Application Identification via Network Traffic Classification

Baris Yamansavascilar
Department of Computer Engineering
Yıldız Technical University
Istanbul, TURKEY
barisyamansavascilar@gmail.com

M. Amac Guvensan, A. Gokhan Yavuz, M. E.Karsligil
Department of Computer Engineering
Yıldız Technical University
Istanbul, TURKEY
{amac, gokhan, elif}@ce.yildiz.edu.tr

***Abstract*— Recent developments in Internet technology have led to an increased importance of network traffic classification. In this study, we used machine-learning methods for the identification of applications using network traffic classification. Contrary to existing studies, which classify applications into categories like FTP, Instant Messaging, etc., we tried to identify popular end-user applications such as Facebook, Twitter, Skype and many more individually. We are motivated by the fact that individual identification of applications is of high importance for network security, QoS enforcement, and trend analysis. For our tests, we used UNB ISCX Network Traffic dataset and our internal dataset, consisting of 14 and 13 well-known applications respectively. In our experiments, we evaluated four classification algorithms, namely J48, Random Forest, k-NN, and Bayes Net. With the complete set of 111 features, k-NN gave the best result for the ISCX Dataset as 93.94% of accuracy using the value of k as 1, and Random Forest gave the best result for the internal dataset as 90.87% of accuracy. During the course of this study, the initial numbers of features were successfully reduced to two sets of 12 features specific to each dataset without a compromise to the success. Moreover, we observed a 2% increase in the success rate for the internal dataset. We believe that individual application identification by applying machine-learning methods is a viable solution and currently we are investigating a two-tier approach to make it more resilient to in category confusion.**

Keywords— Network Traffic Classification, Application-based, Machine Learning

I. Introduction

Classification of the network traffic is essential for ISPs and businesses to observe and manage the traffic according to their purposes. Network traffic classification has many potentials to solve business, personal, ISP and government network problems such as capacity planning, traffic engineering, fault diagnosis, application performance, anomaly detection, and trend analysis [1]. Thus, traffic classification becomes more and more important with Internet and computer networks enlarging with a growing acceleration each passing day.

There are three main methods for network traffic classification: *port-based*, *payload based,* and *machine learning based.* The port-based methods have been used to classify traditional applications, whose ports are assigned by IANA [2]. These methods are useful for applications that exploit specific protocols such as HTTP, FTP, POP3, and ICMP. However, since many applications started to use dynamic port allocation, the success rate of port-based methods has substantially decreased over the years. In addition, some applications have not registered their ports to IANA and this condition brings about a serious limitation for port-based methods. On the other hand, the payload-based methods have been used to classify applications using the payloads of the packets. In this method, payloads are examined and signatures of known applications are sought. However, if the payload is encrypted, this approach fails. In addition to the encryption, if the signature of payload is not recognized, this method requires deep packet inspection. In recent years, machine-learning methods have been used to classify applications using only flow statistical features. Since these features are both port-independent and payload-independent, the machine learning methods provide much more flexibility.

Generally, the main focus of the studies [5, 9, 21, 22, 23] that worked on the network traffic classification is to classify protocols such as *FTP*, *HTTP*, VoIP or categories including *Instant Messaging* (*IM*), *Streaming (Video, Music, or Gaming).* In order to extract features and then create training/test set, they use an ISP or business specific dataset in their studies and then label each class to the related protocol (e.g. HTTP = 80). Afterwards, port-based, payload-based or machine learning-based methods are applied to classify the dataset and to evaluate the performance of the proposed system. This approach could be considered adequate for many cases. However, if the primary aim is to extract specific application flows such as YouTube or WhatsApp traffic from dataset, this approach is insufficient. To address this problem, studies [25, 26] used IP addresses and payloads of the applications to extract flows from the dataset. Moreover, some studies such as [9, 24] collect their own data from network by using *tcpdump* or *Wireshark* tools.

Classification of applications in a particular network is important for several reasons. The first and most important reason is the QoS (Quality of Service), which is related to the network congestion problem [27]. To evade this issue, most QoS control mechanisms have a traffic classification module in order to prioritize different applications properly across the

978-1-5090-4588-4/17/$31.00 ©2017 IEEE

limited bandwidth [23]. Security is the second important reason for the necessity of application identification. A computer network should be vigilant to possible attacks and threats. Considering the fact that todays' attacks are usually performed by exploiting popular applications such as [28, 29], the detection and classification of them become more crucial. Another important reason is that businesses and ISPs want to make trend analysis of the applications. Thus, they could build proper infrastructure for their needs based on the application utilization profile.

