http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet

Asian Network for Scientific Information

308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (9): 1746-1754, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.1746.1754

A Skype Ml Datasets Validation and Detection Mechanism using Snort Rules and Statistical Approaches

Hamza Awad Hamza Ibrahim, Sulaiman Mohd Nor and Izzeldin Ibrahim Mohamed Abdelaziz
Faculty of Electrical Engineering, Universiti Teknologi Malaysia-UTM, Malaysia

Abstract: Internet traffic classification is an area of current research interest. Identification of real time applications such as Skype has gained more attention in the last few years. Skype traffic classification is challenging because Skype uses encrypted traffic and uses no well-known port number. Several methods which used both signature-based and statistical approaches were proposed. However, the training and testing datasets validation have not been formally addressed. This study highlights the problem of Machine Learning (ML) datasets validation and proposes a mechanism based on Snort rules and ML statistical approach to identify Skype traffic. Two different networks environment are considered for Skype traffic to gain insight into the statistical features of Skype traffic. Six Snort rules based on Skype login were proposed to generate training datasets as inputs to ML. Four algorithms within Weka are used to examine the best algorithm for the given datasets. J48 was found to be the best resulting in more than 99% classification True Positive (TP).

Key words: Traffic classification, machine learning, snort rules, skype, ML algorithms

INTRODUCTION

Over the last few years Skype has gained significant attention and has become one of the most popular forms of VoIP software. According to the Skype website [http://www.skype.com/en/], Skype users in the last year spent 1.8 billion h making video calls. Also, at certain times, more than 22 million users were logged onto Skype at the same time. Skype is easy to use and provides a wide range of services such as voice and video calls, data transfer, video conference, instant message, online number, sharing screen etc.

Skype consists of several elements which are responsible for providing the connection between the two communication parts. Skype Client (SC) is a term for the machine and software which runs the Skype application. This includes computers and smart phones. The second element is Super Node (SN) which is a node with public address and adequate specification (CPU, RAM, etc.,). SNs establish networks among themselves, while SC tries to select an SN. Another two elements are Skype Login Server (LS) and Skype Update Server. The first is responsible for authentication checking, the second make checks to update users' versions with each login. More explanations and details of how Skype elements communicate can be found by Adami *et al*. (2012), Price (2012), Baset and Schulzrinne (2006) and Zhang *et al*. (2010a).

Internet Server Provider (ISP) and network operators are usually interested to know the traffic carried in their networks for the purposes of optimising network performance and security issues. Therefore, Internet traffic classification is something important, particularly interactive traffic such as Skype. Based on a review of the literature, we can divide Skype classification methods into three groups namely (1) ML methods (Jesudasan *et al*., 2010; Ibrahim *et al*., 2012; Jun *et al*., 2007; Branch *et al*., 2009; Alshammari and Zincir-Heywood, 2010; Angevine and Zincir-Heywood, 2008; Zhang *et al*., 2010a; Alshammari and Zincir-Heywood, 2009) which calculate flow or packets statistical features by using ML algorithms (2) Algorithms methods (Adami *et al*., 2012; Zhang *et al*., 2010b; Weirong and Gokhale, 2010; Chen *et al*., 2006; Adami *et al*., 2009) which develop or update an algorithm or model depending on features collected from Skype login or connection analysis and (3) Mixed methods (Freire *et al*., 2008) which combine ML statistical values, payload signature and header information (port number). The evaluation metrics depend on the different methods that were used, for example, ML works can be evaluated by True Positive (TP), True Negative (TN), False positive (FP), False Negative (FN), Precision or Recall. Another evaluation method is to consider standard Skype traces (ready data sets) (TNG, 2008) as testing data and test the classifier as to

Corresponding Author: Hamza Awad Hamza Ibrahim, Faculty of Electrical Engineering, Universiti Teknologi Malaysia-UTM, Malaysia

whether the classifier can classify all the Skype packets in the test data as Skype traffic. An additional mechanism to validate Skype classification methods is to compare the results with commercial classifier results (such as Packet Shaper/Packeteer).

The first objective of this study is to discuss the validity of using training and testing datasets for ML Skype classification. This we do by answering the question, are the statistical features of Skype traffic the same or different in different network environment. The second objective of this study is to describe a mechanism to collect valid Skype dataset for ML Skype classification. Our final objective is to describe our proposed ML Skype classification mechanism and discuss the preliminary results based on only Skype login information for training and testing. In this paper, we consider full Skype session (calls) datasets when comparing statistical features of different network environments. However, when we classify Skype traffic using ML classification, only Skype login datasets collected using Snort rules are considered. To the best of our knowledge, this is the first study that combines Snorts rules and statistical features to classify Skype.

