VoIP Classification

Irengbam Tilokchan Singh
Department of Computer Science
Manipur University, Imphal, Manipur, India

Tejmani Sinam
Department of Computer Science
Manipur University, Imphal, Manipur, India

Thounaojam Rupachandra Singh
Department of Computer Science
Manipur University, Imphal, Manipur, India

Abstract- Human communication technology has changed dynamically with time. With emerging trends and technologies, users nowadays are shifting from traditional phone call to VoIP (Voice over IP) applications viz *Skype, Google Talk, Google+ Hangout, Asterisk* etc. These VoIP applications generate a huge amount of network traffic. And these VoIP traffic need a proper classification by Government agencies due to security reason or by ISP or Network operators for billing application specific traffic or even by the Network Administrator of an Institution to implements QoS or to monitor their network. Thus, Network traffic classification plays an important role in the areas of network security, network monitoring, QoS and traffic engineering. In this paper, we propose a novel approach to identify VoIP Network Traffic in the first few seconds of initial state of communication. The proposed classifier works with Machine Learning Techniques based on the statistical features. The experimental results show that the proposed method can achieve over 99% accuracy for all testing dataset.

Keywords – VoIP Traffic identification, Network Traffic Classification, Machine Learning

I. INTRODUCTION

In general, network traffic classification is a fundamental process to classify the network traffic and identify the corresponding applications in modern network security systems, network monitoring, QoS and traffic engineering. Traditional method of traffic classification are done based on the application port mapping which are assigned by IANA (*Internet Assigned Numbers Authority*), protocol format analysis and payload based matching approach. But today, emerging applications uses ephemeral, dynamic and random ports and encrypted payloads for obfuscation. So, the traditional methods of traffic classification (*port based prediction* and *payload based deep inspection method*) [1]–[4] are no longer effective and efficient. Most researchers are diverting away from these old techniques of classification and are adopting the statistical based classification techniques.

Several significant studies have previously been carried out on traffic classification based on Machine Learning (ML) [5]–[11]. Some are focused on clustering techniques which are unsupervised Machine Learning algorithms and some are based on supervised Machine Learning method, which deals with training the classifier with known datasets. The method proposed in this paper is a hybrid approach based on the combination of both unsupervised and supervised methods.

With emerging trends and technologies, users nowadays are shifting from traditional phone call to VoIP applications. These applications are mostly encrypted; some like Skype uses P2P architectures and have the capability to traverse any network conditions. So, there's a lot of interest among research community, network operators and even Government agencies, in identifying these applications. Some of VoIP applications that we have considered in our study are *Skype, Gtalk, Asterisk* and *Google+ Hangouts.*

In the proposed method, media traffic flows of a particular application are gathered first. These flows are further split into *sub-flows* using sliding windows. The term *sub-flow* is defined as subsets of a flow, having the same 5 tuple (*src ip, dst ip, src port, dst port* and *protocol*) with time based windows size (3, 5, 7 and 10 seconds) and are obtained by sliding windows. These windows are overlapping. Let us consider how to to obtain sub-flow window size of *5* second; the 1^{st} window start from *0* second to *5* second; the 2^{nd} window start from *1* second to *6*

second; and k^{th} window start fom *k-1* to *k+4* second; thus these window are sliding with *1* second and overlapping with *4* second. The reasons for considering sub-flows are:

1) for early classification of VoIP media Traffic and

2) to enable the classifier to work in real time or on- the-fly.

The security, monitoring and management systems need the prior information, but not the post-mortem report.

From these *sub-flows* we extract the statistical flow features. These flow features consisting number of packet, packet size, minimum packet size, maximum packet size, the first and second order statistics over packet size, and packet inter-arrival (*minimum, maximum, average and standard deviation*) are obtained using overlapping sliding window.

The traces used in our study were captured on the client side and at the edge of our University Network during 2011- 2013. The ground truth traces are categorized into two set (*training* and *testing*). From these two set, we extracted the eature datasets for each application. And *k*-mean clustering was performed on each application datasets to group the training set for each application. The value of *k* is determined from the result of DBSCAN clustering. Lastly, we balance training and testing set by acquiring the data proportionality of each cluster of each application.