Generally current studies classify popular network applications into a number of specific categories such as *Streaming*, *DNS*, *Bittorent*, and *IM*. They either make use of destination IP addresses or conduct a Deep Packet Inspection (DPI) to identify the services accessed, thus put the application into the respective category. Accordingly, our contribution in this study is twofold: (1) to identify individual network applications instead of classifying them into categories; (2) to make use of machine-learning methods instead of DPI process and/or IP address look-ups. The network applications that were taken into consideration are given in Table I and Table II.

The rest of the paper is organized as follows. In Section II, we introduce related studies and their methods. In Section III, we describe the design of our system. In Section IV, experimental results are discussed in detail. Finally, we conclude the paper in Section V.

II. Related Work

Network traffic classification has taken much attention in the past decade. In network traffic classification, the machine learning methods have been used to classify applications using only flow statistical features. Since these features are both port-independent and payload-independent, the machine learning methods provide much more flexibility. Moore et al. provided a total of 249 features to characterize flows and described them in detail in [3]. This research represents an important role as most of the subsequent research in the field use of a subset of these features. In [4], Li and Luo addressed the performance issues of flow feature extraction. They realized feature extraction using four different design approaches namely *serial*, *parallel*, *pipelined*, and *hybrid*. Their experiments showed that each design approach increases the packet-processing throughput by 5- 7% in comparison with the previous method, respectively. They evaluated the results by considering the metrics of execution time, number of processed packets, processing time per packet, number of flows, and processing time per flow.

There are several different studies that used machine-learning methods. In [5], Williams et al. evaluated and compared performance and efficiency of 10 classification algorithms, and feature selection techniques. They found that the wrapper method provides the best accuracy, but is slow to execute. By using 22 features, they achieved accuracies above 97% with all tested algorithms. In [6], Nguyen and Armitage explained machine learning methods that are used for network traffic classification with details such as classification metrics, limitations, implementations and learning types. Moreover, they reviewed 18 significant studies that covered the dominant period from 2004 to early 2007. In [7], Kaoprakhon and Visoottiviseth presented a combination of keyword matching and statistical behavior profiles to classify web traffic into three types: *normal web traffic*, *video traffic*, and *audio traffic*. For statistical behavior, they used three features including *average received packet size*, *flow duration* and *ratio of packet count*. They achieved the average precision of 100% and the average recall of 84% using their 9 traces. In [8], Callado et al. made a survey on traffic identification. They investigated and evaluated all used methods including *port-based*, *packet-based* (payload), and *flow-based* (statistical) for traffic classification. They concluded their journal by saying that there is no final answer for application recognition in IP networks and none of methods achieved a high accuracy with a high precision in a broad range of applications. In [9], Wang et al. described a detailed workflow of machine learning based network traffic classification in campus network of SunYat-sen University. They collected network traffic of the university and labeled them using bro tool [30] based on payload-based method. They evaluated 7 categories for traffic classification and achieved over 90% of accuracy. In [10], Dong et al. describes the current situation and common methods of network traffic identification. They compared several ML algorithms compared their results. In [11], Huang et al. proposed a framework of cloud-based traffic classification. They designed a training pool for a PC to collect the statistical information. This statistical information was sent to the cloud for training. In the cloud, a database stored this information and a machine learning based training system was also constructed.

Some studies applied hybrid systems in network traffic classification. In [12], Lu and Xue proposed a hybrid framework to classify the Internet traffic, combining well-known port numbers and packet payload signatures. Moreover, they applied a novel heuristic-based co-clustering algorithm to classify the unknown Internet traffic along with three dimensions, namely *source IP addresses*, *destination IP addresses* and *destination port numbers*. The basic idea behind applying co-clustering algorithm was to study the association relationship between known traffic and unknown traffic. In [13], Nascimento et al. presented a hybrid model to classify network traffic by using the Extreme Learning Machine (ELM), along with Feature Selection (FS) and Multi-objective Genetic Algorithms (MOGA). MOGA was used to optimize ELM classifier and to choose the best feature selection algorithm. They aimed to maximize two important metrics, *Flow Accuracy* and *Byte Accuracy*. They achieved an accuracy of 91% and 96% respectively in the experiments. In [14], Dong et al. built a hybrid system for network traffic classification by combining *port-based*, *payload-based*, and *machine learning based* methods. They used Bayesian methods and SVMs in their experiments and achieved over 95% average accuracy. In [15], Singh et al. collected real time Internet traffic using 2 seconds time interval in order to make real time IP traffic classification. They used 5 classifiers for test operations and Bayes Net was the most successful classifier with 88.13% of accuracy.