Several mechanisms and methods were proposed to detect Skype traffic but none of these succeeds correctly in classifying all the Skype traffic cases (Adami *et al*., 2012). Identifying Skype traffic is not always easy for the following reasons:

- Skype is a P2P network, so each user can act as a client or server for other users
- There is no well-known port number for Skype
- Skype has a non-fixed protocol
- Skype uses encrypted payload
- Skype continuously releases new software versions
- The values of the statistical traffic are different depending on Skype services (voice/video/data) (Bonfiglio *et al*., 2007) and version (Adami *et al*., 2009)
- The communication between two end SCs includes other channels in between (SNs), which pose some difficulty on Skype classification

ML DATASETS VALIDATION

The problem encountered in the machine learning Internet traffic classification is the validation of training and testing datasets. Normally, the datasets criterion's is assumed to be similar to the real network environment. The challenge in Skype classification is the difficulty to ensure the similarity of training traffic characteristics (packets/flows features) and the traffic to be tested which can only be guaranteed if both are taken from a single machine with the same network environment. Thus there is likelihood that the traffic features values will differ depending on the network factor. This may imply that ML classification for Skype will be only be accurate when all the training and testing datasets are collected in real time from the same network environment.

According to Nguyen and Armitage (2006, 2008), many Internet applications change their statistical properties over time. Alshammari and Zincir-Heywood (2011a) did a good comparison between classification accuracies when Skype datasets were collected from different networks as well as over different years. The training dataset (Univ07) used was collected in 2007 from a university in Canada. Three testing datasets were considered. The first is where the training and testing datasets were from the same network. The second testing dataset is from the same network as training datasets but for different year (Univ10). The last group is from different country (Italy). The results show that the Detection Rate (DR) is high and False Positive (FP) is low when the training and testing datasets comes from the same network and at the same time. The requirement of using real data (traffic packets) is essential in ML classification. We proposed six Skype Snort rules based on login information to prepare Skype training datasets to be used in ML classification. We look to use valid Skype ML datasets and then use a statistical approach to distinguish between Skype and non-Skype traffic (Skype classification). Another motivation is to identify features that are able to classify Skype traffic in specific networks scenarios. In general, Internet traffic classification can help to increase security issues and decrease malicious users (Ibrahim *et al*., 2012; Alshammari and Zincir-Heywood, 2011b). In particular, when Skype traffic is identified, real characteristics of Skype can be defined. This identification also helps campuses and organizations to manage their internet traffic and also reveal applications which use non-well port number to hide themselves. Operators are usually interested to know the traffic carried by their networks for the purposes of optimising network performance (Molnar and Perenyi, 2011).

RELATED WORK

This section discusses some research work which uses the different ML datasets for the training and testing stage. The shortcomings of this approach are reviewed and studied further in present study.

Alshammari and Zincir-Heywood (2009) aims to classify encrypted traffic and take SSH and Skype as the case study. The authors developed classifier trained data from one network to test on data from an entirely different network. The testing data is collected from three different places (Dalhousie traces, public traces and DARPA99 traces). Each of these traces is trained from the Dalhousie network. However, the question here is how to ensure the validity of output of the testing stage when the training and testing data for both sets of data is totally different.

Branch *et al*. (2009) the authors mentioned that Skype traffic can be identified by observing five seconds of a Skype traffic flow. The classifier achieved more than 98% accuracy and succeeded in identifying suitable traffic features to classify Skype. However, the method and datasets are used only for offline classification. The offline detection has several shortcomings such as the different environment from the online classification.

The authors in Angevine and Zincir-Heywood (2008) used AdaBoost and C4.5 to classifying the traffics into Skype and non-Skype. The Skype traces were collected as labeled data and taken from the campus network. The data were separated into UDP and TCP and classified independently. The classification results are 98 and 94% for UDP and TCP respectively. However, the labeled datasets were collected at different classification times. This causes a difference between classification environment and datasets collection. Moreover, the classifier did not identify all Skype traffics.

The researchers by Jesudasan *et al*. (2010) uses ML in their study. They focused on Skype classifications for versions 2, 3 and 4. The study used ten folds cross-validation with 100 packets sub-flow. The results showed about 98% precision and 86% recall. As has been the case with the previous works, the problem is the use of training and testing datasets which were collected from two different environments. The first group is Skype traces collected in real-time using Tcpdump (of unknown origin). The second group comprises the offline pcap fileswhich were obtained from University of Twente (saved files). Again the question of how to train the classifier by datasets collected from some network and to examine this classifier by datasets from another network, where the characteristics of the two networks may be different.