In our study, we use four supervised Machine Learning classification algorithm viz., *Decision Tree (C4.5), Naive Bayes, Bayesian Belief* and *SVM* [12]–[15]. First of all, we build different classifier model based on *sub-flow* statistic of 22 attributes feature derived by varying window size of 1, 3, 5, 7 and 10 seconds. Analysing the classifier models with deferring windows, we select C4.5 with 3 second window for further analysis as this model gives better performance than other models. Further, single class classifier models are also analysed with the 3 second window based C4.5 classifier model. And from the result it is found that the attribute set has the potential of acquiring the characteristics of applications. We also apply selected attribute model to test Tstat trace [16], [17].

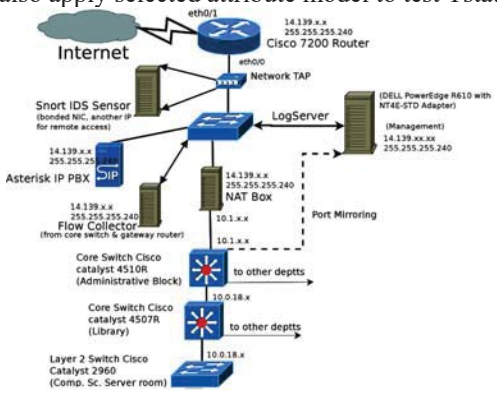

Figure 1. Data Collection Architecture

The rest of this paper is organized as follows. Section 2 covers the related work with more emphasis on statistics based Internet traffic classification approaches. Section 3 outlines the data used and how they are collected. Section 4 describes the methods that are proposed. Section 5 provides a detailed analysis on our classification approach and its performance evaluation regarding the experimental traffic classification. And discuss the performance measures. Section 6 concludes the paper with some final remarks and suggestions of possible future work.

II. Related Work

In the early days, classification of network traffic were obtained through the base knowledge of IANA assigned reserved port numbers. Due to the advertent attempt to bypass traffic using ephemeral ports by newer applications such as those using P2P has rendered port-based classification ineffective. This was confirmed by Karagiannis et. al [18] by identifying P2P on handcrafted signatures. Haffner et.al. [19] automated the construction of application signatures on trained sets by employing supervised machine learning techniques. Jeffrey Erman et.al. [20] showed the performance of k-means clustering better than DBSCAN and EM clustering algorithms. Jeffrey Erman et.al. [21] proposed a semi-supervised learning utilizing k-means clusters. This study was based on statistics of the flow. The k-means clustering has the drawback of assigning the number of cluster and the number of cluster to be formed can not be predicted. So, Yu Wang et.al. [22] employed X-means clustering to their work. Although X-means is basically equivalent to k-means, it does not require the assignment of the number of clusters in advance. Xiang Li et.al. [23] applied Support Vector Machine learning based on flow statistics to identify and classify network applications. Nowadays, researchers are more or less attracted towards statistical based approach as it does not

involves packet payload data. Many researchers attempted to build statistical classifier models based on full flow feature. However, full flow features approaches exhibit slower performance. Recently, researchers have initiated the use of statistical approach based on sub-flow feature sets [5],[24]–[26]. The packet size and inter-arrival time are more effective measurable features in early classification of network flows as shown in [5],[27], [28].

