the proxies by encapsulating their traffic within the normal ones, which leads to increased bandwidth consumptions and other problems in the networks. This has clearly shown the clear need and importance of conducting traffic classification.

In this paper, we make the following contributions:

1. Introducing four important features to be used for the first time in traffic classification
2. Collecting unprecedented traffic dataset from a real network environment.
3. Evaluating four different classification techniques to classify five VoIP and Non-VoIP applications to help service providers/developers to understand the application behaviors and protect them from any malware or attack.
4. Providing the research community with our finding that the meta.AdaBoost (J48) classifier is showing the best performance among other classifiers.
5. Our experimental results show that using the new features can dramatically improve the true positive ratio up to 98% and this is a respectable outcome towards providing accurate traffic classification.

The rest of this paper is organized as follows: Section 2 discusses the related work in the area of traffic classification. The traffic classification framework and a description of the real test-bed that we have considered in this work are explained in Section 3. Our evaluation metrics and the obtained results are discussed in Section 4. Finally, we conclude our work and we list some of our future work in Section 5.

2. RELATED WORK

Recently, Internet researchers have become interested to invest more efforts in the area of traffic classification due to its importance. Knowledge about traffic structure is beneficial for network planning, security and traffic control [23]. However, conducting this type of research is challenging because of the difficulty in comparing the results and approaches since everyone is using different features and techniques to perform the classification task. In addition, the unavailability of standardized benchmarks makes the comparison extremely complicated. This section provides an overview of the related work that has been conducted by the interested researchers to address this challenge.

[6] https://www.wireshark.org/

Classical traffic classification approaches used the port number, the payload information and the encryption technology. These types of features introduce some new issues, where researchers have started to tackle them, by incorporating supervised and unsupervised machine learning techniques. However, supervised learning methods suffer from the lack of labeled instances to train the classification model. In the unsupervised method, the problem was to define which parameters to involve in the process. Mahajan et al. [15] proposed a new Semi-Supervised ML technique that trains the classifier using a set of training datasets that consists of both labeled and unlabeled instances. The results of their experiments showed that the classifier had its best performance at 30% of the labeled instances. And the highest accuracy reached was 94.7%. Tapaswi et al. [12], proposed another estimator that used Naïve Bayes to classify traffic according to features of peer-to-peer (P2P) networks with the largest volume of bandwidth. They were concerned to classify the network into P2P and non-P2P. Their accuracy reached was between 65%-85%.

The large amount of packets and bandwidth made the process of enabling intrusion detection and network protection extremely hard. Qin et al. [2] proposed a solution for large volume of the packets. This solution discriminates traffic on the basis of flows instead of individual packets. They employed the Bi-Flow model to gather traffic packets with the aim of extracting the features of mutual behavior using the various terminals. They used Poisson sampling to reduce the size of gathered data and ease its handling. The experimental results based on the effects of traffic that is collected from their university platform displayed a high level of accuracy.

Alshammari et al. [3], focused on VoIP applications and used various techniques to identify and classify the encrypted traffic flow for applications in order to generate robust signatures for identifying the encrypted traffic. They used three different ML algorithms namely: AdaBoost, C5.0 and Genetic programming (GP). In addition, they applied statistical calculation on a network flow to extract a set of unique features for each application. Furthermore, they used many types of datasets for training and testing. Their finding shows that C5.0 algorithm is the best according their published results. Alshammari et al. [6] incorporated the modern machine learning techniques with a set of statistical and simple packet header feature sets to describe the encrypted application tunnels in the network traffic. Two encrypted applications are used in their study: Skype and Secure Shell (SSH) using various traces form different networks. The results of their experiments showed that it is possible to identify tunnels of the encrypted traffic with high accuracy. In addition, it is possible to identify the services that run in the encrypted tunnels.

A VoIP application, such as Skype is one of the most used applications. However, Skype uses different encryption mechanisms with a proprietary design with closed source; making it very difficult to analyze its traffic. Due to the above reason in [19], they concentrated only on minimizing the cost of the algorithm in detecting Skype traffic. Fonseca et al. [8], presented a study on the techniques used in detecting the traffic in VoIP applications. They focused on profiling the network traffic patterns and on modeling the communication flows for anomaly detection. This work confirmed that the legacy approach of monitoring that depends on the port number and protocol has become less accurate than those modern techniques.

Nearest Neighbor (NN) is a modern machine learning technique that always shows great performance, because there is no need to conduct any training. It is able to deal with a large number of classes and there is no overfitting risk. Zhang et al. [11] proposed a new non-parametric approach to improve the performance of the NN algorithm in traffic classification. They considered the experimental and theoretical aspects to analyze this information that links their performing information to each other.