Zhang *et al*. (2010b) is a flattering work which proposes an online method based on SVM-ML to classify Skype traffics. This work has an advantage over others in data collection. All datasets (covering both training and testing) were collected from the campus network. However one question remains as to how to evaluate the classifier due to the lack of comparison to ensure classification results. We did not find a paper in the literature review that combined Snort rules and ML to identify Skype traffics.

ARE THE STATISTICAL FEATURES OF SKYPE THE SAME IN DIFFERENT NETWORK SCENARIOS?

As discussed earlier, one important issue in ML is to use valid training and testing datasets. We consider full Skype session (call) datasets to answer the question, is Internet application (Skype in particular) traffic features the same when the traffics are collected from different network environments. All data was collected by Wireshark (Orebaugh *et al*., 2006), as well, the statistical values are summarized from the same software.

Eight different Skype calls were considered and divided into two groups. The first group includes four different calls (calls 1.1-1.4). In this group, the Skype sessions are full Skype session (call) between two SCs located in two different countries. This means, both Skype clients are located behind firewall and thus using NATed IP. The second group includes other four calls (calls 2.1- 2.4) of Skype session between two SCs located inside our campus area. This group calls configures with no firewall between both clients (the two clients used real IPs). We aim to generate two different datasets to study Skype traffic features in two different network scenarios.

Table 1 and Fig. 1 show statistical features results of the two groups. For all calls of group one, the TCP rate, UDP rate, average packets per second and average packets per size (bytes) were seen to be have near values.

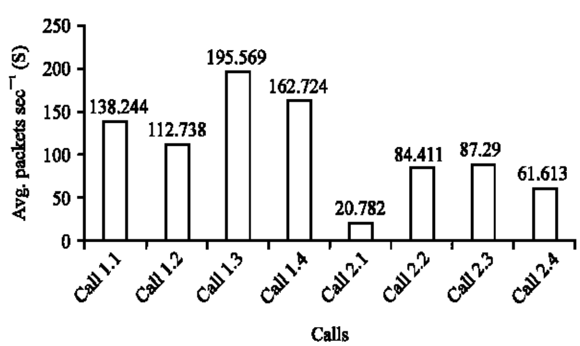

Fig. 1: Average packets per second of the two group's calls

Table 1: Skype some statistical values when network environments are different

Full call connection call	TCP rate (%)	UDP rate (%)	Avg. pckt sec^{-1} (S)	Avg. pckt $size^{-1}$
Between two different networks (firewall is on)-call1.1	99.89	0.11	138.244	128.170
Between two different networks (firewall is on)-call1.2	98.73	1.24	112.738	130.039
Between two different networks (firewall is on)-call1.3	83.16	16.84	195.569	70.633
Between two different networks (firewall is on)-call1.4	99.43	0.56	162.724	75.128
Between two Real IPs-call 2.1	5.29	90.66	20.782	147.465
Between two Real IPs-call 2.2	1.08	98.92	84.411	120.204
Between two Real IPs-call 2.3	0.30	99.70	87.290	121.991
Between one Real IPs and another non-real IP-call 2.4	1.83	98.12	61.613	120.158

This means, the same network environment, generate same traffic features. However, when any call of group one compared with other call from group two, clear differences appear in the TCP rate, UDP rate and average packets per second. This means the statistical features of Skype traffic are not the same when the network environments are different.

PROPOSED ML SKYPE CLASSIFICATION SYSTEM

In order to achieve our goal to classify Skype traffic, a ML Skype classifier system is proposed. The system is divided into two stages. The current stage includes several phases such as, to develop Skype Snort rules to collect valid Skype traffic, to capture the Skype traffic from a mirrored Internet traffic and finally to provide an offline ML Skype classifier. The future stage moves forward with online Skype classification. Both stages are combined in Fig. 3. The continuous lines represent the current stage whilst the dotted lines are components to be implemented in the next stage.