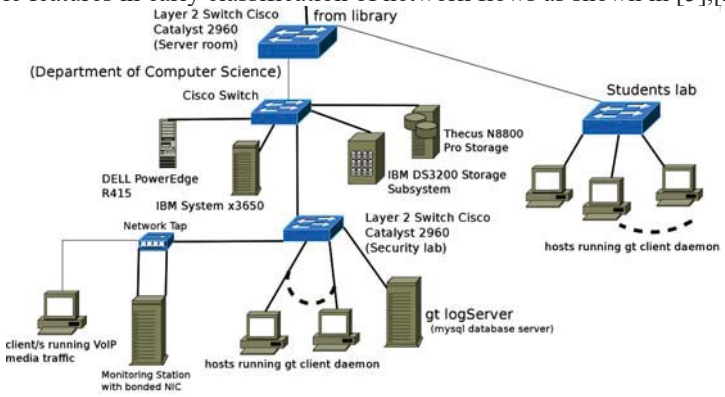

Figure 2. Testbed Laboratory

III. Data Collection

Network traces are collected from our testbed, at the edge of our University Network (Figure 1) and from publicly available traces of *Tstat* [16] [17]. The testbed is setup at Network Security Lab, M.U. (*Manipur University*) (Figure 2) where various VoIP application traces are generated. And using gt's [29] method we collect ground truth application traces. A Napatech data capture card, NT4E-STD[30] was used to capture traces on our log server at the edge of our University Network (*Figure 1*). A VoIP server is also running at the public domain where traces of Asterisk based VoIP applications are collected. We collected various types of Skype and non-Skype traces such as voice, video, silence call, call within LAN and WAN, etc.

Data were generated using the VoIP clients such as Skype (Beta) version 2.2.0.35, linphone 3.5.2 (Windows 7), linphone 3.3.2 (Linux mint 13), sipdroid 2.7 beta, Ekiga Softphone 3.3.2, 3CXPhone 6.0.26523.0 (Windows 7), Gtalk in Google Chrome v20.0.1132.57, Gtalk in Google Chrome v23.0.1271.91, Gtalk with Empathy 3.4.2.3, Google+ Hangout and Asterisk 11.0.0 beta1 using Android 4.0.3 (ICS), Android 4.1-4.3 (Jelly Bean), Windows 7, linux-mint 13 and Ubuntu 12.04-14.04.

For the experiment, we were able to collect ≈ 32 GB of VoIP traces including Skype's 14.71 GB, spread over 3-4 months. And another 3.8 GB of Skype's anonymized traces was obtained from Tstat. We only downloaded end to end Skype UDP traffic from Tstat. From these anonymized traces we extracted about ≈ 586 GB of packet size as derived from the header IP lengths.

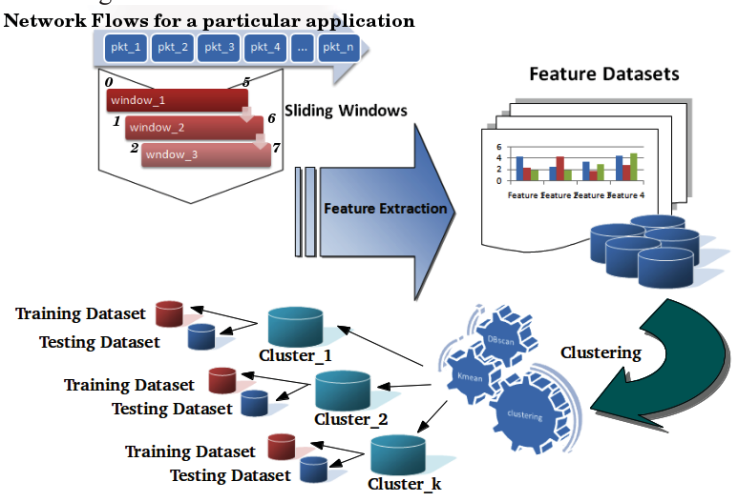

Figure 3. Application Feature Datasets extraction to gather training and testing datasets, proportionality sampling by using clustering