Ibrahim et al. [13], discussed the process of classifying two interactive applications namely: Skype and online TV. These interactive applications have been recently widely used and have gained wide importance. In their work, they used the Wireshark tool to capture the packets that is transmitted over the network. They only considered two features: interval time and packet length. After collecting the data, they applied ten ML classification algorithms and they found that the Random forest algorithm provides the highest accuracy for their datasets. Adami et al. [18] presented a real-time algorithm called Skype-Hunter to classify and detect the Skype traffic. This algorithm uses the signature-based and statistical procedures that are used in classifying data traffic signals, data traffic of calls and data transfer. This algorithm was applied on many datasets that are collected from different network scenarios. The system outperformed the classical statistical ways used for traffic classification. The analysis of the results shows that their algorithm has very good performance for the different types of traffic traces in different network access.

Wicaksana et al. [20] presented a fast architecture and reconfigurable Packet Classification Engine (PCE). The engine used in the firewall is based on the FPGA that depends on tree algorithm. Additionally, it also inspects multi-dimensional fields of the packet header. This approach is based on features, such as Source Port, Source IP Address, destination IP Address, destination Port and Protocol fields of the packet header. The PCE examined the Ethernet packet to know which of these packets is normal and which is potentially dangerous before investigating its content. This technique has shown the importance of filtering and classifying the Ethernet packets within network devices for intrusion detection.

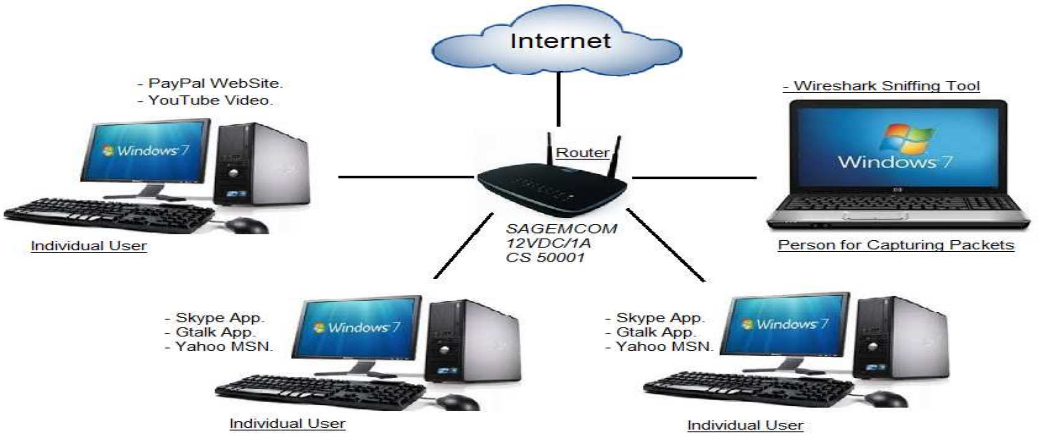

Figure 1: Testbed for Real Network.

3. TRAFFIC CLASSIFICATION FRAMEWORK

The designed test-bed used in this research is described in this section and shown in Figure 1. In addition, the section sheds light on the proposed framework (Figure 2). It consists of two major steps, data generation and traffic classification.

3.1. Test bed specifications

The dataset has been collected from a real test-bed on a network containing four PCs. One PC uses the Wireshark tool to capture the traffic crossing over the network from the examined applications. The remaining PCs communicate together using VoIP and Non-VoIP applications as shown in Figure 1. All used PCs run on Windows 7, Intel(R) Core(TM)2 Duo CPU T6500 @ 2.10GHz 2.0 GB RAM computers. The Internet speed used was 7 Mbps for downloading and 0.91 Mbps for uploading. The Wireshark sniffing tool has been used to capture the real time traffic for the applications namely, Gtalk, PayPal, Yahoo MSN, Skype and YouTube. These applications are different in terms of the requirements and the place in the network layers. Some of them need a secure link, some need a more reliable connection and others need the speed as the main priority. These needs reflect the normal users' daily needs.

