Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 17(5): 4747–4772.
DOI: 10.3934/mbe.2020260
Received: 02 April 2020
Accepted: 27 June 2020
Published: 9 July 2020

Research article

Identifying VoIP traffic in VPN tunnel via Flow Spatio-Temporal Features

Faiz Ul Islam[1], **Guangjie Liu**[2,*] **and Weiwei Liu**[1]

[1] School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China

[2] School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China

* **Correspondence:** Email: gjieliu@gmail.com.

Abstract: The persistent emergence of new network applications, along with encrypted network communication, has make traffic analysis become a challenging issue in network management and cyberspace security. Currently, virtual private network (VPNs) has become one of the most popular encrypted communication services for bypassing censorship and guarantee remote access to geographically locked services. In this paper, a novel identification scheme of VoIP traffic tunneled through VPN is proposed. We employed a set of Flow Spatio-Temporal Features (FSTF) to six well-known classifiers, including decision trees, K-Nearest Neighbor (KNN), Bagging and Boosting via C4.5, and Multi-Layer perceptron (MLP). The overall accuracy, precision, sensitivity, and F-measure verify that the proposed scheme can effectively distinguish between the VoIP flows and Non-VoIP ones in VPN traffic.

Keywords: encrypted traffic; Flow Spatio-Temporal Features; machine learning; virtual private network (VPN); Voice over IP (VoIP)

1. Introduction

Network traffic comprised of data encapsulated in network packets belonging to a variety of Internet-based applications. With the profusion of encrypted Internet applications, network administrators always have the need to monitor some specific network protocols or applications [1]. Traffic identification is fundamental to network traffic management, which helps the network operators to analyze the network traffic and identify the specific applications and protocols. In addition, this topic has received increased attention due to application prioritization for network Quality of Service (QoS) and traffic engineering [2, 3], network management [4] and security protection [5].

With the increasing number of encrypted protocols and obfuscation tools, the accurate identification

of network traffic has become a significant challenge in current security industry [6]. The encryption protocols such as secure socket layer/transport layer security (SSL/TLS) are used to intertwine the network traffic flows and thus lose their unique characteristics. Hence the traditional TCP/UDP port numbers based and deep packet inspection (DPI) classification approaches no longer fulfill efficient recognition due to non-standard port and encryption [7]. To overcome the failure of these traditional network traffic classification approaches, the feature-based traffic classification methods show high accuracy in the accurate identification of encrypted network traffic and further characterization with specific applications and protocols.

Virtual private network (VPN) provide secure communication for remote users to access geographically and content-based locked services, which is a locked tunnel between the remote user device and the Internet, having the capability for avoiding the user traffic from spoofing, sniffing, and censorship. The identification of the VPN-tunneled traffic is a quite challenging task due to its packet-level encryption. VPNs are generally created on the application layer, network layer, or data link layer. The standard protocols used for the mentioned layers include SSL/TLS, IPSec, and L2TP [8]. Due to the implementation of these encryption protocols, VPN traffic analysis still remains many difficult issues. Classification of tunneled traffic depends on the ultimate purpose, such as categorized them according to the specific applications, (e.g., Facebook, YouTube, Skype, GTalk, Primus Softphone, QQ). Furthermore, characterization into specific application type (e.g., Chat, Streaming, File Transfer, Voice call) as many of these mentioned applications support multiple services. Still, the VPN traffic analysis is in the early stages and needs innovative identification techniques.

In the last few years, Voice over IP (VoIP) protocols grows to be the most important services for individuals and companies for making phone calls between VoIP end-users over the Internet due to its dramatic functionality over the traditional telephone network and cost-effectiveness. VoIP is also an important communication form for online meetings and education. VPN tunnels are used to ensure the security of sensitive communication over suspicious Internet infrastructure. The VoIP communication through VPN tunnels hides its content by applying encryption to circumvent firewalls and Network Address Translation (NAT) restrictions. Therefore, a competent classification engine is required to differentiate encrypted VPN traffic according to application type. It is an essential concern for enterprises to assure the appropriate consumption of bandwidth to client applications.

The followings are the main contributions of this paper:

(1) This paper aims to provide the Flow Spatio-Temporal Features (FSTFs) for distinguishing VoIP flows from Non-VoIP ones in VPN traffic. FSTFs are mainly composed of temporal components, because the rate-adaptive techniques widely are used in VoIP resolutions and products. Meanwhile, FSTFs are more suitable for traffic inspection near the enterprise network boundary where the flow temporal signals are more distinct than those captured at the inspection nodes far away from the VoIP users.

(2) The proposed identification method validated via six well-known machine learning algorithms including decision trees (C4.5, Random Forest), K-Nearest Neighbors, bagging and boosting (via C4.5), and neural network classifier (MLP). The algorithms applied to both 10-Fold Cross-Validation and training-testing technique. According to the consideration of the practical implementation efficiency demand, only traditional machine-learning-based classifers are used and tested in this paper.

The structure of the remaining paper is divided into several sections. The related work is given in Section 2. The main design of the proposed scheme is described in Section 3. This section defined the preprocessing steps such as dataset description, flow generation, feature set selection, and generation of final datasets. Furthermore, the learning algorithms and the accessing predictive ability used for the experimental results are explained briefly. Finally, the experimental results are evaluated and discussed at the end of this section. The conclusion is forecasted, and future work is discussed in Section 4.

2. Related works

In this section, we first give a brief review for the methods of network traffic classification. Next, the machine learning-based network traffic classification approaches are discussed from a historical view. Last, we also survey VoIP traffic classification methods that have been proposed so far.

2.1. Methods for network traffic classification

Existing network traffic classification methods can be categorized into three types: Port-based classification, payload-based classification, and statistic-based classification.

2.1.1. Port-based classification

In the early stage of network traffic classification, the port-based technique used to be the most popular and fastest. Port-based classification uses port number in the TCP/UDP header of the IP packet to classify the network traffic without any information storage. Some port numbers reserved for privileged services were designated as 'well-known' port numbers as defined by Internet Assigned Numbers Authority (IANA) [9]. This method has several drawbacks which makes it unreliable. Some applications have registered port numbers (e.g., SSH and SMTP uses port 22 and 25 respectively), however some modern applications associated with non-standard ports, or dynamically allocated port numbers. For instance, online gaming, peer to peer (P2P) applications use random port numbers. A P2P varies the ports, and the client might use TCP port 80 to generate traffic that would appear as HTTP. Due to these allocations of random port numbers to different applications makes the detection very hard to classify network traffic. However, this type of classification is still applicable for specific applications that use their default assigned port numbers, especially for the case where accuracy is not the point of interest (e.g., network traffic monitoring).

2.1.2. Payload-based traffic classification

Payload-based traffic classification generally looking for distinctive application signatures in the payload of an IP packet. Thus with knowledge of these signatures, it is easy to identify individual protocols [7]. Payload-based method inspects the entire packet payload against a set of known protocol signature to classify the packet flow. This technique has a high classification accuracy of approximately 100% for unencrypted traffic [10]. However, payload-based classification has some drawbacks. Firstly, this method fails to classify encrypted packets because the unique patterns, regular expressions, and strings are invisible in encrypted network flow. Secondly, as this method inspect each packet of the entire traffic flow to classify them, therefore the computational cost is much higher.

2.1.3. Statistical classification

In order to maintain client privacy, the accessibility to the payload of the transmitted IP packets is prevented by imposing some legal restrictions such as encrypting payload information. Thus statistical classification method introduced to overcome the challenges for encrypted traffic analysis by using payload-independent statistical network flow attributes that are unique for different applications [11]. This method relies on packet-based or flow-based parameters and does not need to access the packet payload information. Frequently statistical parameters used by researchers are packet-based features (e.g., packet length, packet intervals, packet directions, etc.) and flow-based features (e.g., flow packets per second, flow bytes per second, flow duration, inter-arrival time, etc.) [12]. Therefore, the advantage of this method is that they can be applied to encrypted or tunneled flow, as they have no concern with the actual content of the packet. Machine learning techniques are employed to use these unique statistical attributes to characterize the encrypted network traffic into different applications. In the past, many works have been done on the application of machine learning in the field of network traffic classification [13]. The machine learning techniques used for statistical classification are further categorized as an unsupervised, supervised, and semi-supervised learning technique.

Unsupervised learning technique infers hidden patterns from unlabeled network traffic flows and groups them into a set of clusters with similar properties. The distance measuring approaches define the similarity between two data points (e.g., Manhattan, and Euclidean distance). The k-means and DBSCAN clustering are the most frequently used unsupervised learning techniques. These techniques do not need any labeled instances and training phase. It can accomplish the classification task for similar network traffic categories generated by distinct protocols. Some of the contributed work in the field of network traffic classification based on unsupervised learning are [14–17].

Supervised learning technique used for the classification of the network traffic flows where the class labels are known. A set of pre-labeled instances with statistical features are grouped to build a classification model during the training phase. Further, an unseen network traffic flows are predicted by the trained model developed in the training phase. The ultimate goal of supervised learning is to learn a function that maps an input features to output class. Mathematically, it can be shown as [13]

$$f(m_i, n_j) = (m_1, n_1), (m_2, n_2), \ldots, (m_l, n_k). \tag{2.1}$$

the $f(m_i,\ n_j)$ is the dataset, where m_i are the input attributes corresponding i^{th} instances, and n_j is its output class labels. Wide range of the well-known supervised learning techniques such as Naïve Bayes, decision trees, Support Vector Machine (SVM), Genetic Programming (GP), Multi-Objective Genetic Algorithm (MOGA) [18–25] are applied for the identification of P2P traffic, VoIP services and especially encrypted traffic classification with the accuracy more than 90%.