IV. METHOLOGY

A flow is defined by 5-tuple attribute viz., *Protocol, source IP, source port, destination IP* and *destination port*. We made use of bidirectional flow in our study. Many classification approaches use full flow feature and classification is done only when flow terminates. This approach becomes useless in early identification of the flow with the intention of applying QoS, monitoring, etc. It is essential that the flow is identified as early as possible so that policy decision can be applied. In this study, we make use of *sub-flow* level statistics and avoid deep packet inspection. Researchers have made use of characteristics of *sub-flow* concepts to classify the flow instead of full flow based models [24]–[26] even when few beginning packets are lost. Our model aims to classify the traffic based on few packets. We made use of media traffic generated by VoIP applications viz., *Skype, Gtalk, Hangout* and *Asterisk.* We extracted the statistical *sub-flow* features of the media traffic using overlapping sliding windows as shown in Figure 3. From the training datasets of Figure 3 build the supervised Machine Learning Classifier model. Figures 4 and 5 shows the proposed machine learning traffic classification model building and the classifier which classify both the offline datasets and real time traffic classifier with supervised Machine Learning. Our system goes through the following stages:-

1) Training and testing datasets preparation,
2) Model building and
3) Experiments with Various Traffic Classifier.

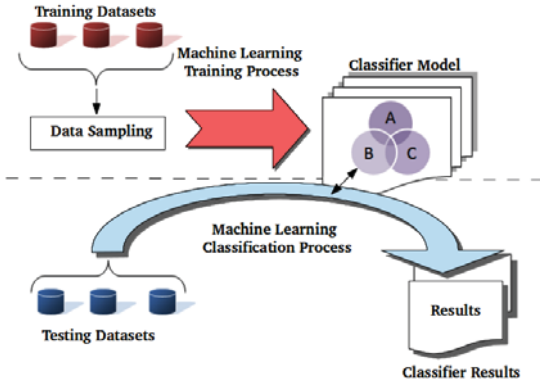

Figure 4. Training the Machine Learning traffic classifier and classification of offline datasets

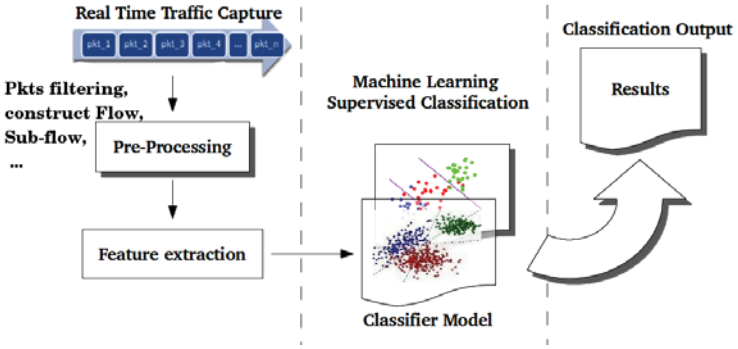

Figure 5. Real Time Traffic classifier with supervised Machine Learning

4.1. Training and Testing Dataset Preparation

We consider only the ground truth VoIP media flow traffic. Application based Feature Dataset extraction is performed to gather the statistical information for training and testing datasets. Our proposed system extracts the *sub-flow* statistical feature of media traffic. The *sub-flow* information is extracted from media flow using overlap sliding windows concept. The statistical features considered in our study are given in table I. These statistical *sub-flow* features of a sliding window consists of packet counter, packet size, packet inter-arrival time and their derivative features (first order statistics *(minimum, maximum and average)* and second order statistic *(standard deviation)*). Based on packet size, the attributes are further categorized into low and high to represent voice and video traffic respectively. We take these extracted and derived statistical features of the media traffic sub-flow and from that we build the classifier signature model. Figure 3 shows the big picture to extract feature dataset for training and testing. Using k-mean clustering, we group similar data points into clusters for each applications, so that sampling of the training data points incorporate the data proportionality of the applications' divergent characteristics

induced by the use of different codec for different applications. The number of cluster *(k)* to be formed is obtained from the result of DBSCAN clustering. Then, sampling of the training data points are carried out through WEKA sampling procedure from the clustered data points of the application incorporating the data proportionality of the application.

For each of the application, a sample of 6,000 tuples is randomly selected from the original data. So, altogether we use 18,000 tuples which is used to build the model. The training dataset is shown in table II. Similarly, testing data points are selected from the separate testing trace. So, altogether the testing data points consists of 18,000 data points contributed by the three application.