Some studies focused on the cross-site problem that cause reduced accuracy when the model trained in one network is used for testing in a different one. In [16], Ubik and Zejdl claimed that data traces from multiple networks, which have different speeds, produced more successful classifier rather than one network. They used 3 different networks, which have 100 Mbit, 1 Gbit, and 10 Gbit speed for data trace. They implemented C4.5 decision tree for the classifier. The success rate of classifier was 96.3% when it ran on the same network. However, if the classifier was run on different network speeds, its success rate decreased to 84.80%. In [17], Szabo et al. focused on efficient combination of clustering and classification algorithms to make the identification system impervious to network conditions. They stated that clustering is more robust to network parameter changes thus the accuracy drops less when the test set is measured in a different network than the training set compared to the classification algorithms. In their method, the result of clustering was considered as an additional feature and was added to the supervised feature set. Afterwards, this extended feature set was used in automatic supervised classification system

In this study, we classified popular end-user applications such as Facebook, Twitter, YouTube etc. by using machine learning based methods, whereas other studies focused on the classification of application categories.

III. System Design

The general outline of our proposed system is depicted in Figure 1. It consists of classical steps inherent to supervised learning methods.

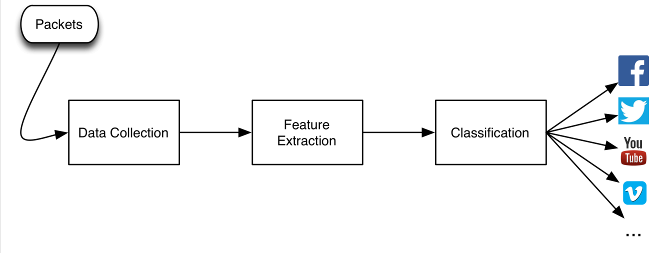

Figure 1. The general outline of our proposed system for application identification via network traffic classification

A. Data Collection

In our study, we used two datasets, namely the UNB ISCX Network Traffic dataset [18, 19] and our internal dataset. The UNB ISCX Network Traffic dataset has six general categories including *E-mail*, *Instant Messaging*, *Streaming*, *File Transfer*, *VoIP*, and *P2P*. Inside the categories, there are 14 different applications given in Table I. On the other hand, since some categories such as music services and social media do not exist in ISCX Dataset, we created our own dataset, namely internal dataset. Moreover, we collected data for some identical categories such as *File Transfer*, *IM*, *Video Streaming* etc. in order to cross compare the results. Totally, 13 applications were included into our internal dataset. The categories and applications in each dataset are given in Table I and Table II, respectively.

Since most popular applications such as *WhatsApp* and *Foursquare* do run on smartphones, we engaged a setup, which facilitated us to obtain network flows also from such mobile devices. To this end we turned a server equipped with wireless network interface into a special wireless access point functioning as a bridge between local clients and Internet. The same server also received a copy of wired network traffic via switch port mirroring feature. The server was setup to run Wireshark packet capturing software to capture and file all the network data passing in both directions. The specifics of the setup are given in Figure 2.

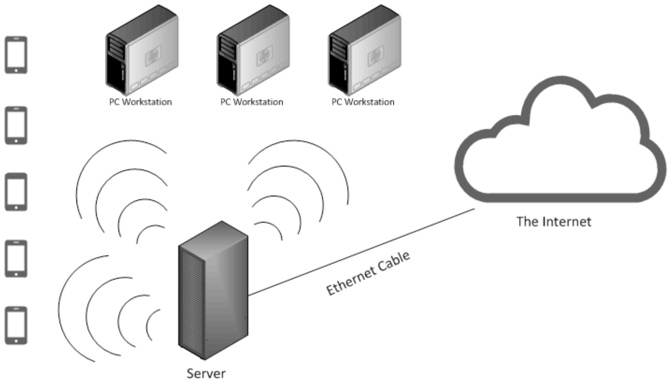

Figure 2. Packet capturing setup for mobile and desktop clients used to produce the internal dataset