Semi-supervised learning technique falls in between supervised and unsupervised learning. The dataset contains both a small amount of labeled and a large number of unlabeled flows that are fed into a clustering algorithm to build the classification model [26]. Many authors used semi-supervised learning models for the classification of different protocols and applications [26–28].

2.2. Machine learning based traffic classification

This section provides a brief overview of the recent contributions in the field of traffic classification using machine learning. Most of the recent researchers used machine learning as a key methodology

with the flow and packet-based features to replace traditional classification methods (port-based or payload-based) to classify encrypted traffic efficiently. Machine-learning-based classification is specialized in encrypted and tunneled traffic with acceptable computational complexity and accuracy respecting user's privacy. The statistic clustering, machine learning technique, and some other heuristic approaches have been applied to identify the network traffic and specific applications.

In the early 90's Paxson et al. [29, 30] led the foundation of statistical feature based traffic classification. Later on, McGregor et al. [31] have proposed a method based on the Expectation-Maximization (EM) algorithm which utilizes the flow features for flows clustering the traces into single and multiple transactions, bulk transfer, and interactive traffic. Moore et al. [32] proposed one of the first studies considering machine learning for traffic classification. Naïve Bayes estimator and a Bayesian neural network were employed to categorize network traffic flows into different applications, respectively. Nguyen et al. [13] reviewed the emerging research from 2004 to early 2007 in the field of IP traffic classification approach based on machine learning techniques. Erman et al. [33] applied a semi-supervised technique (k-means algorithm with Euclidean distance) as an internet traffic classifier using flow statistics. The TCP flows contain few type-known examples and many type-unknown examples that were successfully classified into a variety of applications including EMAIL, FTP, HTTP, P2P, and CHAT. Este et al. [18] extensively studied SVM for network traffic classification. The proposed method has been validated with the three traces from different locations, namely: UNIBS dataset, LBNL dataset, and CAIDA dataset. A simple optimization procedure was used to derive the ideal parameters for SVM and further applied for classification of TCP bi-directional flows into multiple applications such as HTTP, HTTPS, POP3, eDonkey, SMTP, BitTorrent, MSN, Gnutella, and FTP. Li et al. [34] used a set of flow features derived from the initial few packet headers to classify network traffic flows using C4.5 and Naïve Bayes. C4.5 shown better results for the classification of network traffic flows compared with the other classifiers in all cases. Bacquet et al. [35] identified encrypted traffic via unsupervised learning techniques: DBSCAN, EM, basic k-means, Semi-supervised k-means, and MOFA. NetMate toolset [36] is used for flow generation and feature extraction. MOGA shows the best result with the detection rate (DR) of 93.5% and a false positive rate (FPR) of 0.7%. Alshammari et al. [22] tested AdaBoost, C4.5, and GP to identify traffic traces of encrypted applications: SSH and Skype. They employed public traffic traces to distinguish encrypted applications traffic and non-encrypted applications traffic. Furthermore, Jun Zhuang et al. [37] proposed a new bag-of-flow (BoF) based traffic classification scheme to aggregate Naïve Bayes predictions of correlated flows generated by applications. Huang et al. [38] proposed an early identification method of different network application traffic based on L7 (application layer). Six machine learning algorithms were employed to classify 59 protocols from TCP/UDP flows. C4.5 tree algorithm attains better performance among the other schemes with the average overall accuracy of 92.88%.

Recently, the use of mobile messaging apps increased significantly because of many conveniences, such as sharing photos, texting each other, video chatting, booking tickets, paying bills, and shopping, etc. The widespread usage of mobile apps generates an enormous amount of mixed encrypted traffic every day. Therefore, network management and mobile companies need to classify the service usages of mobile applications (Apps). Fu et al. [39] developed a system, to classify traffic flow of famous Apps WhatsApp and WeChat into different services such as text, picture, voice note, stream video call, location sharing, and a short video. Features based on packet lengths and time delays were

employed to ensemble classifiers for classification of service usages in mobile messaging Apps. Mauro et al. [40] studied Android encrypted network traffic generated by mobile apps to investigate the user's actions. They studied user's actions for different apps such as Facebook, Gmail, Twitter, Tumbler, Dropbox, Google+, and Evernote, and classified them through supervised and unsupervised machine learning algorithms. Zhen et al. [41] presented a method, named Extended Labeled Data (ELD), to identify unknown mobile traffic with the flow and byte accuracy more than 96%. Giuseppe et al. [42] used a multi-classification system to enhance the classification performance of encrypted traffic generated by mobile apps. Different classifier fusion techniques were used to classify a dataset composed of 607 real-traffic traces generated from 49 mobile apps.

Furthermore, Yu et al. [2] proposed a novel scheme to classify network video traffic using effective statistical features. Hierarchical KNN classifier employed to classify the captured internet video traffic consist of six different applications (ASD, AHD, QQ, HTTP-download, Xunlei, and Sopcast). Klenilmar et al. [43] presented an identification scheme for video streaming traffic based on Naïve Bayes algorithm, which was able to classify YouTube streaming video, Netflix streaming video, and background traffic with an average accuracy of 98.88%. Antônio et al. [44] used packet-based features to classify encrypted IoT traffic and characterize the behavior of IoT devices. The evaluation included five popular classifiers KNN, RF, DT, SVM, and MV applied to classify IoT devices from the non-IoT device. These mentioned works depicted that it was promising to use statistical features based on flows and packets for efficient classification of network traffic and applications. More studies are still required to identify encrypted traffic within VPN tunnels.

2.3. VoIP traffic classification

In this section, we highlighted some of the related work related to methods, techniques, and tools to analyze VoIP traffic. Traffic classification of VoIP services has been an enormous interest by several recent data analysts.

Some of the authors proposed signature-based identification schemes to identify the VoIP traffic [45, 46]. Moreover, due to encryption and tunneling of VoIP services, it is impossible to do payload-based analysis. The machine learning algorithms, statistic clustering, and some other heuristic approaches have been presented to identify the VoIP traffic.

Toshiya et al. [47] presented one of the earlier studies of VoIP traffic identification. The authors considered the flow-level features were more suitable for real-time VoIP application identification. Five different applications were studied based on the packet size and inter-arrival time and classified them into voice, video and file-sharing applications.

Skype is one of the most popular peer-to-peer (P2P) VoIP service provider, developed in 2002. Skype comprises a robust encryption mechanism, making it difficult to be analyzed which attracted the attention of the research community. Some of the work related to Skype traffic analysis are listed here [48–51]. Bonfiglio et al. [48] tested two approaches for the identification of Skype voice calls. The first approach used Pearson's Chi-Square testing to explore the statistical characteristics of the traffic payload, which analyzed the randomness of the message content generated by cipher during encryption. The second approach applied the Naïve Bayesian classifier to characterize Skype traffic. The combination of these two approaches for detecting Skype voice traffic showed negligible FPR and FNR. Alshammari et al. [49] employed flow-based features to classify VoIP flows in the particular Skype and GTalk traffic. Three popular machine learning approaches, namely, C4.5, AdaBoost, and

GP, were evaluated to accomplish the classification task. C4.5 provided an optimal result with the DR of 99% approximately and less than 1% FPR. Ibrahim et al. [50] studied ten machine learning algorithms to classify captured traffic successfully into Skype and OnlineTV traffic. Davide et al. [51] proposed a joint statistical and signature-based approach to classify Skype traffic and distinguish file transfer, voice, and video calls.

Moreover, Khan et al. [52] proposed a method to recognize a perpetrator from the encrypted VoIP conversation. The variable packet-length-based features are taken into account to identify the speaker from an encrypted conversation in case of the variable bit speech encoding mechanism. Taner et al. [53] proposed one of the earlier studies in the classification of VoIP traffic in the VPN tunnel. The authors proposed a simple technique to identify VoIP traffic in the IPSec tunnel for improving VoIP QoS. Li et al. [54] proposed a method based on the host behavior estimation and the flow statistical analysis to identify the VoIP traffic at the transport layer. The port numbers and IP addresses were tested as the host behavior while the inter-packet arrival time and entropy value (to model the packet size) were selected as the flow features to make identification. Express Talk, Tom-Skype, QQ voice and video, and windows live messenger (MSN) voice and video were selected as typical VoIP applications, and web-browser, multimedia applications, online gaming, and file-sharing applications were considered as Non-VoIP applications to accomplish the identification evaluation. Alshammari et al. [55] tested the C4.5, GP and Adaboost classifiers for encrypted VoIP traffic identification. They used the NetMate toolset [36] to generate flow from captured traffic traces and extract flow-based attributes. Two different experiments were studied. In the first experiment, the captured traffic traces were classified into Skype and non-Skype traffic. In the second experiment, the traffic traces were separately labeled as VoIP (GTalk, Skype, and Primus softphone) applications and Non-VoIP applications. Qin et al. [56] developed a traffic identification system based on the packet size distribution of the first few packets to divide the network traffic into VoIP and P2P applications. And further, give the specific application type, such as Skype. Recently, Mazhar et al. [57] proposed a method to detect the voice call flows from encrypted and tunneled traffic based on statistical features. The proposed scheme is with the real-time VoIP calls detection ability using 6 seconds captured traffic after the call initiation. The proposed method achieved a TPR of 97.54% and FPR of 0.00015% .