Proposed system framework

Current stage: In this section, we describe the current real time system classifier to classify Skype traffic. This includes several elements namely (1) traffic mirror to capture the Internet traffic (2) Snort sensor containing the Skype login signature (3) Skype traffic samples and (4) Offline ML for Skype classification. Figure 2 illustrates the classifier stages. Internet traffic is mirrored at one switch of the campus network. Snort sensor which includes six Skype rules, is linked to the mirror port so as to capture only Skype traffic login information. This Snort filtering will be an important stage for the Skype datasets collection. This ensures the taking of near real time training datasets from the same traffic we want to classify. The third part in the classifier collects samples of Skype traffic and prepares the samples as ML datasets. This resulting Skype datasets is input to ML as a part of the offline training and testing datasets. In parallel with this, samples of other traffics (non-Skype) will be collected from the same mirror port and input to the ML. The training datasets was preprocessed based on features selection and data amount selection. The result of the offline ML will be used to divide the traffic between Skype and non-Skype. Snort is suitable only in the offline classification stage since Snort is signature based which requires certain minimal resources (CPU processing and memory usage). In a real time of high speed network, the use of Snort for online classification will likely cause packets to be dropped.

Future stage: In addition to all parts of current stage, two new elements will added (1) Extended Skype samples which depending on traffic samples features, the packets will then be extended to get the samples of full flows or part of Skype session. (2) Online ML classification, where upon getting the benefits of offline classification in current stage, the online classification will executed.

Skype snort rules: Snort is a popular open-source network intrusion prevention and detection system. At times of attacks, network intrusion detection systems are essential for any network environment (Muthuregunathan *et al*., 2009). Snort was configured to run in three modes namely (1) Sniffer mode, which simply reads the packets of the network, (2) Packet Logger mode, which logs the packets to disk and (3) Network Intrusion Detection System (NIDS) mode, which allows Snort to analyse network traffic for matches against user-defined rules (Roesch and Sturges, 2011). The main advantage of Snort is its ability to give real-time alerts in the case when the appropriate packet contents matches with any of the rule set. At the same time, the rules can be developed and edited based on what we need to detect. With a flexible and robust rules definition language, Snort is capable of detecting any specific traffic that crosses the network. We propose six Snort rules to identify Skype traffics. These rules are all based on login and initiation information assuming that Skype is already installed in the systems and is ready for use. Consequently, we propose Skype Snort rules as a bridge before collecting ML datasets. With this, we can achieve the following:

- Collect real time datasets from the same network that we need to identify
- Ensure training and testing datasets are taken from the same place and at the same time

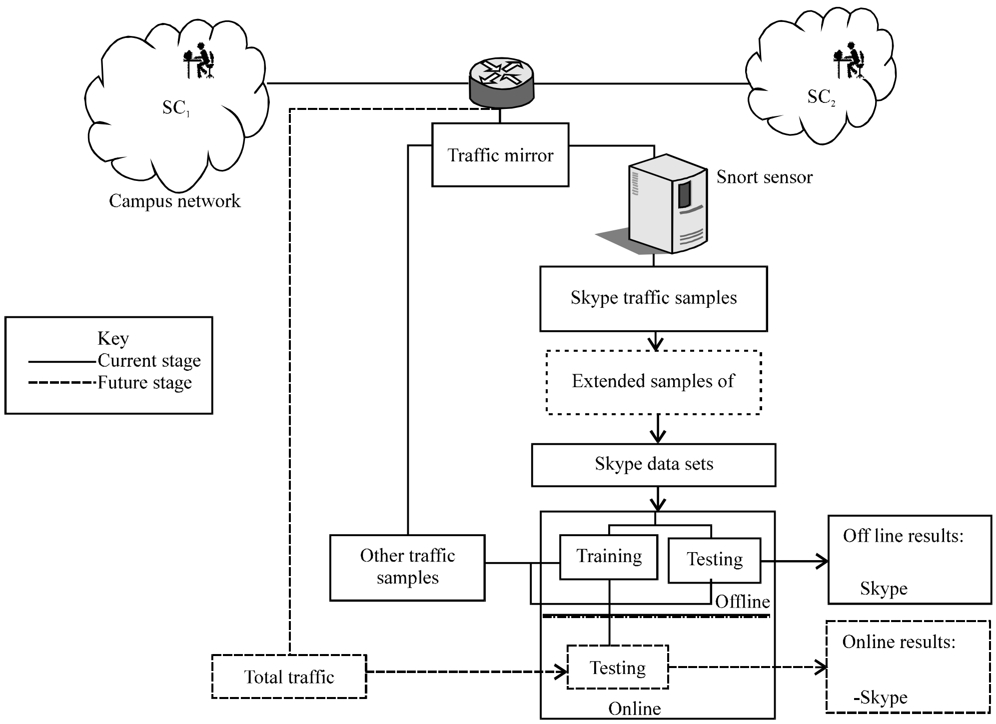

Fig. 2: Classification system stages

- Obtain valid training datasets from near real time
- Qualify the performance of the classifier