TABLE I. Categories and Applications in the ISCX Dataset

Category	Application
File Transfer	Filezilla
Instant Messaging	AIM, Facebook Chat, Gmail Chat, Hangouts, ICQ, Skype
Mail	Mail
P2P	Torrent
Streaming	Vimeo, YouTube
VoIP	Facebook Audio, Hangouts Audio, Skype Audio

TABLE II. Categories and Applications in the Internal Dataset

Category	Application
File Transfer	Filezilla
Instant Messaging	Hangouts, Skype, WhatsApp
Mail	Mail
Music	Apple Music, Spotify
Social Media	Facebook, Twitter, Foursquare
Video	Vimeo, YouTube
Web Browsing	Various Web Surfing Flows

To create our internal dataset, several subjects exploited the applications using smartphones and wired clients in our setup. Just after the completion of each application usage, the captured packets were saved in pcap file format and labeled appropriately. For example, if the subject used Apple Music, the resulting pcap file was saved as “Apple-Music” with a date and time stamp appended under the Music folder. For ISCX Dataset, this arrangement had been already prepared.

Raw packet data cannot be directly given into the classification algorithms, so it must be first processed and transformed into instances. We considered each network flow as an instance. To define a network flow, we used the standard

5-tuple consisting of *source IP address, destination IP address, source port number, destination port number,* and *the protocol id*. To obtain the instances, we handled all the captured files of both datasets using a tool that we have developed in C. We initially extracted 111 features [3] from the datasets. Subsequently the instances for the two datasets were converted into the *arff* format, which is used by the Weka tool [20].

IV. Experimental Results

The number of instances for each application class is given in Table III. Using Weka, we evaluated four different classification algorithms including *J48*, *Random Forest*, *k-NN* (k is selected experimentally as 1), and *Bayes Net*. We applied 10-fold cross validation during our tests.

TABLE III. The number of instances for each dataset

Application	**ISCX Dataset Instance (Flow) Count**	**Internal Dataset Instance (Flow) Count**
Filezilla	145	36
Hangouts	98	47
Skype	166	575
WhatsApp	0	26
AIM	114	0
Facebook Chat	180	0
Gmail Chat	449	0
ICQ	45	0
Mail	536	115
Torrent	1045	0
AppleMusic	0	638
Spotify	0	218
Vimeo	433	135
Youtube	879	136
Facebook	0	238
Twitter	0	88
Foursquare	0	79
WebBrowsing	0	1417
Facebook Audio	9366	0
Hangouts Audio	1915	0
Skype Audio	91	0
Overall	**15462**	**3748**

The results of the classification algorithms for the ISCX dataset are given in Table IV, whereas the results for the internal dataset are presented in Table VI in the left side column labeled “Original” under each classification algorithm.

Evaluating both ISCX and Internal datasets showed that the overall accuracy varies between 85% and 94%. While Random Forest algorithm has given the best result for the internal dataset, k-NN algorithm is the most successful one for the ISCX dataset. However, apart from the overall accuracy, it is evident that the individual accuracies of some of the applications would vary yielding misclassification. For example, within the ISCX dataset *FileZilla*, *AIM*, *Hangouts*, *ICQ*, and *SkypeAudio* and within our internal dataset WhatsApp and Foursquare were the applications to be confused. To tackle this issue, we wanted to explore the option of feature selection. A process, which could not only eliminate correlated features and thus boosting the accuracy, but which could also significantly contribute to a reduction of the computational power requirements of the classification, has been applied.

TABLE IV. Accuracy results for the ISCX dataset using 111 features

	J48	**Random Forest**	**k-NN**	**Bayes Net**
Filezilla	50.30%	62.80%	66.90%	46.20%
AIM	59.60%	36.80%	59.60%	6.10%
FacebookChat	73.90%	68.30%	71.70%	58.90%
GmailChat	83.30%	77.70%	80.80%	53.90%
Hangouts	61.20%	27.60%	49.00%	19.40%
ICQ	33.30%	48.90%	48.90%	64.40%
Skype	78.90%	83.10%	75.90%	84.90%
Mail	91.20%	95.70%	94.00%	68.10%
Torrent	93.70%	93.80%	94.10%	83.20%
Vimeo	79.20%	82.70%	80.10%	61.40%
Youtube	87.40%	90.40%	86.30%	61.20%
FacebookAudio	98.30%	98.40%	98.40%	94.40%
HangoutsAudio	95.80%	93.80%	95.60%	87.80%
SkypeAudio	37.40%	42.90%	36.30%	45.10%
Overall	**93.84%**	**93.74%**	**93.94%**	**85.44%**

A. Feature Selection

We used *CfsSubsetEval* and *ChiSquaredAttributeEval* evaluators in Weka for the feature selection process. While *CfsSubsetEval* evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them, *ChiSquaredAttributeEval* assess the worth of an attribute by computing the value of the chi-squared statistic with respect to the class. The number of selected features along their names for each dataset is given in Table V.