Most of the researchers focused on the classification of VoIP applications (Skype, YouTube, Facebook, GTalk, etc.) and further into specific application types (Voice calls, File Transfer, P2P, Streaming, Chat, etc.). The enormous growth of encrypted and tunneled VoIP traffic opens the new research area for the internet community. There are still quite a few studies on the VPN-tunneled VoIP traffic analysis. To improve the quality, block the forbidden traffic, to prevent the illegal use of network resources and characterize bandwidth used by VPN tunnels for different applications, we need more accurate solutions for efficient classification, identification, and characterization of VPN-tunneled VoIP traffic.

3. The proposed scheme

The proposed scheme aims to identify VoIP traffic from Non-VoIP traffic in the VPN tunnel accurately. The general framework of the proposed scheme is given in Figure 1. The key idea we leveraged is divided into five discrete phases. In the first phase, we extract the captured VPN traffic (.pcap files) from the mixed traffic traces containing both VPN and Non-VPN traffic traces. In the

second phase, traffic flows are generated based on three different flow latency periods (FLPs). In the third phase, data cleaning, feature selection, and labeling of the flows are described. In the fourth phase, three exclusive datasets are generated based on 15, 60 and 120 sec FLPs with unique flow instances. In the last phase, different classifiers are employed to distinguish VPN-VoIP traffic from VPN-Non_VoIP traffic. The refined datasets obtained after several preprocessing steps are presented in Section 3.1. Section 3.2 describes the machine learning algorithms employed for classification in the proposed work. In Section 3.3, we explained the performance assessment metrics to evaluate the efficiency of the scheme. Furthermore, Section 3.4 summarized the experimental results.

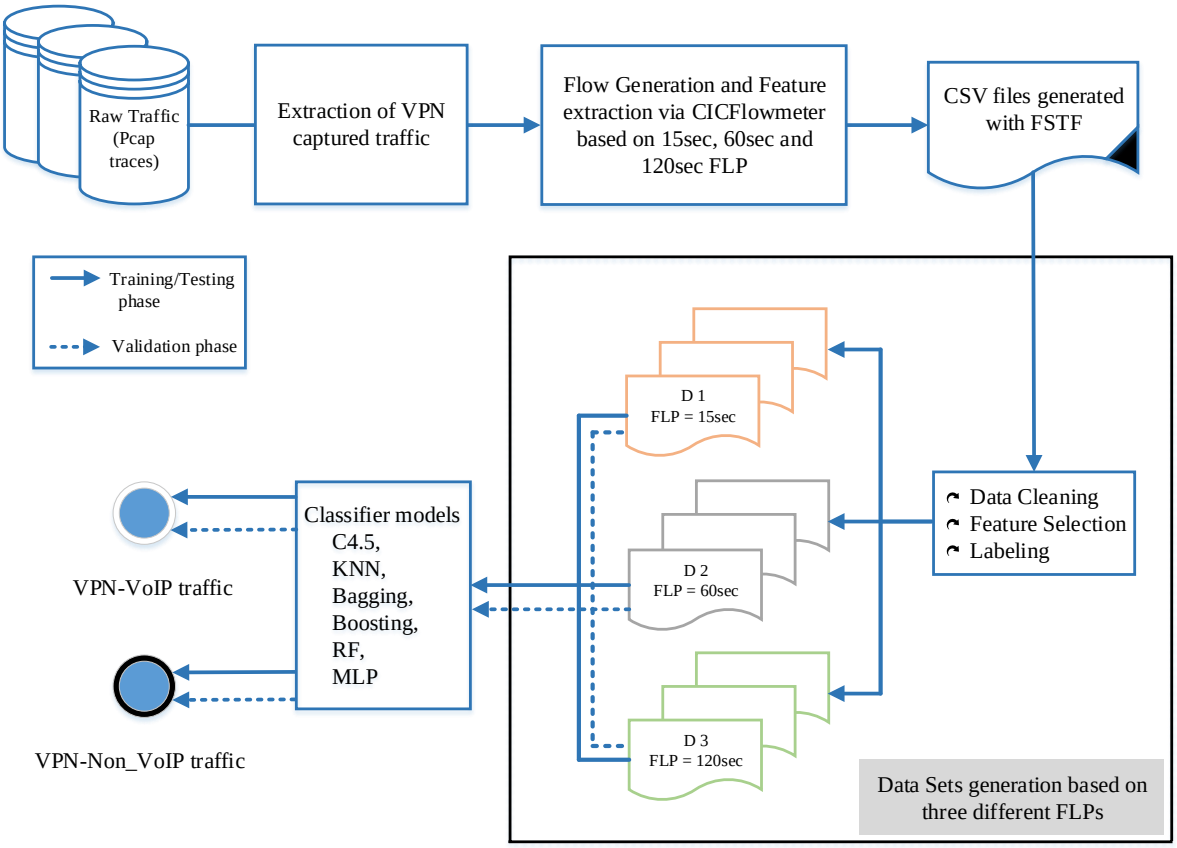

Figure 1. Flow chart of the proposed scheme for characterization of encrypted VPN traffic.

3.1. Preprocessing

3.1.1. Dataset description

The VPN-Non VPN dataset published by CIC, the University of New Brunswick, is employed to validate the aforementioned problem [58]. The extensive dataset consists of encrypted and VPN raw traffic traces generated by VoIP, Email, File Transfer, P2P, Streaming, Chat, and Web browsing (seven traffic traces for regular encrypted traffic and seven traffic traces for VPN traffic) in pcap format with particular application labels. However, the valid dataset has the actual number of 12 application classes because some files are related to two different classes, which can be considered in a single category at the same time. Wireshark and tcpdump are used to capture the traffic generated by most popular

applications which yield diversity in service types. We choose only the VPN tunneled traffic traces files (about 2.3 GB) for our interest to accomplish the classification task. Furthermore, we divided all the raw traffic types into two categories, VPN-VoIP and VPN-Non_VoIP. The detailed content type of these two classes is listed in Table 1. The voice calls (for 1-hour duration) traffic flow generated by Skype, VoIPbuster, Hangouts, and Facebook are grouped into VPN-VoIP traffic. On the other hand, traffic traces collected from P2P, Chat, Email, Streaming, and File Transfer are considered to be VPN-Non_VoIP traffic.

Table 1. List of captured VPN traffic.

Traffic Labels	*Content Type*	
VPN-VoIP	Skype, VoIPbuster, Hangouts and Facebook voice calls for the duration of 1hour	
VPN-Non_VoIP	P2P	μTorrent and Bittorent
	Chat	AIM, Skype, Facebook, Hangouts, ICQ
	Email	SMPTS, IMAPS, POP3S
	Streaming	Netflix, Vimeo, YouTube and Spotify
	File Transfer	FTPS, SFTP and Skype

3.1.2. Flow generation

For computing the flows and features, we use CICFlowmeter, an open-source java-based application [59] as the network flow generator. The input of CICFlowmeter is raw traffic traces captured in pcap format, and it will generate bidirectional flows based on forward (src to dst) and backward (dst to src) directions. The output of CICFlowmeter is the Comma Separated Values (CSV) file, where each flow is defined by Flow ID, consists of five parameters, i.e., Src IP, Dst IP, Src port, Dst port, and Protocol (TCP or UDP) with 76 statistical traffic features. The duration of the FLP can be controlled and adjusted by an individual in the source code. All the captured VPN raw traffic traces mentioned in Table 1 are aggregated into traffic flows by using the java source code of CICFlowmeter with 15, 60 and 120 sec FLP to get the corresponding CSV files for both classes. In order to classify the traffic flows according to Table 1, we labeled the corresponding obtained instances according to the source applications: Streaming, P2P, Chat, Email, VoIP (Skype, Voipbuster, hangouts, and Facebook voice calls) and File Transfer. All the flows obtained from P2P, Chat, Email, Streaming, and File Transfer are labeled as VPN-Non_VoIP class, and the flows obtained from the Skype, Voipbuster, hangouts, and Facebook voice calls traces are labeled as VPN-VoIP class. In the end, to get a refined data set, we removed all the duplicate flows.

3.1.3. Flow Spatio-Temporal Features selection

According to the 76 statistical features, we can make the Flow Spatio-Temopral Features selection. Firstly, we remove all the features with null values because features with zero values do not affect the classification task. Secondly, some distinct features are considered as ineffective because of zero values for more than 70% of flows. We only considered that statistical feature which has dispersed and majority non-zero values. A minimum number of features are selected to reduce computational time and complexity. In the past, most work [22, 49, 55] used NetMate toolset [36], which generates flows with the maximum number of 22 distinct flow features. CICFlowmeter produces a wide variety

of features, and also the duration of the flow is changeable, which shows better performance in contrast to NetMate toolset. The selected effective feature set for classification is listed in Table 2.

Table 2. Flow Spatio-Temporal Features set.