4.2. Model Building

4.2.1. Determination of best classifier model and ?-second windows: We build various classifier based on the following machine learning algorithms: *NB (Naive Bayes), BBN (Bayesian Belief Network), C4.5* and *SVM (Support Vector Machine)*. Implementation of all these machine learning algorithms are done using *WEKA* [31]. The *sub-flow* information are overlapping windows sliding through the media flow. The *sub-flow* performance are compared with different sliding windows size. Table III shows result of testing the classifier model with window size of 1, 3, 5, 7 and 10 second.

From table III, we can see that C4.5 classifier is the best among the ML classifiers used in our experiments. C4.5 classifier model on 3-second sliding window achieved the highest result of 99.47%. So, C4.5 classifier algorithm is determined as the best classifier model with 3-seconds window and chosen for further studies. The precision and recall value for 3-seconds window based on C4.5 classifier is shown in the table IV.

TABLE I. DESCRIPTION OF STATISTICAL FEATURE DATASET

No.	Feature Description	Abbreviation
1	Total number of packets in a window	**P_num**
2	Total number of bytes in a window	**Total_P_size**
3	Minimum packet size in a window	**min_P_size**
4	Maximum packet size in a window	**max_P_size**
5	Average packet size in a window	**Ave_P_size**
6	Standard packet size in a window	**std_P_size**
7	Total number of packets in the low category	**lowP_num**
8	Total number of bytes in the low category packets	**low_Total_P_size**
9	Minimum packet size in a window in the low category packets	**low_min_P_size**
10	Maximum packet size in a window in the low category packets	**Low_max_P_size**
11	Average packet size in a window in the low category packets	**low_Ave_P_size**
12	Standard packet size in a window in the low category packets	**low_std_P_size**
13	Total number of packets in the high category	**high_P_num**
14	Total number of bytes in the high category packets	**high_Total_P_size**
15	Minimum packet size in a window in the high category packets	**high_min_P_size**
16	Maximum packet size in a window in the high category packets	**high_max_P_size**
17	Average packet size in a window in the high category packets	**high_Ave_P_size**
18	Standard packet size in a window in the high category packets	**high_std_P_size**
19	minimum of the interarrival in a window	**min_time**
20	maximum of the interarrival in a window	**max_time**
21	Average of the interarrival in a window	**Ave_time**
22	Standard deviation of the interarrival in a window	**std_time**

TABLE II . NUMBER OF DATASET SAMPLE POINTS

Model	**Data sample points**
Skype vs (Gtalk+Hangouts+Asterisk)	6000 (Skype) + 6000 (Other)
Asterisk vs (Gtalk+Hangouts+Skype)	6000 (Asterisk) + 6000 (Other)
Gtalk/Hangout vs (Skype+Asterisk)	6000(Gtalk/Hangout)+6000(Other)

TABLE III. COMPARISON OF PERFORMANCE MEASUREMENT WITH DIFFERENT ML ALGORITHMS BASED ON 22 ATTRIBUTES WITH DIFFERENT WINDOWS SIZE (MULTICLASS CLASSIFIERS)

Window	**C4.5**	**BBN**	**NB**	**SVM**
1-second	99.18%	95.99%	42.23%	98.25%
3-second	**99.47%**	96.61%	37.88%	99.05%
5-second	99.30%	96.64%	40.81%	97.63%
7-second	99.24%	97.32%	42.88%	96.56%
10-second	99.36%	98.06%	48.06%	98.93%

TABLE IV. PRECISION AND RECALL VALUE FOR 3-SECOND WINDOW BASED C4.5 CLASSIFIER (MULTICLASSC LASSIFIERS)

Class	**Precision**	**Recall**
Asterisk	0.99	0.99
Skype	0.99	0.99
Gtalk	0.99	0.99

4.3 Experiments with Various Traffic Classifier.

4.3.1. Analysis of Single Class Classifier:

Experiments was carried out for single class classifier for each of the application (*Skype, Gtalk, Hangouts* and *Asterisk*). The training of the model is done with 10-fold cross-validation. In order to perform the experiments, we used training dataset sample points as given in table II. This training dataset consists of 22-attributes obtained using 3-seconds sliding windows. From table V, we conclude that the C4.5 classifier performs the best, giving more than 99% accuracy in all the single class classifier. The experiment carried out in this approach uses all the 22-features which comprises of time-relevant and time-irrelevant. In the next experiment, we will discuss the performance measured based on time-relevant and irrelevant attributes feature and with different feature selection algorithms.