TABLE V. Selected features for each dataset

Methods	**Selected Features for ISCX Dataset**	**Selected Features for Internal Dataset**
CfsSubset Eval	minPacketLengthF flowCountForConnection minPacketLengthB	maxPacketLengthF flowCountForConnection maxWindowSizeF density4B totalNumberOfRSTPacketsB maxWindowSizeB minWindowSizeB
ChiSquar edAttribut eEval	flowCountForConnection maxPacketLengthB numberOfBytesF maxPacketLengthF maxPacketSizeToStdDeviationB ratioOfBytesFandB avgPacketLengthB numberOfBytesToPacketCoun-tB numberOfBytesB minPacketLengthB avgPacketSizeToStdDeviationF avgPacketSizeToStdDeviationB	flowCountForConnection maxPacketLengthB numberOfBytesF maxPacketLengthF minWindowSizeB maxWindowSizeB minWindowSizeF maxIntervalPacketTimeB avgWindowSizeB maxIntervalPacketTimeF avgWindowSizeF totalNumberOfPUSHPackets F

The CfsSubsetEval evaluator left only three features selected for the ISCX dataset, and consequently running the classifiers with these newly selected features reduced the respective accuracies considerably. On the other hand, the *ChiSquaredAttributeEval* evaluator produced two sets of features both consisting of 12 features for both of the datasets and running the classifiers with these new sets of features showed that the accuracy remained the same for the ISCX dataset and increased by 2% for the internal dataset. The comparative results running the classifiers with the full set of 111 features versus the selected set of 12 features for each classifier is given in Table VI.

Subsequent to feature selection and running the classifiers with the selected features, we examined the confusion matrix to get a better idea about the misclassified instances. We decided to examine the confusion matrix for the Random Forest classifier (Table VII) for the internal dataset, as this classifier has the best accuracy for this dataset.

From Table VII it can be observed that the misclassified instances tend to be members of the same category. For example, in the case of WhatsApp the misclassified instances were classified as either Skype or Hangouts. A closer look at the packet traces for these applications showed us that they had very similar network flows for most of the time and thus yielding the misclassification.

TABLE VI. ACCURACY RESULT OF THE INTERNAL DATASET USING THE SELECTED FEATURES AND ORIGINAL 111 FEATURES

	J48		**Random Forest**		**k-NN**		**Bayes Net**	
Application	Original	Selected F.	Original	Selected F	Original	Selected F	Original	Selected F
Filezilla	86.10%	80.60%	83.30%	91.70%	86.10%	91.70%	72.20%	86.10%
Hangouts	74.50%	78.70%	70.20%	85.10%	85.10%	83.00%	46.80%	48.90%
Skype	95.30%	96.70%	96.90%	96.70%	95.30%	96.00%	72.00%	82.80%
WhatsApp	46.20%	38.50%	50.00%	50.00%	50.00%	46.20%	26.90%	26.90%
Mail	62.60%	74.80%	58.30%	67.80%	66.10%	71.30%	42.60%	44.30%
AppleMusic	91.80%	92.90%	93.70%	96.40%	93.30%	94.40%	74.60%	91.80%
Spotify	83.00%	84.90%	88.50%	90.40%	86.20%	89.00%	57.80%	75.20%
Vimeo	76.30%	85.20%	78.50%	88.10%	80.70%	83.00%	70.40%	75.60%
Youtube	76.50%	81.60%	83.80%	85.30%	83.10%	80.90%	74.30%	82.40%
Facebook	84.50%	86.60%	84.50%	89.50%	78.60%	82.40%	71.40%	78.60%
Twitter	68.20%	75.00%	69.30%	76.10%	67.00%	73.90%	59.10%	60.20%
Foursquare	55.70%	77.20%	67.10%	75.90%	54.40%	68.40%	15.20%	44.30%
WebBrowsing	91.90%	91.90%	97.40%	97.60%	89.80%	92.10%	75.50%	83.50%
Overall	**87.49%**	**89.45%**	**90.87%**	**92.99%**	**87.35%**	**89.42%**	**69.90%**	**80.20%**