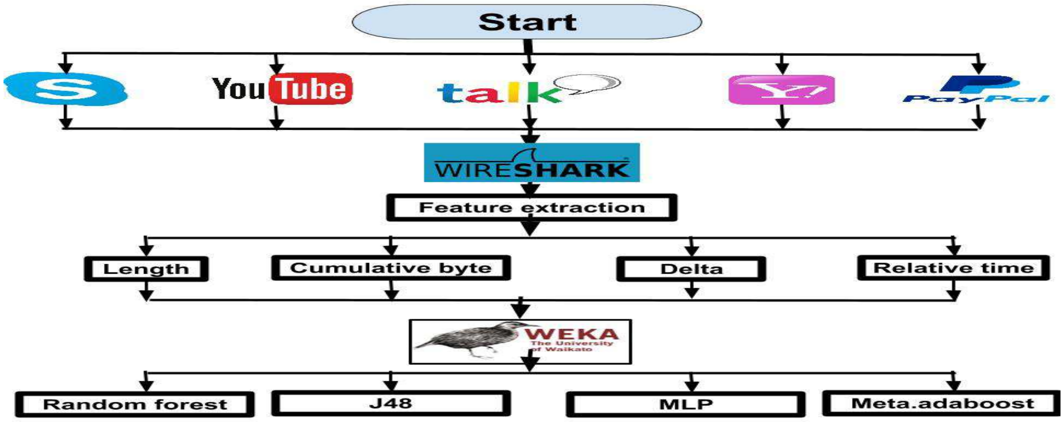

Figure 2: Data Generation And Classification Framework.

3.2. Dataset Generation and Classification Framework

Figure 2 shows the traffic classification framework that shows the major phases namely, data generation and traffic classification. Firstly, a Wireshark tool was used to capture the traffic for each of the aforementioned VoIP and Non-VoIP applications. The second phase, a preprocessing task was performed on the collected data to clean it and convert it into the required format, which was used with the ML tool (WEKA). Next, the needed features were selected out of the data. These features are Packet Length (P_L), Cumulative Byte (C_B), Delta time (D_t) and Relative time (R_t).

$\boldsymbol{P_L}$ is one of the important features that has been used in traffic classification processing for a while now. It shows the length of each packet that crosses the real network; the length is controlled by the hardware and the software that the network is using. Some protocols use fixed size packets and others do not. For the dynamic size it has minimum and the maximum length, each application has its own length or deals with a range of lengths. The network layer is responsible for the packets and assures the packet is transported from the source point to its final destination. For instance, on the Ethernet network the original size of the transmitted user data is between 46 and 1500 byte.

$\boldsymbol{D_t}$ is called the inter-arrival time that is the calculated time between the arrivals of two successive packets. In other words, it is the time since the previous packet was arrived or captured. $\boldsymbol{D_t}$ is used to measure network roundtrip and server response time as well as other delays. The formula for $\boldsymbol{D_t}$ is given in Equation (1):

$$D_t = P_{t1} - P_{t0} \qquad (1)$$

where $\boldsymbol{P_{t1}}$ is the arrival time for the next packet, and $\boldsymbol{P_{t0}}$ is the arrival time for the previous packet.
$\boldsymbol{C_B}$ is the cumulative byte that shows the amount of data that can be transmitted between the sender and receiver when a large block of data crosses over the network. It is considered the scale that measures the total bytes that are transmitted in the time interval from the captured traffic. $\boldsymbol{C_B}$ can be calculated using Equation (2):

$$C_{B1} = C_{B0} + P_{L1} \qquad (2)$$

where C_{B0} is the previous cumulative byte and P_{L1} is the current packet length.

The Relative time $(\boldsymbol{R_t})$ is the elapsed time between the first packet and the current packet, sometimes it is called the cumulative time. It is the total captured time from the beginning of the captured process to the last packet that stopped.

$$R_{t1} = R_{t0} + D_{t1} \qquad (3)$$

where $\boldsymbol{R_{t0}}$ is previous relative time and $\boldsymbol{D_{t1}}$ is current delete time.

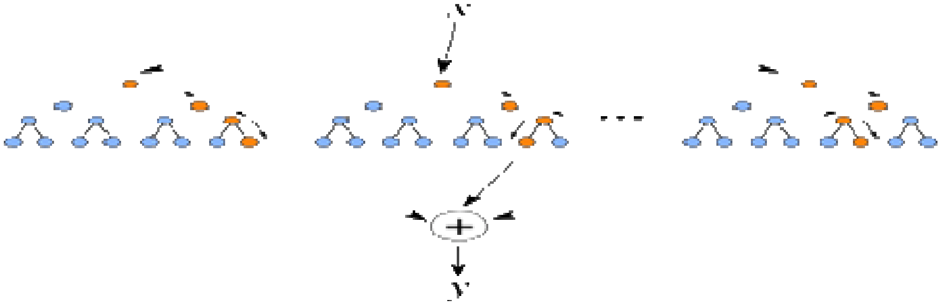

Figure 3: Random Forest Structure

In this work, we adopted the statistical classification which is based on the nature of the services that are available on the web (Non-VoIP vs VoIP). Most classifiers investigate the traffic measurements (features) to characterize the traffic of different services. A comprehensive list of a large number of possible traffic discriminators can be found in [35]. In this work, we study the most common and effective features that reflect the type of traffic to help the classifiers achieving best results with the minimum number of features.