TABLE V. PERFORMANCE ACCURACY FOR THE EXPERIMENTATION OF SINGLE CLASS CLASSIFIERS

Model	**C4.5**	**BBN**	***NB***	**SVM**
Skype vs (Other)	**99.45%**	97.83%	36.16%	99.09%
Asterisk vs (Other)	**99.56%**	97.91%	56.87%	98.81%
Gtalk/Hangout vs (Other)	**99.56%**	97.73%	32.86%	98.76%

4.3.2. *Attribute selection algorithms:*

We used feature selection algorithms viz., CON, CFS, CHI-SQUARE. The CFS (*Co-relation based Feature Selection*) [32] and CON (*CONsistency based feature selection*) [33] are both subset selection procedure based on best first search methods, where as CHI- SQUARE feature selection is based on ranking methods [34].

TABLE VI. ATTRIBUTES SELECTED BY DIFFERENT FEATURE SELECTION ALGORITHMS

Algorithm	Selected Features
CFS	(5 features subset selected) min P size, max P size, low min P size, high min P size, max interArrival time
CON	(5 features subset selected) Total P size, min P size, max P size, low Total P size, max interArrival time
CHI	(first 9 feature ranking attribute) min P size, max P size, high min P size Total P size, high max P size, low min P size, high Ave P size, high Total P size, Ave P size

From the 22 statistical features as shown in table I, the most relevant attributes are selected using the above feature selection algorithms. These features include the time relevant information (table VI). The above selected features (by *CFS, CON, CHI*) are used to train and test C4.5 classifier. The results are shown in table VII. Among them, top 9 rank CHI based selected features has highest accuracy.

TABLE VII. THE PERFORMANCE RESULT BASED ON TIME-RELEVANT AND IRRELEVANT FEATURE WITH 3-SECOND WINDOW SIZE C4.5 CLASSIFIER MODEL**.**

	time-relevant feature	time-irrelevant feature
CFS	98.82%	96.80%
CON	98.98%	98.88%
CHI	**99.15%**	**99.15%**

The CHI selected features are found to give higher accuracy than the other feature selection algorithms (table VII). Table VIII shows the performance measurement (precision and recall) of using CHI based selected features. The recall value of Asterisk is not good in comparison with that of Gtalk and Skype.

TABLE VIII. PRECISION AND RECALL USING CHI BASE FEATURE DATASET

Class	Precision	Recall
Asterisk	0.99	0.98
Skype	0.99	0.99
Gtalk	0.98	0.99

V.CONCLUSION

We design a network traffic classifier based on the statistical features extracted from network flows. Instead of deriving the statistical characteristics per flow, our model makes use of features extracted from the first few seconds of each flows. The first few seconds of each flow is divided into overlapping time-based windows. This approach enables our classifier to classify each flow early. Our approach of network traffic classification utilize the statistical information contained in the packet header and do not utilize any information contained in the payload. Classification involving payload is not always feasible due to encryption and also due security and privacy concerns.

Attribute selection algorithms Chi-Square, CON and CFS are used to obtain the optimal subset of features. We give a comparative analysis of the result on the said approach based on the classification algorithms (Decision tree (C4.5), Naive Bayes, Bayesian Belief Network and SVM). We also present a single class classifier implementation of C4.5 algorithm. The experimental results show that the proposed method can achieve over 99% accuracy for all testing dataset. Using the proposed method, C4.5 algorithm delivers high speed and accuracy.