When Skype starts, it will ping using some UDP packets to see whether the remote host is alive. Other packets will also be sent for the purpose of searching for SNs. This searching, like all types of Skype signaling traffic, is encrypted. However, there are some important observable activities that can be noted. SNs then respond to the Skype client with similar UDP packet to present its status. The Skype client, by using a random port number will create a TCP connection with the first responding SN. After the TCP session has been established with SN, the Skype client will connect to login to the login server. This is always done using TCP packets. Once the Skype client has connected to the login server, the first packet with PSH and ACK flags set will be sent. Skype will check for the last version with ui.skype.com and try to update to the newest one if that version is not already installed. More details about Skype architecture and traffic is discussed in Bonfiglio *et al*. (2007), Adami *et al*. (2012) and Price (2012). Depending on the above scenarios, the following Snort rules are developed:

- Alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"Skype client look for newest version, login1"; flow:to_server,established; content:"Host|3A| ui.skype.com"; classtype:policy-violation; sid:1000001; rev:1)
- Alert udp any any -> any any (msg:" Skype login signature "; flow:to_server,established; content:"|04 64 73 6e 37 01|"; classtype:policy-violation; sid:1000002; rev:1)
- Alert tcp any any -> any any (msg:" Skype login signature "; flow:to_server,established; content:"| 04 63 6f 6e 6e 05 73 6b 79 70 65 03 63 6f 6d 00|"; classtype:policy-violation; sid:1000003; rev:1)
- Alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:" Skype login signature"; flow: to_server,established; content:"conn. skype.com"; classtype:policy-violation; sid:1000004; rev:1)
- Alert tcp EXTERNAL_NET any -> $HOME_NET any (msg:" Skype login signature"; content:"sn7.d.skype. net"; sid:1000005; rev:1)
- Alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:" Skype login signature, login6"; flow: to_client,established;content:"|6b 79 70 65 03|"; depth:31;classtype:policy-violation;sid:1000006; rev:2)Rules

The above rules were tested using Linux (Centos 5.5) and Windows7 with Snort version 2.9.3.1. Manual observation was carried out through the campus network. This was done by running Skype from our server at university computer center and monitoring certain IP addresses and the generated alerts whenever the rules are matched. The rules were successful in getting alerts with Skype login. In the same manner, the rules were able to give alerts for controlled single machines, such as laptops or desktop computers.

ML CLASSIFICATION

For our ML classification, three different experimental scenarios were considered, (1) Training and testing datasets collected from NATednetwork i.e., a private network (same dataset environment), (2) Training and testing datasets collected from real IP network (same dataset environment) and (3) Both from different networks.

For the first case, the idea of the classifier stages in Fig. 2 was applied. We used our Snort rules to capture Skype traffic (only login packets). The other traffic (non-Skype) was captured from the same mirror of the campus network. These datasets were divided semi equally (less than 10 packets difference) into two parts, training and testing. Our goal is to train ML by data collected from near real time from the network we need to classify. Skype traffic and non-Skype traffic were mixed to be input to the ML. The non-Skype datasets were captured in real time from the campus network which includes all internet applications traffic.

In the second case, we repeated the same scenario of the previous case but using real public network. We put both the SCs in our DMZ segment which uses real public IP without any firewall rules. In the third case, we used the same training dataset of the previous experiment (case 2). However, the testing datasets were collected from the private network. This means training and testing data were from different networks with different characteristics. The Skype testing data was collected in the same manner as the first testing data (using Snort rules).

Different experiments were conducted to select the best ML algorithms and features so as to get the optimum results. We applied feature filtering to reach better classification results. After conducting many experiments and observations, we found that the high accuracy with low False Positive (FP) can be obtained when we used only packet length as the traffic feature. Remember, here we are ONLY considering Skype login datasets. Furthermore, more than ten ML algorithms were tested before the optimum classifier could be identified. We compared the results of only four of these algorithms namely (1) Trees.J48 (2) Trees. ADtree (3) Trees.randomForest and (4) Rules.decisionTable. All these algorithms are part of Weka. Weka itself is a collection of algorithms and has data preprocessing tools (Witten and Frank, 2005).