After the feature selection step, the data is entered into the ML tool (WEKA) and four well-known classifiers were used to classify the collected traffic to its application. The next subsection provides a quick introduction about these four classifiers.

3.3. Machine Learning Classifiers

Machine Learning classifiers are used to classify the network traffic into the application that is generating it. The goal is to build a model from classified objects and use the model to classify new objects as accurately as possible. Our framework is illustrated by applying these four famous ML classifiers on the generated datasets. These classifiers are supervised learning algorithms that use labelled training data. The following subsections give more details about the classifiers used.

3.3.1. meta.AdaBoost (J48) Classifier

The Adaptive Boosting (AdaBoost) algorithm was proposed by Yoav Freund and Robert Schapire [25]. It is an important ensemble-based classifier. The idea behind AdaBoost, is that it can be combined with other classifiers (weak learners) to enhance their accuracy and performance. AdaBoost starts with a base classifier built on the training data, it assigns equal weights to all samples of training data, according to the performance of the classifier, and the weight of each sample of training data is then modified. Another classifier is then established to concentrate on the examples from the training data that were obtained incorrectly from the base classifier. The process of adding further classifiers is repeated until the number of models or accuracy reaches a specified desired value, then the output of these classifiers (weak learners) is combined into a weighted sum that represents the final model of the boosted classifier. The AdaBoost algorithm can attain a very accurate rate of prediction. It has great simplicity; therefore it has been applied widely and successfully. Many empirical studies have shown that AdaBoost is affected less by the overfitting problem than other learning algorithms and it is presented as one of the top 10 algorithms in the research community. AdaBoost uses the decision tree for the base classifier. Another version of the AdaBoost algorithm is called the AdaBoostM1 algorithm [25] as presented by Freund and Schapire, in order to deal with multi-class problems. In this paper, we used the boosting process to improve the performance of J48, which is a decision tree ML algorithm. More details about this algorithm will be given in Section 3.3.3. Essentially, this means that the AdaBoost algorithm we have considered in our work is the meta.AdaBoost (J48) algorithm.

3.3.2. Random Forest Classifier

Random Forest was proposed by Leo Breiman [28] as an ensemble learning method for classification or regression. It uses a tree classification algorithm to generate a large number of decision trees where each tree is built by a different bootstrap of samples from the original data by using random feature selection in the tree induction process, so that the final Random Forest model is a classifier in the form of many individual decision trees. Thus, in ensemble terms, the decision trees are weak learners and the random forest is a strong learner. Figure 3 shows an example of random forest architecture. After constructing the forest model, any new object to be classified is entered in every tree in the forest. Each decision tree gives a vote about the class of the object, and finally the random forest chooses the class with the majority votes for the object. The Random Forest algorithm is considered to be an accurate classification method, more robust to noise and is capable of handling multiple inputs and missing values.

3.3.3. J48 Classifier

J48 classifier was developed by Ross Quinlan [26]. It is the implementation of a decision tree classifier called C4.5, which is a binary decision tree based classification algorithm. It is a top-down induction of decision trees and uses the key concepts of information theory to know which attribute to select. Users can easily understand its tree. Also it stops the split process when the number of nodes becomes very small (i.e. the default value is two nodes).

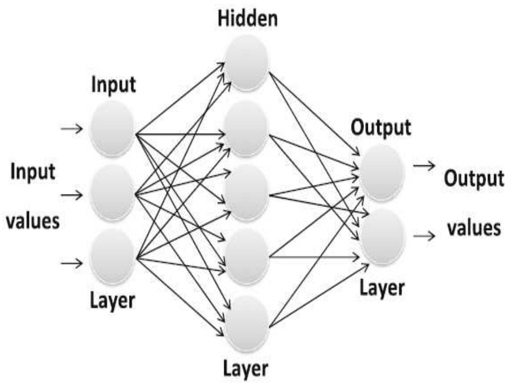

Figure 4: MultiLayer Perceptron (MLP) sturcture

3.3.4. MultiLayer Perceptron Classifier

MultiLayer Perceptron (MLP) is a feedforward artificial neural network (ANN) algorithm. MLP is the most common model and widely used class of ANN. It consists of one input layer, one output layer, and one or more hidden layers (Figure 4). MLP aims to create a relationship that maps a set of inputs into a set of suitable outputs, then the MLP model can be used to extract unknown outputs [29]. In addition to that, it is a modified version to the standard linear perceptron and can discriminate the data that are not linearly separable. In non-linear activation function each node called a neuron or processing element. Each neuron has a value that is calculated from weighted values of its previous input neurons and summed up with inputs values, individual for each neuron, plus the bias term.