	Abbreviation	*Description*
1	$feduration$	Duration of the flow in Microsecond
2	max_fpktl	Maximum size of packet in forward direction
3	std_fpktl	Standard deviation size of packet in forward direction
4	max_bpktl	Maximum size of packet in backward direction
5	std_bpktl	Standard deviation size of packet in backward direction
6	tot_fhlen	Total bytes used for headers in the forward direction
7	tot_bhlen	Total bytes used for headers in the backward direction
8	$init\_win\_bytes$ $\_backward$	The total number of bytes sent in initial window in the backward direction
9	$fpkts\_persec$	Number of flow packets per second
10	$min\_flowiat$	Minimum inter-arrival time of packet
11	tot_fiat	Total time between two packets sent in the forward direction
12	tot_biat	Total time between two packets sent in the backward direction
13	min_biat	Minimum time between two packets sent in the backward direction
14	max_active	Maximum time a flow was active before becoming idle
15	min_active	Minimum time a flow was active before becoming idle
16	max_idle	Maximum time a flow was idle before becoming active
17	min_idle	Minimum time a flow was idle before becoming active

3.1.4. Generation of datasets

After the flow generation, labeling, and feature extraction, we obtain a reasonable amount of instances for each class listed in Table 3. We get three distinct data sets (D1, D2 and D3), as shown in Figure 1 based on the duration of the flow. All the datasets consist of the entry with 17-dimension FSTFs and the class label (i.e., VPN-VoIP and VPN-Non_VoIP). Furthermore, the experiments are based on two different scenarios. In Scenario 1, Cross-Validation with 10-Folds is implemented to test all the three datasets. In Scenario 2, the datasets are divided into training (X^{train}) and testing (X^{test}) subsets. The X^{train} contains 70% of random instances, while the X^{test} contains the remaining 30% instances. The X^{test} comprises of unseen data which shows the generalization of the algorithm. The details of X^{train} and X^{test} datasets are given in Table 3 according to different FLP.

Table 3. Details of testing and training datasets.

Datasets	*VPN-VoIP and VPN-Non_VoIP instances*		
(based on FLP)	(X^{train})	(X^{test})	(X^{Total})
D1	16124	7212	23336
D2	12102	5326	17428
D3	10799	4905	15704

3.2. Machine learning algorithms

According to the consideration of the practical implementation efficiency demand, instead of deep learning algorithms, only traditional machine-learning-based classifers are used and tested in this paper. Several typical machine learning algorithms are considered, they are decision trees (C4.5, RF), KNN, Bagging (via C4.5), Boosting (via C4.5), and MLP. This section provides a brief overview of them. The literature indicates the superior performance of these machine learning algorithms in the field of network traffic classification [22, 34, 38, 49, 55].

3.2.1. Decision trees

Decision trees are a famous supervised learning algorithm. It follows the divide and conquer strategy to classify input data. The decision tree structure is composed of decision nodes and terminal leaves, which represent the attributes and the final class or label, respectively. We used two popular decision trees based algorithms C4.5 and RF in the proposed work.

C4.5: C4.5 is a well-known classification tree algorithm that can perform both regression and classification tasks, developed by Ross Quinlan [60]. It is a successor to Iterative Dichotomiser (ID3) invented by Ross Quinlan. C4.5 uses training instances to construct a decision tree with the concept of information entropy evaluation function [61].

$$E(X) = -\sum_{m=1}^{N} \left\{ \left[{}^{freq(Y_m,X)}/_{|X|} \right] \log_2 \left[{}^{freq(Y_m,X)}/_{|X|} \right] \right\} \tag{3.1}$$

where X represent the training set. ID3 uses gains while the C4.5 relies on gain ratios to avoid overfitting as given below [61].

$$Gain\ Ratio(A) = Gain(A)/Splitinfo(A) \tag{3.2}$$

where A represent the test attribute and the $Splitinfo(A)$ which in turn is defined as [61]

$$Splitinfo(A) = -\frac{1}{2}\sum_{m=1}^{L} \left[\frac{|X_m|}{|X|} \log_2 \frac{|X_m|}{|X|} + \left(1 - \frac{|X_m|}{|X|}\right) \log_2 \left(1 - \frac{|X_m|}{|X|}\right) \right] \tag{3.3}$$

The attribute with large Gain Ratio is considered as the root of the classification tree.

Random Forest (RF): Random Forest (RF) is another supervised classification algorithm based on ensembles of decision trees proposed by Breiman [62]. RF makes the final prediction on the majority-voting concept visualized in Figure 2.

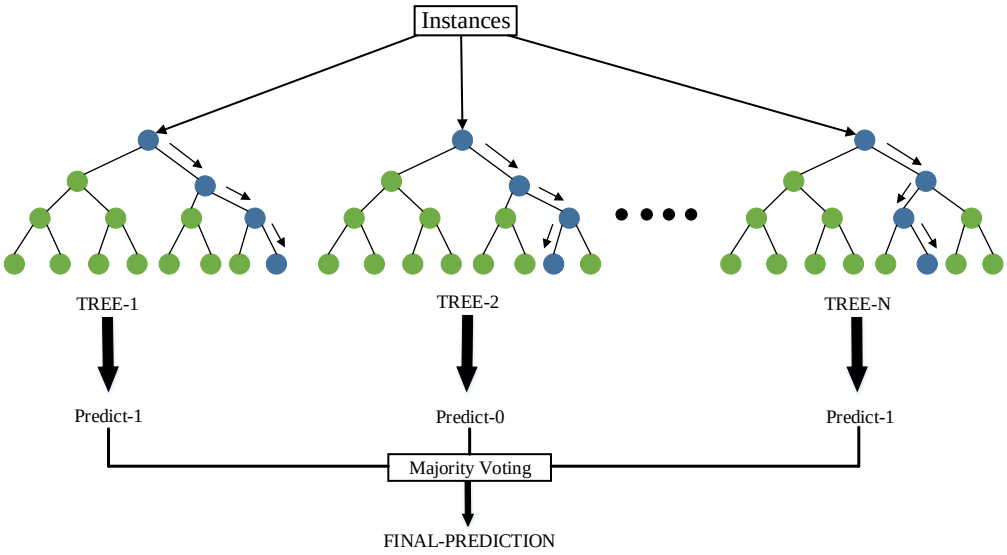

Figure 2. Visualization of Random Forest model.

3.2.2. K-Nearest Neighbor (KNN)

KNN is a supervised machine learning technique used for regression and classification problems. It classifies testing instance based on the k closest samples (where k = positive integer) from the training set. And a testing instance is categorized by calculating the distance to the nearest training case. The distance metric defines the similarity between two data points. Euclidean, Manhattan, and Minkowski distances are the used cases. Mostly used distance metrics are listed in Table 4. Mainly, two factors affect the performance of the KNN classifier, value of k, and the selection of distance metric.

Table 4. Types of Distance metrics used by KNN.

Euclidean distance	$D(y, y') = \sqrt{(y_1 - y_1')^2 + (y_2 - y_2')^2 + ... + (y_n - y_n')^2}$
Manhattan distance	$D(y, y') = \sum_{m=1}^{K} \lvert y_m - y_m' \rvert$
Minkowski distance	$D(y, y') = \sqrt[b]{\left(\sum_{m=1}^{K} \lvert y_m - y_m' \rvert^b\right)}$

In the proposed scheme, the k value is 1, and the Manhattan distance metric is selected for better performance.

3.2.3. Boosting

Boosting is an ensemble machine learning technique proposed by Freund to train multiple weak models sequentially and estimate a set of weights to produce a strong estimator [63]. Adaptive Boosting

(AdaBoost) in one of the most influential boosting algorithm. It uses a set of base classifiers during the training process and evaluates the weights for the instances for the final prediction. The weights of the correctly classified instances remain the same and the weights of the incorrectly classified instances increases. The normalized weights of all the instances are further employed for the next classifier. The final classification is aggregated by weighted based classifiers.

3.2.4. Bagging

Bagging (Bootstrap aggregating) is proposed by Breiman [64]. In contrast to boosting, multiple base classifiers are trained in parallel. Each classifier is trained through tiny alteration in the training dataset (a.k.a bootstrap sampling). The majority voting over the class labels decides the final prediction. In this study, the ensemble classifier consists of C4.5 as a base classifier.

3.2.5. Multi-Layer perceptron (MLP)

MLP is a kind of feedforward artificial neural network, consists of a minimum of three layers (input, hidden, and output layer), as shown in Figure 3. Mathematically, it can be express as

$$F(n) = Y(x^{(2)} + W^{(2)}(s(x^{(1)} + W^{(1)}n))) \tag{3.4}$$

where $x^{(1)}$, $x^{(2)}$ are bias vectors; $W^{(1)}$, $W^{(2)}$ are weight matrices; Y and s are the activation functions. Each neuron work as a summation unit and an activation function.

The number of hidden layers increases the complexity of the model. We selected nine hidden layers for the used case.