TABLE VII. CONFUSION MATRIX OF RANDOM FOREST ALGORITHM FOR INTERNAL DATASET USING CHISQUAREDATTRIBUTEEVAL

Classified as ---->	a	b	c	d	e	f	g	h	i	j	k	l	m
a = Filezilla	33	0	0	0	0	0	0	0	0	1	0	0	2
b = Hangouts	0	40	5	1	0	0	0	0	0	0	0	1	0
c = Skype	0	2	556	4	1	2	1	0	0	1	1	4	3
d = WhatsApp	0	3	7	13	0	1	0	0	0	1	0	1	0
e = Mail	0	0	7	0	78	2	3	0	2	2	0	0	21
f = AppleMusic	0	1	3	0	1	615	1	1	1	1	0	1	13
g = Spotify	0	1	0	0	2	1	197	2	2	1	1	0	11
h = Vimeo	0	0	0	1	0	4	1	119	1	3	0	0	6
i = Youtube	0	0	0	0	4	1	3	1	116	0	0	2	9
j = Facebook	1	0	3	0	1	3	0	2	0	213	2	1	12
k = Twitter	0	0	2	0	0	1	5	1	0	1	67	0	11
l = Foursquare	1	2	3	0	0	2	1	3	2	0	1	60	4
m = WebBrowsing	0	0	1	0	1	14	6	2	2	6	0	2	1383

V. CONCLUSION AND FUTURE WORK

In this study, we explored machine-learning methods towards application identification via network traffic classification. In contrast to existing studies that used categories like *FTP*, *Instant Messaging*, etc. for classification, we considered popular end-user applications such as *Facebook*, *Twitter*, *Skype* etc. individually. We used two datasets, the UNB ISCX network traffic dataset and our internal dataset. Four classification algorithms, *J48*, *Random Forest*, *k-NN*, and *Bayes Net* were used as classifiers. With the complete set of 111 features, k-NN gave the best result for the ISCX Dataset with 93.94%, and Random Forest gave the best result for the internal dataset with 90.87%.

To increase the accuracy of the results for both datasets, and to reduce the computational complexity we applied feature selection. *ChiSquaredAttributeEval* evaluator gave the most satisfying result with a 90% reduction of 111 features into 12 selected features for both datasets. The feature reduction ends up with the same accuracy for the ISCX dataset and with a 2% increase for the internal dataset. A closer examination of the confusion matrix showed us that misclassification is occurring within the same categories rather than across categories. Therefore, as future work, we plan to implement a two-tier classification scheme. The first tier will classify the network flows into application categories similar to existing studies whereas the second tier will implement our proposed solution for a fine-grained individual application classification.

REFERENCES

[1] J. Kelner, A. Callado, C. K. Member, G. Szabó, and B. Péter, “A Survey on Internet Traffic Identification,” vol. 11, no. 3, pp. 37–52, 2009.

[2] Internet Assigned Numbers Authority (IANA), http://www.iana.org/assignments/service-names-port-numbers/service-names-port- numbers.xhtml, as of April 5, 2016.

[3] A. Moore, D. Zuev, and M. Crogan, “Discriminators for use in flow-based classification,” Queen Mary University of London, Department of Computer Science, August, 2005.

[4] S. Li and Y. Luo, “High performance flow feature extraction with multi-core processors,” *Proc. - 2010 IEEE Int. Conf. Networking, Archit. Storage, NAS 2010*, pp. 193–201, 2010.

[5] N. Williams, S. Zander, G. Armitage, Evaluating machine learning algorithms for automated network application identification, in: Center for Advanced Internet Architectures, CAIA, Technical Report 060410B, 2006.

[6] T. T. T. Nguyen and G. Armitage, “A survey of techniques for internet traffic classification using machine learning,” *IEEE Commun. Surv. Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[7] S. Kaoprakhon and V. Visoottiviseth, “Classification of audio and video traffic over HTTP protocol,” *2009 9th Int. Symp. Commun. Inf. Technol. Isc. 2009*, no. August, pp. 1534–1539, 2009.

[8] A. Callado, C. Kamienski, G. Szabo, B. P. Gero, J. Kelner, S. Fernandes, D. Sadok, “A Survey on Internet Traffic Identification”, *IEEE Commun. Surv. Tutorials*, vol. 11, no. 3, pp. 37–52, 2009.