4. EVALUATION AND RESULTS

4.1. Evaluation Metrics

In this paper, we used well-known evaluation criteria to measure the classifiers performance, such as accuracy, precision and recall. The basic performance is indicated by the confusion matrix (Table 1). The confusion matrix contains the numbers about actual and predicted class of the model used.

Table 1 Confusion Matrix for two classes.

	Predicted Class		
Actual Class		Positive	Negative
	Positive	TP	FP
	Negative	FN	TN

The confusion matrix consists of four rates used to evaluate the performance of the classification model used: True Positive (TP) refers to the correct prediction rate of the positive traffic instances. False Positive (FP) refers to the ratio of negative traffics that were incorrectly classified as positive. True Negative (TN) refers to the ratio of negative traffic instances that were correctly classified as negative. False Negative (FN) refers to the ratio of positive instances that were incorrectly classified as negative. The below metrics used the ratios form the confusion matrix to evaluate the model used.

Table 2 Confusion Matrices For The Used ML Algorithms.

meta.AdaBoost(J48)						Randomforest					
	PayPal	Gtalk	Yahoo MSN	Skype	YouTube		PayPal	Gtalk	Yahoo MSN	Skype	YouTube
PayPal	959	5	0	34	2	PayPal	954	10	0	34	2
Gtalk	6	983	3	8	0	Gtalk	7	985	0	7	1
Yahoo MSN	0	3	995	1	0	Yahoo MSN	0	5	993	1	0
Skype	14	6	1	988	0	Skype	20	10	1	978	0
YouTube	0	0	0	2	992	YouTube	3	0	0	0	991
J48						**MLP**					
	PayPal	Gtalk	Yahoo MSN	Skype	YouTube		PayPal	Gtalk	Yahoo MSN	Skype	YouTube
PayPal	932	17	0	48	3	PayPal	655	52	0	290	3
Gtalk	18	960	7	14	1	Gtalk	6	818	64	112	0
Yahoo MSN	2	5	992	0	0	Yahoo MSN	0	29	967	3	0
Skype	32	18	0	959	0	Skype	65	156	0	783	5
YouTube	2	0	0	0	992	YouTube	1	3	0	3	987

Accuracy: measures the rate of the correctly classified traffic instances of all applications.

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FN + TN} \qquad (4)$$

Precision: It represents the ratio of the number of relevant traffic instances retrieved to the total number of irrelevant and relevant retrieved. It is also called positive predictive, which can be calculated by the following equation.

$$\text{Precision} = \frac{TP}{TP + FP} \qquad (5)$$

Recall: It represents the ratio of the number of relevant traffic instances retrieved to the total number of relevant traffic instances. It is also called positive sensitivity value, which can be calculated by the following equation.

$$\text{Recall} = \frac{TP}{TP + FN} \qquad (6)$$

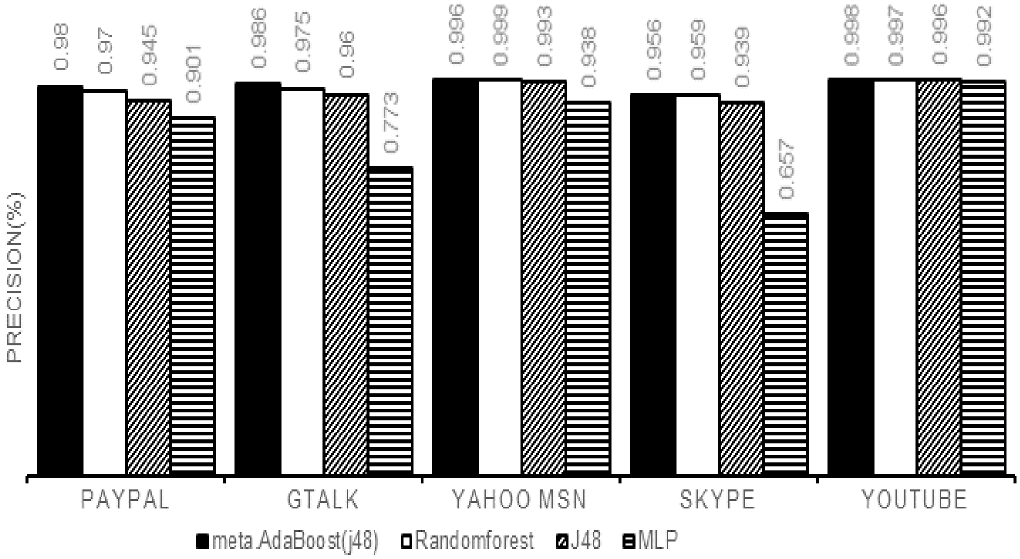

Figure 5: Precision Results.