Table 2-4 and Fig. 3 and 4 show the ML classification results. TP rate, FP rate, precision and recall are considered as metrics for our classifier. The results show that TP is better and FP is low when both training and testing data are obtained from the same network

Table 2: Training and testing datasets from same network (NAT)

	TP rate (%)	FP rate (%)	Precision (%)	Recall (%)	Time (sec)
J48					
Training	98.4	0.30	98.5	98.4	0.04
Testing	99.2	0.50	99.2	99.2	0.01
Trees ADtree					
Training	94.1	1.30	95.6	94.1	0.03
Testing	88.0	3.00	92.7	88.0	0.02
Trees Random Forest					
Training	98.4	2.50	98.4	98.4	0.06
Testing	99.1	0.90	99.1	99.1	0.04
Rules Decision Table					
Training	96.3	0.08	97.0	96.3	0.06
Testing	88.3	3.30	92.7	88.3	0.01

Table 3: Training and testing datasets from same network (Real IPs)

	TP rate (%)	FP rate (%)	Precision (%)	Recall (%)	Time (sec)
J48					
Training	99.4	1.2	99.4	99.4	0.02
Testing	99.2	1.2	99.2	99.2	0.00
Trees AD tree					
Training	99.3	0.8	99.3	99.3	0.03
Testing	99.2	1.2	99.2	99.2	0.01
Trees Random Forest					
Training	99.9	0.0	99.9	99.9	0.04
Testing	99.6	0.1	99.6	99.6	0.03
Rules Decision Table					
Training	99.5	0.4	99.5	99.5	0.05
Testing	99.2	18.1	99.2	99.2	0.02

Table 4: Training from Real IP network and testing from NAT network (different environment)

	TP rate (%)	FP rate (%)	Precision (%)	Recall (%)	Time (sec)
J48					
Training	99.4	1.2	99.4	99.4	0.03
Testing	82.4	26.0	85.5	82.4	0.01
Trees ADtree					
Training	99.3	0.8	99.3	99.3	0.05
Testing	82.4	25.6	85.5	82.4	0.02
Trees Random Forest					
Training	99.9	0.0	99.9	99.9	0.04
Testing	82.8	24.5	85.9	82.8	0.03
Rules Decision Table					
Training	99.5	0.4	99.5	99.5	0.04
Testing	82.6	24.6	85.8	82.6	0.02