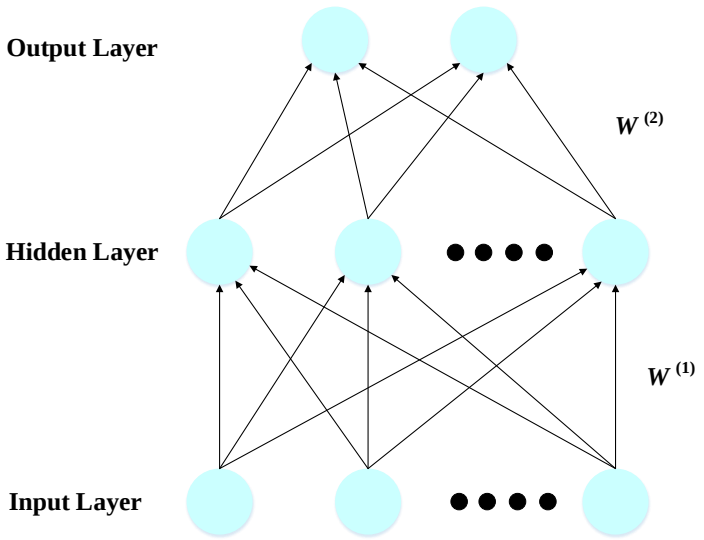

Figure 3. An MLP with single hidden layer.

3.3. Assessing predictive ability

The four basic parameters which are employed to compute the performance assessment metrics are [43]:

- True Positive (TP): When the predicted and actual instances are YES.
- False Positive (FP): When the predicted instance is YES and the actual instance is NO.
- True Negative (TN): When the predicted and actual instance are NO.
- False Negative (FN): When the predicted instance is NO and the actual instance is YES.

The number of performance assessment metrics are utilized to assess the predictive power of the machine learning algorithms. Four performance assessment metrics. i.e., Precision (Pr) or Positive Predictive Value (PPV), Sensitivity (Sen) or Recall (Rc), Accuracy (Acc), and F-measure (F-m) are utilized to assess the performance of the proposed algorithms, which are defined as follows [43].

3.3.1. Precision (Pr)

Pr represents the classifier correctness and also known as Positive Predictive Value (PPV). It is the number of True Positive results divided by the total number of positive class results predicted by the classifier, defined as:

$$Pr = \frac{TP}{(TP + FP)} \tag{3.5}$$

3.3.2. Sensitivity (Sen)

Sen reflects the completeness rate of the classifier. It is the ratio of True Positive results to the True Positive and False Negatives results, as defined below:

$$Sen = \frac{TP}{(TP + FN)} \tag{3.6}$$

3.3.3. Accuracy (Acc)

Acc is the ratio of the sum of all correctly classified flows to the sum of all the traffic flows tested.

$$Acc = \frac{\sum T_i}{\#Samples} \times 100\% \tag{3.7}$$

where T_i denotes all correctly classified flows, and $\#Samples$ denotes the sum of all the traffic flows exists in the dataset.

3.3.4. F-measure (F-m)

$F - m$ is the harmonic mean of Sen and Pr, which conveys the balance weight between Sen and the Pr. The value near to 1 is the best result, $F - m$ can be calculated as:

$$F - m = 2 \times \frac{\text{Pr} \times \text{Rc}}{\text{Pr} + \text{Rc}} \qquad 0 \leq F - m \leq 1 \tag{3.8}$$

3.4. Experiment setup

3.4.1. Environment

This work runs on the computer with a 2.4 GHz Intel Core-i3 CPU, a 6.0 GB of Random Access Memory (RAM), and the operating system Windows 7. All the pcap traces are processed through CICFlowmeter, which is written in java language to get the CSV files. The FLP adjusted accordingly in the script. WEKA platform is used for further classification, and evaluation indexes, which are discussed in Section 3.3 are employed to examine the predictive ability of the proposed scheme.

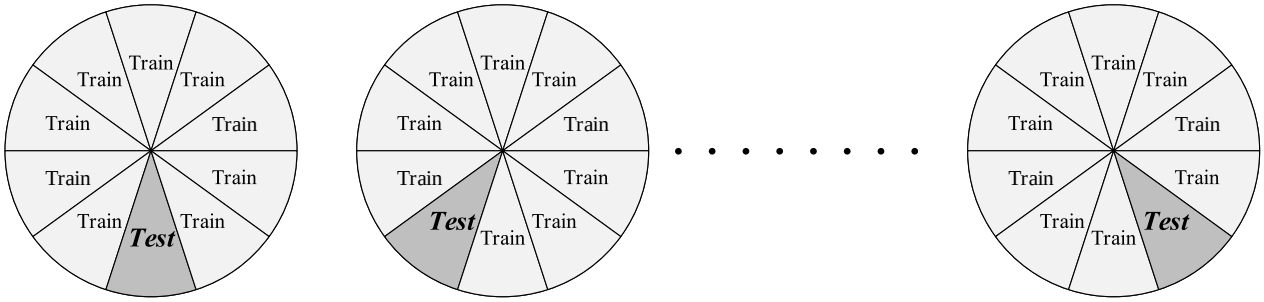

Figure 4. K-Fold Cross-Validation (k = 10).

3.4.2. Identification results

In this section, a series of comparative experiments are performed to comprehensively highlight the effectiveness of the proposed FSTF identification method. The proposed identification process of encrypted VPN traffic comprises of two major stages. One is the flow generation followed by dataset generation based on FSTF set with different FLPs. And the other is to perform the experiments according to the developed models to characterize the VPN traffic into VoIP and Non-VoIP classes. We evaluated the performance of each of the six learning algorithms (decision trees (C4.5, RF), KNN, Bagging (via C4.5), Boosting (via C4.5), and MLP) discussed in Section 3.2 for identification of VoIP services through VPN tunnel to validate the effectiveness of the FSTF. In order to evaluate the performance of the developed system, two types of experiments have been implemented: a) K-Fold Cross-Validation, and b) training-testing scenario. The FSTF set listed in Table 2 outperformed to classify the three datasets based on FLP with regard to the four evaluation metrics Pr, Sen, $F-m$, and Acc described earlier.

3.5. K-Fold Cross-Validation

K-Fold Cross-Validation is a resampling procedure that includes the random shuffling of the dataset, followed by splitting the dataset into k number of groups. In each run, one group is holded for testing and the remaining groups is used for training the model. The process continues until the last group used as the testing data, as described in Figure 4. During the first phase of experiments, the K-Fold Cross-Validation is employed to classify the VPN traffic into two categories, i.e., VoIP and Non-VoIP

traffic flows. In our experiments, we selected $k = 10$ for all the six learning algorithms.

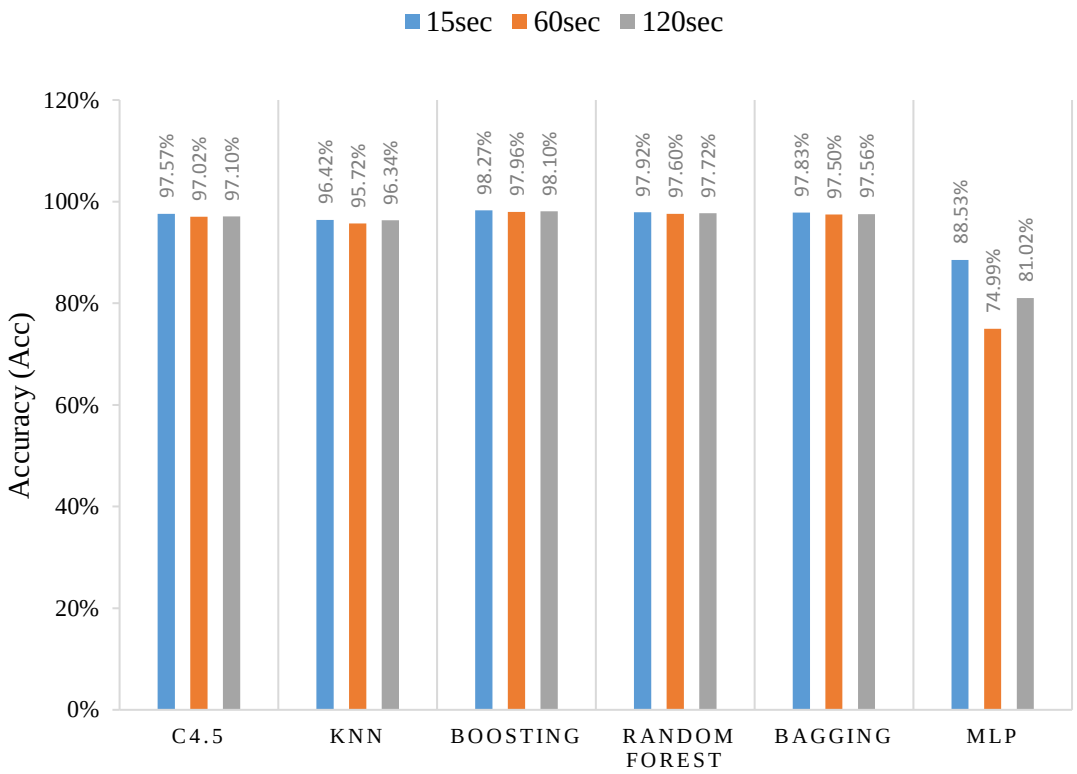

Figure 5. Accuracy of the used learning algorithms to identify VPN traffic (k = 10).

Figure 5 lists the accuracy level achieved by all machine learning algorithms based on FSTF set for three different datasets generated with 15, 60 and 120 sec FLP. It is evident that we obtained the best results for 15 sec of FLP throughout the six algorithms. All the classification algorithms show an accuracy level of more than 95.72% except MLP. In contrast to these results, MLP shows average performance with the highest accuracy of 88.53% and a minimum accuracy of 74.99%. The results look very promising for better classification of traffic through a VPN tunnel. All the machine learning algorithms achieved better accuracy. During experiments, the decision trees, KNN, Boosting and Bagging have almost similar performance results. The VoIP traffic was detected with the highest accuracy of 98.27% obtained by applying boosting (Adaboost via C4.5) with 15 sec FLP. The experimental results depict that the features listed in Table 2, along with the shorter FLP (15 sec in this case) are the best options to identify and classify the VoIP traffic in a mixed VPN traffic flows.