4.2. Results and Discussion

The evaluation metrics described in the previous section are used to measure the performance of the ML classifiers that were tested on our collected dataset. The confusion matrices for the meta.AdaBoost (J48), Random Forest, J48 and MLP classifiers are shown in Table 2. To display the best results, the diagonal of the matrices must have the highest values than other upper and lower values. Therefore, the results show that the meta.AdaBoost (J48) classifier demonstrate the highest values on the diagonal than the other classifiers. All the matrices are allocated in one table to make it easier for the reader to make quick and easy comparisons.

Table 2 reveals clearly that the maximum accuracy achieved by the meta.AdaBoost (J48) classifier is 98.3007%. It also shows that Random Forest and MLP classifiers achieved 96.661%, 84.166%, respectively. Hence, the MLP classifier showed a poor performance in comparison to other classifiers.

As been mentioned in the evaluation metrics section, we considered the Precision and Recall metrics to demonstrate the performance of the ML classifiers. The precision expresses the percentage of the predicted traffic that is correctly classified. Figure 5 depicts the Precision rates for all ML classifiers for all tested applications. It is clear that the classifiers managed to retrieve high precision rates in all applications except that MLP showed the lowest precision rate for GTalk and Skype traffic. We also measured the Recall rate to consider the sensitivity of the ML classifiers. Figure 6 illustrates the Recall rates that are closer to the Precision rates; this indicates that ML classifiers have shown high sensitivity and specificity. Recall rates also showed that MLP classifier's performance was not as good as other classifiers when classifying Skype and PayPal.

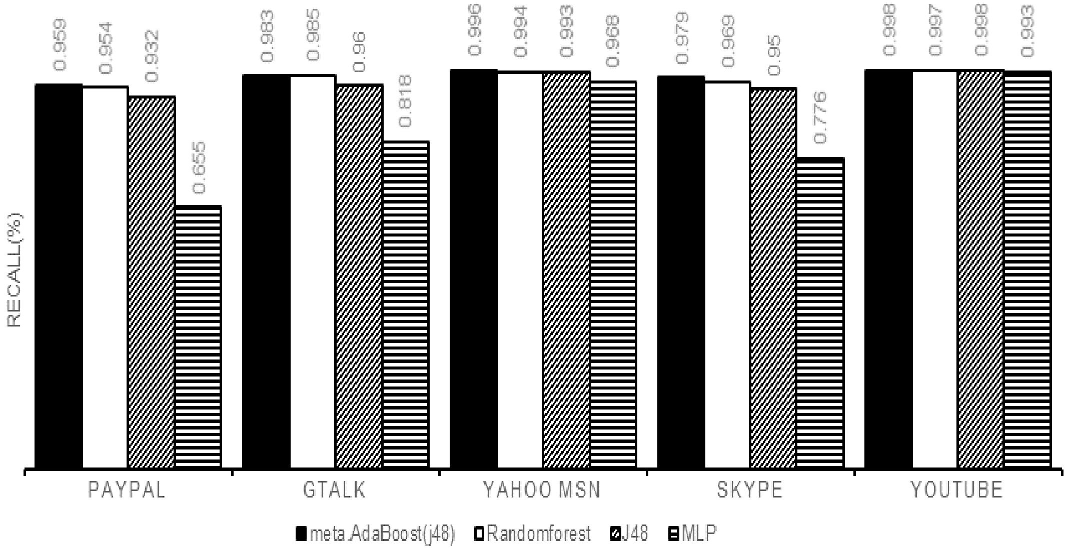

Figure 6: Recall Results.