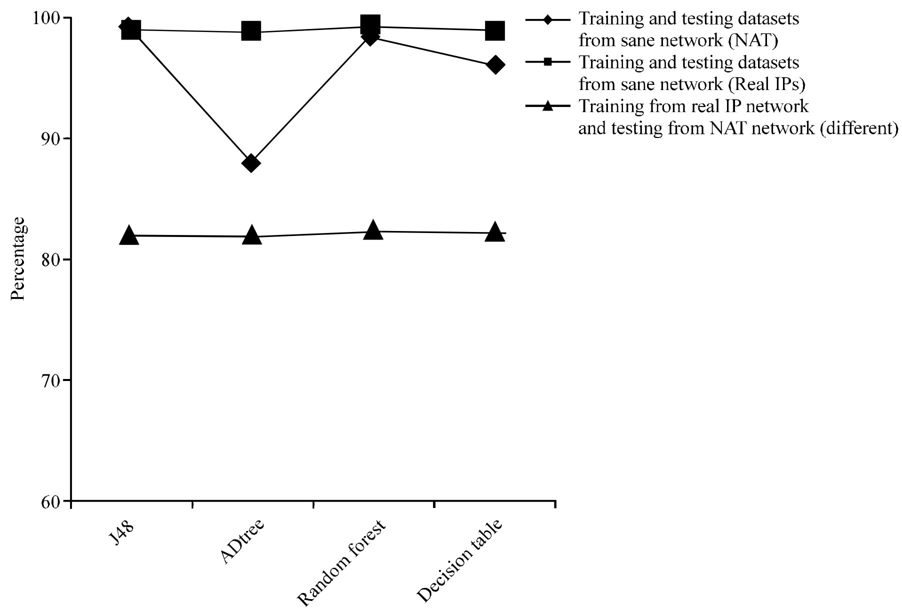

Fig. 3: TP for datasets from same and different networks

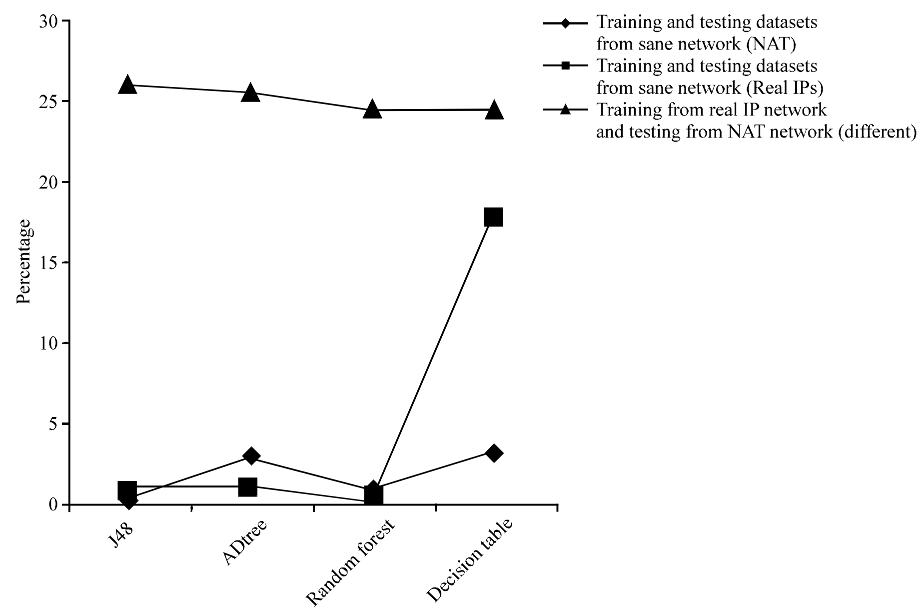

Fig. 4: FP for datasets from same and different networks

environment. As for the ML algorithms, J48 is the best between the other three algorithms, which achieved higher TP (99.2%) and lower FP (0.5%).

CONCLUSION, LIMITATIONS AND FUTURE WORK

Interactive applications such as Skype and online games use several approaches to prevent from being detected. Supplying ML with real and valid training datasets is an important issue for IP traffic classification. In this paper, two network scenarios are considered to check Skype statistical features. We conclude that some of Skype statistical features such as packets per second are varying when network environment is different.

We propose a mechanism based on Snort rules and statistical approaches to obtain valid ML Skype dataset.

Six basic Snort rules were proposed to prepare Skype data to be used later for the classifier. All rules are based on Skype login information. Here, the main benefit is to train ML in near real-time by datasets from the same traffic segment that we want to classify. Three different experimental scenarios are considered. In the first and second case, the training and testing datasets were collected from same network segment and the third from different networks. Four ML algorithms: J48, ADtree, Random Forest and Decision Table are used to evaluate the results of the three cases. The comparisons show that the TP is high and FP is low when both training and testing data are from same network environment.

The method has some limitations. Firstly, the training and testing data of Skype was taken from only login data (extended stage is not applied yet). Secondly the features selected are expected to change in the next work which will be more appropriate with realistic data traffic. Finally, the number of data (packets) input to the classifier is not many. We anticipate some of the problems will be overcome once we move to the next stage where the full Skype session including the user's sessions will be included. We hope to extend this even further and find the best features to classify Skype traffic irrespective of the training data.

REFERENCES

Adami, D., C. Callegari, S. Giordano, M. Pagano and T. Pepe, 2009. A Real-Time Algorithm for Skype Traffic Detection and Classification. In: Smart Spaces and Next Generation Wired/Wireless Networking, Balandin, S., D. Moltchanov and Y. Koucheryavy (Eds.). Springer-Verlag, Berlin, Heidelberg, Germany, pp: 168-179.

Adami, D., C. Callegari, S. Giordano, M. Pagano and T. Pepe, 2012. Skype-Hunter: A real-time system for the detection and classification of Skype traffic. Int. J. Commun. Syst., 25: 386-403.

Alshammari, R. and A.N. Zincir-Heywood, 2009. Machine learning based encrypted traffic classification: Identifying SSH and Skype. Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, July 8-10, 2009, Ottawa, ON, USA., pp: 1-8.

Alshammari, R. and A.N. Zincir-Heywood, 2010. An investigation on the identification of VoIP traffic: Case study on Gtalk and Skype. Proceedings of the International Conference on Network and Service Management, October 25-29, 2010, Niagara Falls, ON, USA., pp: 310-313.

Alshammari, R. and A.N. Zincir-Heywood, 2011a. Can encrypted traffic be identified without port numbers, IP addresses and payload inspection? Comput. Networks, 55: 1326-1350.

Alshammari, R. and A.N. Zincir-Heywood, 2011b. Is machine learning losing the battle to produce transportable signatures against VoIP traffic? Proceedings of the IEEE Congress on Evolutionary Computation, June 5-8, 2011, New Orleans, LA., USA., pp: 15436-15450.