3.6. Training-testing scenario

In this section of the paper, we illustrate the identification method for VoIP traffic traces using trained models and later on tested their performance with unseen testing traffic flows using the same FSTF set. The experimental setup for this scenario is portrayed in Figure 6, where all the six models are initially trained with the X^{train} traffic flows and then tested with the unseen X^{test} traffic flows with FSTF set given in Table 2. Details of the training and testing datasets (D1, D2 and D3) are given in Table

3. The generalization of the system is examined by four performance assessment metrics explained in Section 3.3.

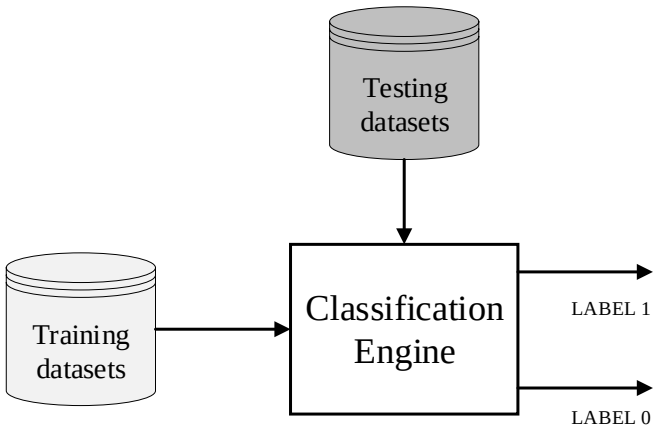

Figure 6. Illustration of training a model and testing with unseen data.

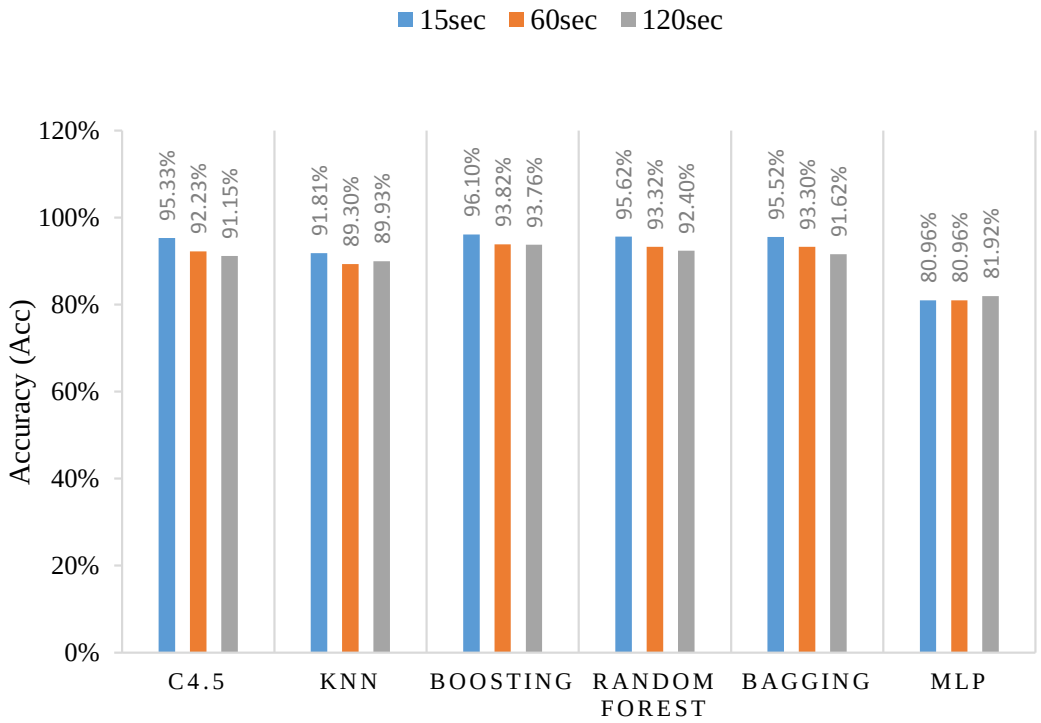

Figure 7. Accuracy of the used learning algorithms to identify VPN traffic using training and testing scenario.

Figure 7 shows classification accuracy obtained by testing a trained model with unseen VPN traffic flows. From Figure 7, it can be seen that all the six algorithms have consistent performance and show

the accuracy level of more than 80.96%. Among these trained classifier models, the boosting model shows attractive results with the highest accuracy of 96.1% for FLP of 15 sec. Repeatedly, the shorter FLP enhanced the accuracy level and achieved better identification results. As a result, the boosting technique offered significantly better performance as an identification engine in both scenarios for all FLP values.

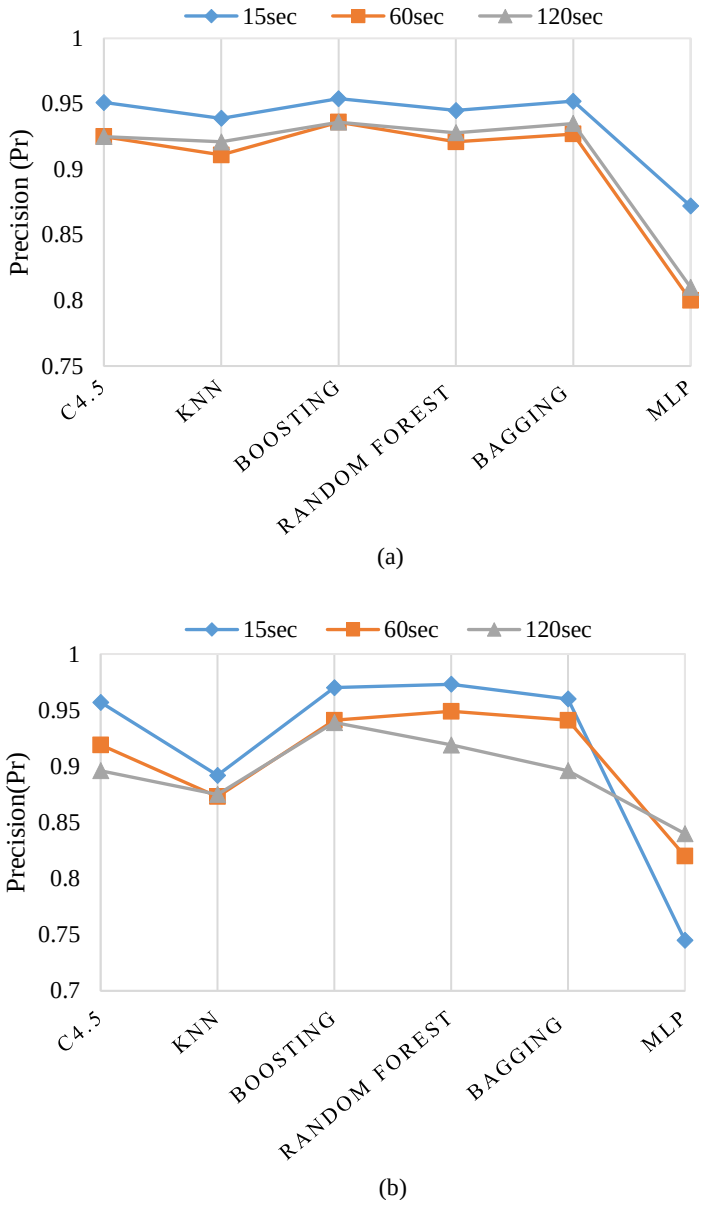

Figure 8. Precision (Pr) of (a) VPN-Non_VoIP and (b) VPN-VoIP traffic flows.

The Pr, Sen, and $F-m$ have been calculated to assess the results further. Except for MLP, all the classifier models presents attractive identification results. Here we will only explain the performance metrics obtained from a trained model based on the boosting technique. Figure 8 illustrates the Pr

of the VPN traffic traces, which are classified into VoIP and Non-VoIP traffic flows. The optimal Pr values for Non-VoIP and VoIP traffic flows are 0.954 and 0.97 respectively for 15 sec of FLP. On the other hand, when we increase the FLP to 120 sec, the Pr decreased to 0.936 and 0.939, respectively. Figure 9 exhibits similar behavior, as the Sen of boosting via C4.5 classifier for Non-VoIP and VoIP traffic flows have been decreased from 0.978, 0.937 to 0.95, and 0.923, respectively. Finally, the $F-m$ count shown in Figure 10 for Non-VoIP and VoIP traffic flows are 0.966 and 0.954 for 15 sec FLP, while a slight decrease detected for 120 sec, which is 0.943 and 0.931 respectively.