Based on our experience of building a number of off-line classifiers we have developed a number of online network traffic classifiers based on Machine Learning and heuristics techniques. These classifiers will be deployed in our university network and the experience gained will be utilized to further refine our classifiers. Our group is in the process of obtaining more datasets from various sources which we intend to use to test our classifier for robustness.

VI. ACKNOWLEDGMENT

This work was supported by a grant from the Department of Electronics and Information Technology (DeitY), Ministry of Communication and Information Technology, Government of India by funding the project “Network Traffic Classification and Analysis”.

REFERENCES

[1] S. Sen, O. Spatscheck, and D. Wang, "*Accurate, scalable in- network identification of p2p traffic using application signatures*," in Proceedings of the 13th International Conference on World Wide Web. New York, NY, USA: ACM, 2004, pp. 512–521.

[2] "*l7-filter application layer packet classifier for linux*," 2009, http: //l7filter.sourceforge.net.

[3] T. Sinam, I. T. Singh, P. Lamabam, and N. N. Devi, "*An efficient technique for detecting skype flows in udp media streams*," in Advanced Networks and Telecommuncations Systems (ANTS), 2013 IEEE International Conference, Dec 2013, pp. 1–6.

[4] T. Sinam, I. T. Singh, P. Lamabam, N. N. Devi, and S. Nandi, "*A technique for classification of voip flows in udp media streams using voip signalling traffic*," in Advance Computing Conference (IACC), 2014 IEEE International, Feb 2014, pp. 354–359.

[5] T. Sinam, N. N. Devi, P. Lamabam, I. T. Singh and S. Nandi, "*Early Detection of VoIP Network Flows based on Sub-Flow Statistical Characteristics of Flows using Machine Learning Techniques*," in Advanced Networks and Telecommuncations Systems (ANTS), 2014 IEEE International Conference, Dec 2014.

[6] L. Grimaudo, M. Mellia, E. Baralis, and R. Keralapura, "*Select: Self- learning classifier for internet traffic*," IEEE Transactions on Network and Service Management, vol. 11, no. 2, pp. 144–157, 2014.

[7] T. T. Nguyen and G. Armitage, "*A survey of techniques for internet traffic classification using machine learning*," Commun. Surveys Tuts., vol. 10, no. 4, pp. 56–76, Oct. 2008.

[8] J. Chandrakant and D. Lokhande Shashikant, "*Analysis of early traffic processing and comparison of machine learning algorithms for real time internet traffic identification using statistical approach*," in Advanced Computing, Networking and Informatics- Volume 2, ser. Smart Innovation, Systems and Technologies, M. Kumar Kundu, D. P. Mohapatra, A. Konar, and A. Chakraborty, Eds. Springer International Publishing, 2014, vol. 28, pp. 577–587.

[9] R. Yan and R. Liu, "*Principal component analysis based network traffic classification*," JCP, vol. 9, no. 5, pp. 1234–1240, 2014.

[10] J. M. Reddy and C. Hota, "*P2p traffic classification using ensemble learning*," in Proceedings of the 5th IBM Collaborative Academia Research Exchange Workshop, ser. I-CARE '13. New York, NY, USA: ACM, 2013, pp. 14:1–14:4.

[11] M. Korczynski and A. Duda, "*Markov chain fingerprinting to classify encrypted traffic*," in IEEE Conference on Computer Communikations, INFOCOM , Toronto, Canada, April 27 - May 2, 2014. IEEE, 2014, pp. 781–789.

[12] "*libsvm-3.0*," http://www.csie.ntu.edu.tw/∼cjli n/libsvm/.

[13] N. Cristianini and J. Shawe-Taylor, *An Introduction to support Vector Machines and other Kernel-based Learning Methods*. Cambridge University Press, 2003.

[14] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and techniques*. Elsevier Inc., 2005.

[15] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Elsevier Inc., 2006.

[16] "*Tstat - skype traces*," http://tstat.tlc.polito.it/ traces-skype.shtml.