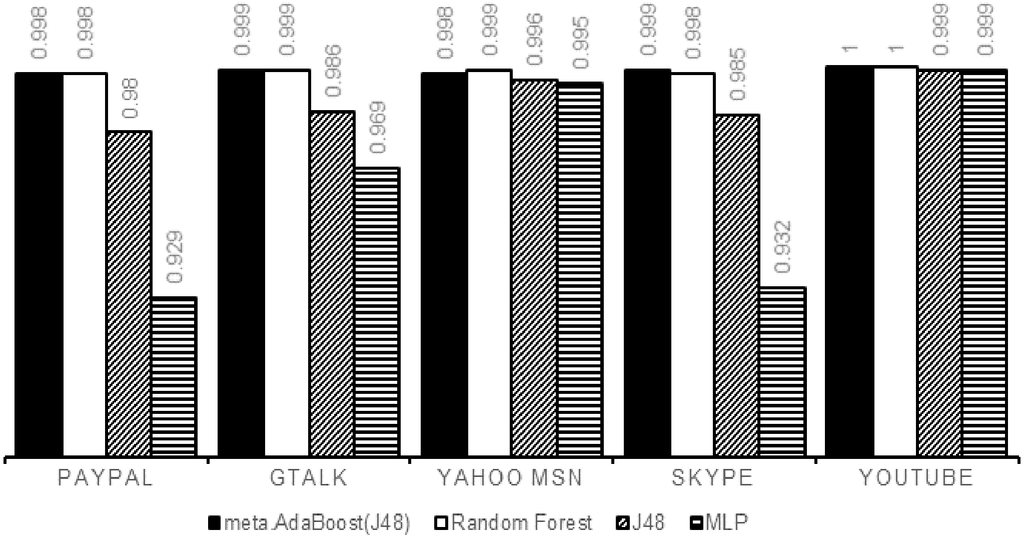

Figure 7: The Area Under Roc For The Ml Algorithms

To show the tradeoff between sensitivity (True Positive rate) and specificity (False Positive rate), we considered the Receiver Operating Characteristic (ROC) as shown in Figure 7. We represented the ROC curve as a column chart since we have very close performance for our tested classifiers. Figure 7 demonstrates and confirms the Precision and Recall results that have been shown in Figures 5 and 6. Our tested ML classifiers demonstrate high accuracy (almost 100%) in classifying the YouTube traffic as shown in Figure 7. The reason behind this high accuracy is that YouTube traffic data (features) has a great consistency; this means that our training dataset has high uniformity with the testing dataset. As an overall result, it should be noted that meta.AdaBoost (J48) showed the highest accuracy in classifying all types of traffic generated form all applications. However, the MLP classifier showed, for all test cases, poor accuracy in comparison to other classifiers. This is, to the best our knowledge, due to the nature of the MLP classifier; as it needs a longer training time as well as the complexity that comes form the number of parameters that need to be set very carefully.

5. CONCLUSION AND FUTURE WORK

In this paper, we collected some unprecedented traffic dataset from an actual network environment. The dataset consisted of four important features, which were used for the first time in traffic classification. We evaluated four different classification techniques to classify five VoIP and Non-VoIP applications to help service providers/developers to understand the applications' behavior and protect them from any malware or attack. We provided the researchers with our finding that the meta.AdaBoost (J48) classifier shows the best performance among other classifiers. Our experimental results showed that using new features we managed to improve the true positive ratio up to 98%. Our future studies will concentrate on examining other features to increase the accuracy. We will also study the accurate and detailed classification methods for the studied traffic by considering other ML techniques.

REFRENCES:

[1] Ham, Jae-Hyun, Hyun-Min An, and Myung-Sup Kim. "Application Traffic Classification using PSS Signature." KSII Transactions on Internet and Information Systems (TIIS) 8.7 (2014): 2261-2280.

[2] Qin, Tao, et al. "Robust application identification methods for P2P and VoIP traffic classification in backbone networks." Knowledge-Based Systems 82 (2015): 152-162.

[3] Alshammari, Riyad, and A. Nur Zincir-Heywood. "Identification of VoIP encrypted traffic using a machine learning approach." Journal of King Saud University-Computer and Information Sciences 27.1 (2015): 77-92.

[4] Shao, Yiyang, et al. "Towards time-varying classification based on traffic pattern." Communications and Network Security (CNS), 2014 IEEE Conference on. IEEE, 2014.

[5] Sinam, Tejmani, et al. "A technique for classification of VoIP flows in UDP media streams using VoIP signalling traffic." Advance Computing Conference (IACC), 2014 IEEE International. IEEE, 2014.

[6] Alshammari, Riyad, and A. Nur Zincir-Heywood. "Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?."Computer networks 55.6 (2011): 1326-1350.

[7] Masud, Mohammad M., Umniya Mustafa, and Zouheir Trabelsi. "A data driven firewall for faster packet filtering." Communications and Networking (ComNet), 2014 International Conference on. IEEE, 2014.

[8] Fonseca, Hugo, et al. "A comparison of classification techniques for detection of VoIP traffic." Next Generation Mobile Apps, Services and Technologies (NGMAST), 2014 Eighth International Conference on. IEEE, 2014.