[9] J. M. Wang, C. L. Qian, C. H. Che, and H. T. He, “Study on Process of Network Traffic Classification Using Machine Learning,” *The Fifth Annual ChinaGrid Conference, pp. 262-266*, 2010.

[10] S. Dong, D. Zhou, and W. Ding, “The Study of Network Traffic Identification Based on Machine Learning Algorithm,” *2012 Fourth International Conference on Computational Intelligence and Communication Networks*, pp. 205–208, 2012.

[11] N. F. Huang, G. Y. Jai, C. H. Chen, and H. C. Chao, “On the cloud-based network traffic classification and applications identification service,” *2012 International Conference on Selected Topics in Mobile and Wireless Networking, ICOST 2012*, pp. 36–41, 2012.

[12] W. Lu and L. Xue, “A Heuristic-Based Co-clustering Algorithm for the Internet Traffic Classification,” *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, no. 5, pp. 49–54, 2014.

[13] Z. Nascimento, D. Sadok, and J. Kelner, “Multi-Objective Optimization of a Hybrid Model for Network Traffic Classification by combining Machine Learning Techniques,” *International Joint Conference on Neural Networks (IJCNN)*, pp*. 2116-2122*, 2014.

[14] H. Dong, G. Sun, and D. Li, “A Hybrid Method for Network Traffic Classification,” *2nd International Conference on Measurement, Information and Control*, pp. 653–656, 2013.

[15] K. Singh, S. Agrawal, and B. S. Sohi, “A Near Real-time IP Traffic Classification Using Machine Learning,” *International Journal of Intelligent Systems and Applications*, vol. 5, no. 3, pp. 83–93, 2013.

[16] S. Ubik and P. Žejdl, “Evaluating application-layer classification using a machine learning technique over different high speed networks,” *Proceedings - 5th International Conference on Systems and Networks Communications, ICSNC 2010*, pp. 387–391, 2010.

[17] G. Szabo, J. Szule, Z. Turanyi, G. Pongracz “Multi-level Machine Learning Traffic Classification System,” *The Eleventh International Conference on Networks*, pp. 69–77, 2012.

[18] http://www.unb.ca/research/iscx/dataset/ISCX-network-traffic-VPN-dataset.html, as of April 5, 2016

[19] G. D. Gil, A. H. Lashkari, M. Mamun, A. A. Ghorbani, "Characterization of Encrypted and VPN Traffic Using Time-Related Features", In Proceedings of the 2nd International Conference on Information Systems Security and Privacy(ICISSP 2016) , pages 407-414, 2016.

[20] M.Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," ACM SIGKDD Explorations Newsletter 11, no. 1, pp. 10-18, 2009.

[21] W. De Donato, A. Pescapé, and A. Dainotti, “Traffic Identification Engine: An Open Platform for Traffic Classification,” IEEE Network, pp. 56–64, 2014.

[22] J. Zhang, Y. Xiang, Y. Wang, et.al., (2013), “Network Traffic Classification Using Correlation Information”, IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No. 1.

[23] J. Zhang, X. Chen, et.al., 2014, “Robust Network Traffic Classification”, IEEE/ACM Transactions On Networking, Vol. PP, No. 99.

[24] C. Tayşi, M.E. Karslıgil, et al. (2013), “Makine Öğrenmesi Tabanlı IP Trafik Sınıflandırılması”, Signal Processing and Communications Applications Conference (SIU), pp. 1-4.

[25] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and A. V. Vasilakos, “An effective network traffic classification method with unknown flow detection,” *IEEE Transactions on Network and Service Management*, vol. 10, no. 2, pp. 133–147, 2013.

[26] S. Alcock and R. Nelson. “Application Flow Control in YouTube Video Streams” SIGCOMM Comput. Commun. Rev., 41:24–30, April 2011.

[27] "E.800: Terms and definitions related to quality of service and network performance including dependability". *ITU-T Recommendation*. August 1994.

[28] http://thehackernews.com/2014/12/crash-your-friends-whatsapp-remotely_1.html, as of December 1, 2014.

[29] http://blog.trendmicro.com/trendlabs-security-intelligence/vulnerability-in-spotify-android-app-may-lead-to-phishing/, as of August 5, 2014.

[30] V. Paxon. Bro: A system for detecting network intruders in real-time. In *Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX, 1998.