[17] "*Tstat - tcp statistic and analysis tool*," http://tstat.tlc.polito.it/index. shtml.

[18] T. Karagiannis, A. Broido, N. Brownlee, K. C. Claffy, and M. Faloutsos, "*Is p2p dying or just hiding?*" in Proceedings of the GLOBECOM 2004 Conference. IEEE Computer Society Press, November 2004.

[19] P. Haffner, S. Sen, O. Spatscheck, and D. Wang, "*Acas: Automated construction of application signatures*," in Proceedings of the 2005 ACM SIGCOMM Workshop on Mining Network Data, ser. MineNet '05. New York, NY, USA: ACM, 2005, pp. 197–202.

[20] J. Erman, A. Mahanti, M. F. Arlitt, I. Cohen, and C. L. Williamson, "*Semi-supervised network traffic classification*," in SIGMETRICS, 2007, pp. 369–370.

[21] J. Erman, M. Arlitt, and A. Mahanti, "*Traffic classification using clustering algorithms*," in Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data, ser. MineNet '06. New York, NY, USA: ACM, 2006, pp. 281–286.

[22] Y. Wang, Y. Xiang, and S.-Z. Yu, "*An automatic application signature construction system for unknown traffic*." Concurrency and Computation: Practice and Experience, vol. 22, no. 13, pp. 1927–1944.

[23] X. Li, F. Qi, D. Xu, and X. Qiu, "*An internet traffic classification method based on semi-supervised support vector machine*." in ICC. IEEE, 2011, pp. 1–50.

[24] T. N. Thuy T. and G. Armitage, "*Training on multiple sub-flows to optimise the use of machine learning classifiers in real-world ip networks*," in in Proceedings of the IEEE 31st Conference on Local Computer Networks, 2006.

[25] S. Zander, T. T. T. Nguyen, and G. J. Armitage, "*Sub-flow packet sampling for scalable ml classification of interactive traffic*," in LCN, 37th Annual IEEE Conference on Local Computer Networks. Clearwater Beach, FL, USA: IEEE, October 22-25 2012, pp. 68–75.

[26] G. Xie, M. Iliofotou, R. Keralapura, M. Faloutsos, and A. Nucci, "*Sub-flow: Towards practical flow-level traffic classification*," in Proceedings of the IEEE INFOCOM. Orlando, FL, USA: IEEE, March 25-30 2012, pp. 2541–2545.

[27] A. Este, F. Gringoli, and L. Salgarelli, "*On the stability of the information carried by traffic flow features at the packet level*," SIGCOMM Comput. Commun. Rev., vol. 39, no. 3, pp. 13–18, Jun. 2009.

[28] L. Peng, H. Zhang, B. Yang, and Y. Chen, "*Feature evaluation for early stage internet traffic identification*," in Algorithms and Architectures for Parallel Processing, ser. Lecture Notes in Computer Science, X.-h. Sun, W. Qu, I. Stojmenovic, W. Zhou, Z. Li, H. Guo, G. Min, T. Yang, Y. Wu, and L. Liu, Eds. Springer International Publishing, 2014, vol. 8630, pp. 511–525.

[29] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano, F. Risso, and kc Claffy, "*Gt: picking up the truth from the ground for internet traffic*," Computer Communication Review, vol. 39, no. 5, pp. 12–18, 2009.

[30] "*Napatech*," http://www.napatech.com/.

[31] "*Weka3.6.2*," 2011 , http://www.cs.waikato.ac. nz /ml/weka.

[32] M. A. Hall, "*Correlation-based feature selection for machine learning*," Department of Computer Science, The University of Waikota, Hamilton, NewZealand, Tech. Rep., 1998.

[33] M. Dash and H. Liu, "*Consistency-based search in feature selection*," Artif. Intell., vol. 151, no. 1-2, pp. 155–176, 2003.

[34] K. Pearson, "*On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling*," Philosophical Magazine Series 5 50 (302), pp. 157– 175, 1900.