[9] Duan, Qi, and Ehab Al-Shaer. "Traffic-aware dynamic firewall policy management: techniques and applications." Communications Magazine, IEEE51.7 (2013): 73-79.

[10] Xue, Yibo, Dawei Wang, and Luoshi Zhang. "Traffic classification: Issues and challenges." Computing, Networking and Communications (ICNC), 2013 International Conference on. IEEE, 2013.

[11] Zhang, Jun, et al. "Network traffic classification using correlation information."Parallel and Distributed Systems, IEEE Transactions on 24.1 (2013): 104-117.

[12] Tapaswi, Shashikala, and Ananya Sen Gupta. "Flow-based P2P network traffic classification using machine learning." Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2013 International Conference on. IEEE, 2013.

[13] Ibrahim, Hamza Awad Hamza, et al. "Taxonomy of machine learning algorithms to classify real time interactive applications." International Journal of Computer Networks and Wireless Communications 2.1 (2012): 2012.

[14] Dainotti, Alberto, Antonio Pescape, and Kimberly C. Claffy. "Issues and future directions in traffic classification." Network, IEEE 26.1 (2012): 35-40.

[15] Mahajan, Vinod Shantaram, and Brijesh Verma. "Implementation of network traffic classifier using semi supervised machine learning approach."Engineering (NUiCONE), 2012 Nirma University International Conference on. IEEE, 2012.

[16] Zhao, Shupeng, et al. "A novel online traffic classification method based on few packets." Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on. IEEE, 2012.

[17] Bujlow, Tomasz, Tahir Riaz, and Jens Myrup Pedersen. "A method for classification of network traffic based on C5. 0 Machine

Learning Algorithm."Computing, Networking and Communications (ICNC), 2012 International Conference on. IEEE, 2012.

[18] Adami, Davide, et al. "SkypeHunter: A realtime system for the detection and classification of Skype traffic." International Journal of Communication Systems 25.3 (2012): 386-403.

[19] Del Río, PM Santiago, et al. "On the processing time for detection of Skype traffic." Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International. IEEE, 2011.

[20] Wicaksana, Arya, and Arif Sasongko. "Fast and reconfigurable packet classification engine in FPGA-based firewall." Electrical Engineering and Informatics (ICEEI), 2011 International Conference on. IEEE, 2011.

[21] Abderrahim, Hamza, Mohammed Reda Chellali, and Ahmed Hamou. "Forecasting PM10 in Algiers: efficacy of multilayer perceptron networks."Environmental Science and Pollution Research (2015): 1-8.

[22] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1. 10-18.

[23] Zeng, Yi, and Thomas M. Chen. "Classification of traffic flows into qos classes by unsupervised learning and knn clustering." KSII Transactions on Internet and Information Systems (TIIS) 3.2 (2009): 134-146.

[24] Ho, Tin Kam. "Random decision forests." Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on. Vol. 1. IEEE, 1995.

[25] Yoav Freund, Robert E Schapire, A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, Journal of Computer and System Sciences, Volume 55, Issue 1, August 1997, Pages 119-139, http://dx.doi.org/10.1006/jcss.1997.1504.

[26] Ross Quinlan. 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[27] Rosenblatt, Frank. Principles of Neurodynamics: Perceptrons and The Theory of Brain Mechanisms. No. VG-1196-G-8. CORNELL AERONAUTICAL LAB INC BUFFALO NY, 1961.

[28] Breiman, Leo (2001). "Random Forests". Machine Learning 45 (1): 5–32. doi:10.1023/A:1010933404324.

[29] Simon Haykin. 1998. Neural Networks: A Comprehensive Foundation (2nd ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.

[30] Snort - The de facto standard for intrusion detection/prevention, http://www.snort.org, as of January 28, 2016.

[31] Bro intrusion detection system - Bro overview, http://bro-ids.org, as of January 28, 2016.

[32] V. Paxson, “Bro: A system for detecting network intruders in real-time,” Computer Networks, no. 31(23-24), pp. 2435–2463, 1999.

[33] Wang, Y., Xiang, Y., Zhou, W., & Yu, S. (2012). Generating regular expression signatures for network traffic classification in trusted network management. Journal of Network and Computer Applications, 35(3), 992-1000

[34] Roughan, M., Sen, S., Spatscheck, O., & Duffield, N. (2004, October). Class-of-service mapping for QoS: a statistical signature-based approach to IP traffic classification. In Proceedings of the 4th ACM SIGCOMM conference on Internet measurement (pp. 135-148). ACM.

[35] Moore, A., Zuev, D., & Crogan, M. (2005). Discriminators for use in flow-based classification. Queen Mary and Westfield College, Department of Computer Science.