Angevine, D. and A.N. Zincir-Heywood, 2008. A Preliminary investigation of Skype traffic classification using a minimalist feature set. Proceedings of the 3rd International Conference on Reliability and Security, March 4-7, 2008, Barcelona, Spain, pp: 1075-1079.

Baset, S.A. and H.G. Schulzrinne, 2006. An analysis of the Skype peer-to-peer internet telephony protocol. Proceedings of the 25th IEEE International Conference on Computer Communications, April 2006, Barcelona, Spain, pp: 1-11.

Bonfiglio, D., M. Mellia, M. Meo, D. Rossi and P. Tofanelli, 2007. Revealing Skype traffic: When randomness plays with you. Proceedings of the Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, August 27-31, 2007, Kyoto, Japan, pp: 37-48.

Branch, P.A., A. Heyde and G.J. Armitage, 2009. Rapid identification of Skype traffic flows. Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video, June 3-5, 2009, Association Computing Machinery, New York, USA, pp: 91-96.

Chen, K.T., C.Y. Huang, P. Huang and C.L. Lei, 2006. Quantifying skype user satisfaction. Proceedings of the Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, September 11-15, 2006, Pisa, Italy, pp: 399-410.

Freire, E.P., A. Ziviani and R.M. Salles, 2008. Detecting Skype flows in Web traffic. Proceedings of the Network Operations and Management Symposium, April 7-11, 2008, Salvador, Bahia, Brazil, pp: 89-96.

Ibrahim, H.A.H., S.M. Nor, A. Mohammed and A.B. Mohammed, 2012. Taxonomy of machine learning algorithms to classify real time interactive applications. Int. J. Comput. Networks Wireless Commun., 2: 69-73.

Jesudasan, R.N., P. Branch and J. But, 2010. Generic attributes for Skype identification using machine learning. Technical Report No. 100820A, Centre for Advanced Internet Architectures, Swinburne University of Technology

Jun, L., Z. Shunyi, X. Ye and S. Yanfei, 2007. Identifying Skype traffic by random forest. Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, September 21-25, 2007, Shanghai, China, pp: 2841-2844.

Molnar, S. and M. Perenyi, 2011. On the identification and analysis of Skype traffic. Int. J. Commun. Syst., 24: 94-117.

Muthuregunathan, R., S. Siddharth, R. Srivathsan and S.R. Rajesh, 2009. Efficient snort rule generation using evolutionary computing for network intrusion detection. Proceedings of the 1st International Conference on Computational Intelligence, Communication Systems and Networks, July 23-25, 2009, Indore, MP, India, pp: 336-341.

Nguyen, T.T.T. and G. Armitage, 2006. Training on multiple sub-flows to optimise the use of Machine Learning classifiers in real-world IP networks. Proceedings of the 31st IEEE Conference on Local Computer Networks, November 14-16, 2006, Tampa, FL., USA., pp: 369-376.

Nguyen, T.T.T. and G. Armitage, 2008. Clustering to assist supervised machine learning for real-time IP traffic classification. Proceedings of the IEEE International Conference on Communications, May 19-23, 2008, Beijing, China, pp: 5857-5862.

Orebaugh, A., G. Ramirez and J. Beale, 2006. Wireshark and Ethereal Network Protocol Analyzer Toolkit. Syngress, USA., ISBN-13: 9780080506012, Pages: 448.

Price, C., 2012. How Skype work? http://ezinearticles.com/?How-Skype-Works&id=496462

Roesch, M. and S. Sturges, 2011. Snort users manual 2.9.2, 2011. http://www.snort.org/.

TNG, 2008. Skype traces. Telecommunication Networks Group. http://tstat.tlc.polito.it/traces-skype.shtml.

Weirong, J. and M. Gokhale, 2010. Real-time classification of multimedia traffic using FPGA. Proceedings of the International Conference on Field Programmable Logic and Applications, August 31-September 1, 2010, Milano, Italy.

Witten, I.H. and E. Frank, 2005. Data Mining Practical Machine Learning Tools and Techniques. 2nd Edn., Morgan Kaufman, San Francisco, CA.

Zhang, D., C. Zheng, H. Zhang and H. Yu, 2010a. Identification and analysis of Skype peer-to-peer traffic. Proceedings of the 5th International Conference on Internet and Web Applications and Services, May 9-15, 2010, Barcelona, Spain, pp: 200-206.

Zhang, H., G. Zhimin and T. Zhenqing, 2010b. Skype traffic identification based SVM using optimized feature set. Proceedings of the International Conference on Information Networking and Automation, October 18-19, 2010, Kunming, China, pp: 431-435.