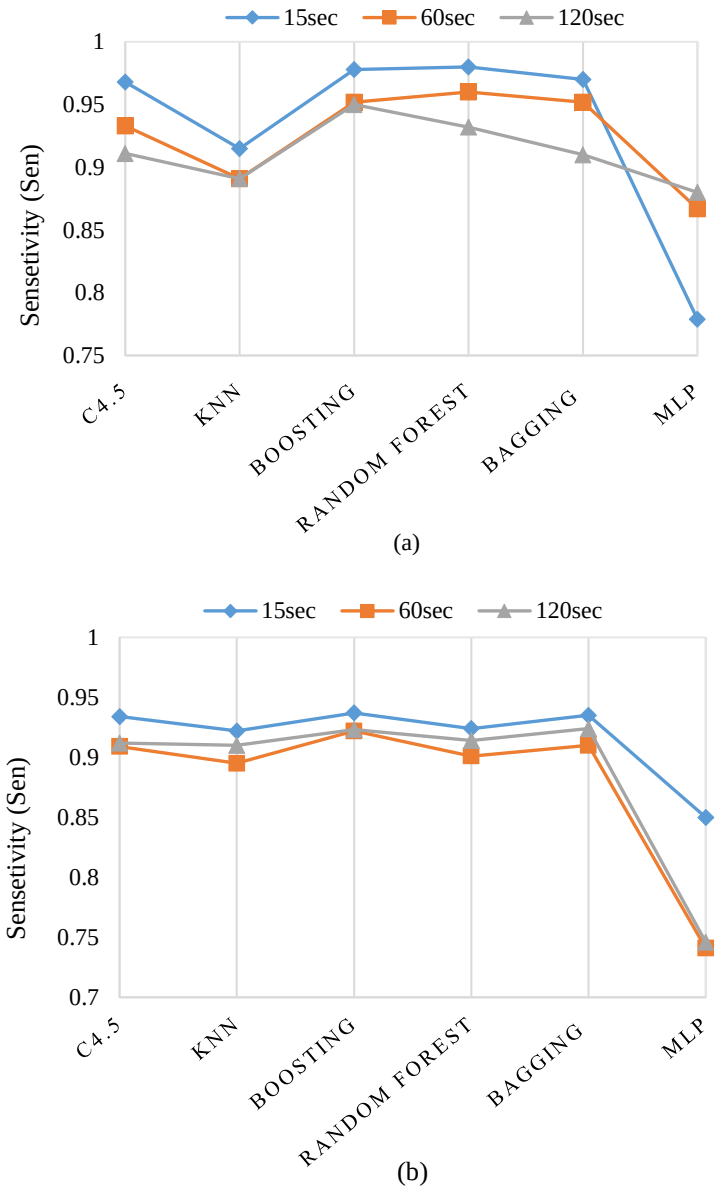

Figure 9. Sensitivity (Sen) of (a) VPN-Non_VoIP and (b) VPN-VoIP traffic flows.

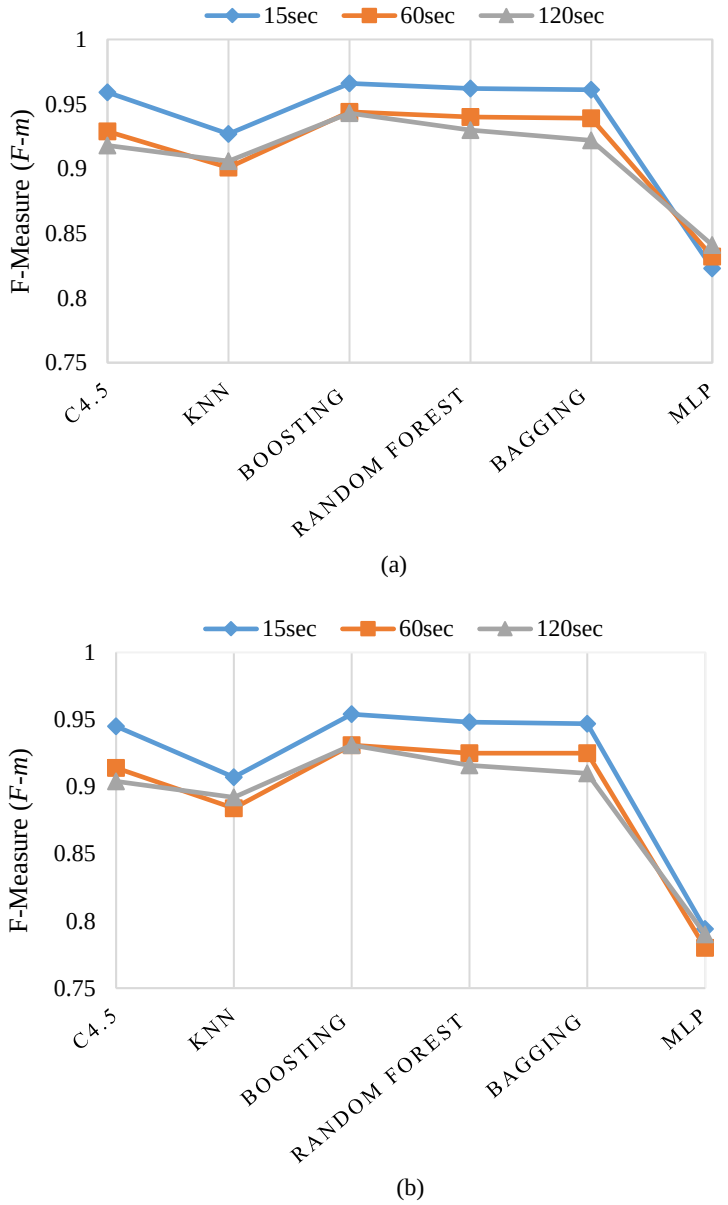

Figure 10. F-measure (F-m) of (a) VPN-Non_VoIP and (b) VPN-VoIP traffic.

Moreover, all the learning algorithms except MLP show prominent classification results for distinguishing the VPN traffic, the Pr is greater than 0.873, the Sen is greater than 0.891, and the $F-m$ is greater than 0.884. The qualitative analysis proved that FSTF set along with shorter FLP are the best choice for the classification and identification of VPN traffic flows. Furthermore, VoIP traffic detection is easily and efficiently achievable.

During the experiments, MLP shows the lowest and inconsistent performance in all three experiments. The corresponding highest Pr, Sen, and $F-m$ for Non-VoIP and VoIP traffic flows are 0.872, 0.779, 0.823 and 0.745, 0.85, and 0.794 with 15 sec FLP respectively. With the longer FLP i.e. 120 sec, the corresponding Pr, Sen, and $F-m$ are 0.81, 0.88, and 0.841 for Non-VoIP, while 0.84, 0.746, and 0.79 in the case of VoIP. In this case, Non-VoIP Pr decreases while Sen and $F-m$ shows

increment with the increase of FLP from 15 to 120 sec. On the contrary, VoIP traffic flows show an increase in Pr, while decreasing results for Sen and $F-m$.

Additionally, the MLP shows clear improvement with the increase in the training time. Figure 11 clearly illustrates that increasing the training time of the model shows enhanced results in both scenarios. Here, we selected the number of epochs, $t_1 = 500$ and $t_2 = 1000$. Almost, all the FLP based datasets show improvement in the Acc. The number of epochs can be selected according to the requirement of the classification model to do a trade-off between computational time and accuracy.

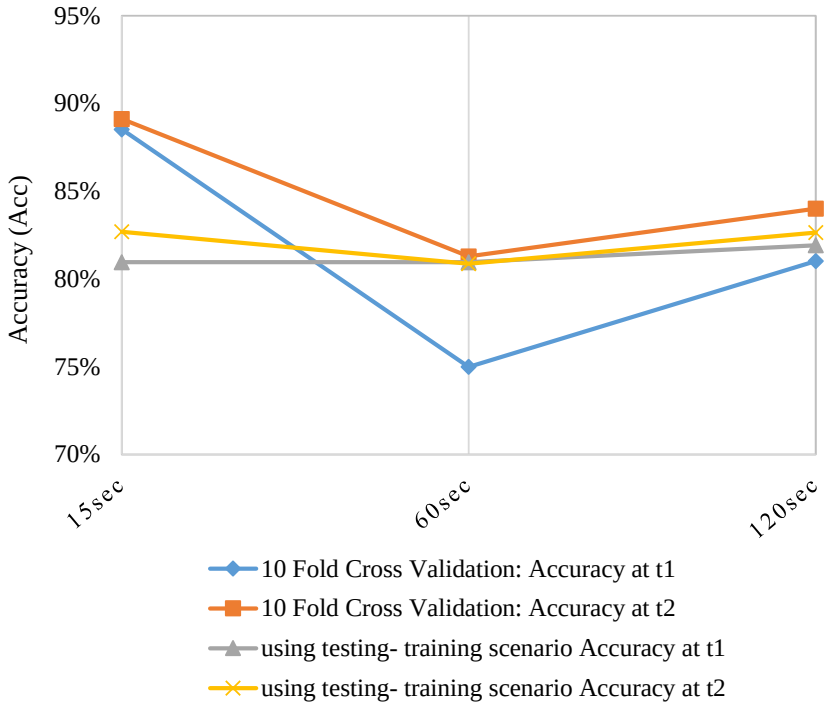

Figure 11. MLP investigation for encrypted VPN traffic classification.

4. Conclusion and future work

The work of this paper is to distinguish the VPN tunneled traffic into VoIP and Non-VoIP traffic. Overall, our approach can identify VoIP traffic generated from different applications among diverse network traffic in the VPN tunnel. Two aspects of experiments represent the efficiency of the proposed identification method. The one is that shorter FLP show consistently better results. The other is the FSTF employed to machine learning algorithms for modeling and identification of network traffic instead of DPI or port numbers. The selected feature set exhibits accurate identification performance for a wide range of encrypted traffic flows. Six learning algorithms were employed during the identification process. During the series of experiments, among the tested six learning algorithms, the boosting technique consistently performed the best on all the given datasets using the FSTF. The boosting based classifier attains the *acc* of (98.27, 97.69 and 98.1%) by 10-Fold Cross-Validation for 15, 60 and 120 sec FLP respectively. Furthermore, the trained model achieved *acc* of (96.1, 93.82 and 93.76%) for 15, 60 and 120 sec FLP, respectively, with testing on unseen testing data. Training time

in all the experiments took several seconds. The proposed method generates three different datasets consists of enough samples in the training data sets from which the mapping between traffic flows and traffic type may be learned. The shorter FLP enhanced the accuracy level and achieved better identification results. As a result, the boosting technique (via C4.5) exhibits significantly better performance as an identification model in both scenarios for all FLP values.

We conclude that our scheme can successfully identify VoIP traffic flows from the mixed network traffic flows generated by various applications based on the FSTF. Future work includes further analysis of application-based VPN-VoIP traffic with further less FLP. The more accurate approaches based on deep learning with less computation complexity are also future targets.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant no. U1836104, 61702235 and 61921004, and partly supported by the Fundamental Research Funds for the Central Universities under Grant no. 30918012204.

Conflict of interests

The authors declare that they have no conflicts of interests.

References

1. M. Shen, M. W. Wei, L. H. Zhu, M. Z. Wang, Classification of encrypted traffic with second-order markov chains and application attribute bigrams, *IEEE Trans. Inf. Forensics Secur.*, **12** (2017), 1830–1843.
2. Y. N. Dong, J. J. Zhao, J. Jin Novel feature selection and classification of internet video traffic based on a hierarchical scheme, *Comput. Networks*, **119** (2017), 102–111.
3. S. E. Middleton, S. Modafferi, Scalable classification of QoS for real-time interactive applications from IP traffic measurements, *Comput. Networks*, **107** (2016), 121–132.
4. P. Burnap, M. L. Williams, Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making, *Policy Internet*, **7** (2015), 223–242.
5. M. Korczyński, A. Duda, *Markov chain fingerprinting to classify encrypted traffic*, In IEEE INFOCOM 2014-IEEE Conference on Computer Communications, 2014, 781–789.
6. P. Velan, M. Čermák, P. Čeleda, M. Drašar, A survey of methods for encrypted traffic classification and analysis, *Int. J. Network Manage.*, **25** (2015), 355–374.
7. Z. Cao, G. Xiong, Y. Zhao, Z. Z. Li, L. Guo, *A survey on encrypted traffic classification*, In International Conference on Applications and Techniques in Information Security, Springer, Berlin, Heidelberg, 2014, 73–81.
8. W. B. Diab, S. Tohme, C. Bassil, *Critical vpn security analysis and new approach for securing voip communications over vpn networks*, In Proceedings of the 3rd ACM workshop on Wireless multimedia networking and performance modeling, 2007, 92–96.

9. J. Khalife, A. Hajjar, J. Diaz-Verdejo, A multilevel taxonomy and requirements for an optimal traffic-classification model, *Int. J. Network Manage.*, **24** (2014), 101–120.
10. N. Namdev, S. Agrawal, S. Silkari, Recent advancement in machine learning based internet traffic classification, *Proc. Comput. Sci.*, **60** (2015), 784–791.
11. M. Finsterbusch, C. Richter, E. Rocha, J. A. Muller, K. Hanssgen, A survey of payload-based traffic classification approaches, *IEEE Commun. Surv. Tutorials*, **16** (2013), 1135–1156.
12. K. S. Shim, J. H. Ham, Baraka D. Sija, M. S. Kim, Application traffic classification using payload size sequence signature, *Int. J. Network Manage.*, **27** (2017), e1981.
13. T. T. Nguyen, G. Armitage, A survey of techniques for internet traffic classification using machine learning, *IEEE commun. Surv. Tutorials*, **10** (2008), 56–76.
14. J. Erman, M. Arlitt, A. Mahanti, *Traffic classification using clustering algorithms*, In Proceedings of the 2006 SIGCOMM workshop on Mining network data, 2006, 281–286.
15. R. Keralapura, A. Nucci, C. Chuah, A novel self-learning architecture for p2p traffic classification in high speed networks, *Comput. Networks*, **54** (2010), 1055–1068.
16. J. Zhang, Y. Xiang, W. L. Zhou, Y. Wang, Unsupervised traffic classification using flow statistical properties and IP packet payload, *J. Comput. Syst. Sci.*, **79** (2013), 573–585.
17. Y. Wang, Y. Xiang, J. Zhang, W. L. Zhou, G. Y. Wei, L. T. Yang, Internet traffic classification using constrained clustering, *IEEE Trans. Parallel Distrib. Syst.*, **25** (2014), 2932–2943.
18. A. Este, F. Gringoli, L. Salgarelli, Support vector machines for TCP traffic classification, *Comput. Networks*, **53** (2009), 2476–2490.
19. A. Finamore, M. Mellia, M. Meo, D. Rossi, Kiss: Stochastic packet inspection classifier for udp traffic, *IEEE/ACM Trans. Networking*, **18** (2010), 1505–1515.
20. L. Zhenxiang, H. Mingbo, L. Song, W. Xin, *Research of P2P traffic comprehensive identification method*, In 2011 International Conference on Network Computing and Information Security, 2011, 307–310.
21. D. J. Arndt, A. Nur Zincir-Heywood, *A comparison of three machine learning techniques for encrypted network traffic analysis*, In 2011 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2011, 107–114.
22. R. Alshammari, A. Nur Zincir-Heywood, Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?, *Comput. networks*, **55** (2011), 1326–1350.
23. T. T. Nguyen, G. Armitage, P. Branch, S. Zander, Timely and continuous machine-learning-based classification for interactive IP traffic, *IEEE/ACM Trans. Networking*, **20** (2012), 1880–1894.
24. W. Ye, K. Cho, Hybrid P2P traffic classification with heuristic rules and machine learning, *Soft Comput.*, **18** (2014), 1815–1827.
25. L. Peng, B. Yang, Y. H. Chen, Effective packet number for early stage internet traffic identification, *Neurocomputing*, **156** (2015), 252–267.
26. J. Zhang, C. Chen, Y. Xiang, W. L. Zhou, *Semi-supervised and compound classification of network traffic*, In 2012 32nd International Conference on Distributed Computing Systems Workshops, 2012, 617–621.

27. J. Yuan, Z. Li, R. Yuan, *Information entropy based clustering method for unsupervised internet traffic classification*, In 2008 IEEE International Conference on Communications, 2008, 1588–1592.
28. M. Zhang, H. L. Zhang, B. Zhang, G.Lu, *Encrypted traffic classification based on an improved clustering algorithm*, In International Conference on Trustworthy Computing and Services, Springer, Berlin, Heidelberg, 2012, 124–131.
29. V. Paxson, Empirically derived analytic models of wide-area TCP connections, *IEEE/ACM Trans. Networking*, **2** (1994), 316–336.
30. V. Paxson, S. Floyd, Wide area traffic: The failure of Poisson modeling, *IEEE/ACM Trans. Networking*, **3** (1995), 226–244.
31. A. McGregor, M. Hall, P. Lorier, J. Brunskill, *Flow clustering using machine learning techniques*, In International workshop on passive and active network measurement, Springer, Berlin, Heidelberg, 2004, 205–214.
32. T. Auld, A. W. Moore, S. F. Gull, Bayesian neural networks for internet traffic classification, *IEEE Trans. Neural Networks*, **18** (2007), 223–239.
33. J. Erman, A. Mahanti, M. Arlitt, I. Cohen, C. Williamson, Offline/realtime traffic classification using semi-supervised learning, *Perform. Eval.*, **64** (2007), 1194–1213.
34. W. Li, M. Canini, A. W. Moore, R. Bolla, Efficient application identification and the temporal and spatial stability of classification schema, *Comput. Networks*, **53** (2009), 790–809.
35. C. Bacquet, K. Gumus, D. Tizer, A. Nur Zincir-Heywood, M. I. Heywood, A comparison of unsupervised learning techniques for encrypted traffic identification, *J. Inf. Assur. Secur.*, **5** (2010), 464–472.
36. D. Arndt, How to: Calculating flow statistics using netmate, 2011. Available from: http://dan.arndt.ca/nims/calculating-flow-statistics-using-netmate/.
37. J. Zhang, C. Chen, Y. Xiang, W. L. Zhou, Y. Xiang, Internet traffic classification by aggregating correlated naive bayes predictions, *IEEE Trans. Inf. Forensics Secur.*, **8** (2013), 5–15.
38. N. F. Huang, G. Y. Jai, H. C. Chao, Y. J. Tzang, H. Y. Chang, Application traffic classification at the early stage by characterizing application rounds, *Inf. Sci.*, **232** (2013), 130–142.
39. Y. J. Fu, H. Xiong, X. Lu, J. Yang, C. Chen, Service usage classification with encrypted internet traffic in mobile messaging apps, *IEEE Trans. Mobile Comput.*, **15** (2016), 2851–2864.
40. M. Conti, L. V. Mancini, R. Spolaor, N. V. Verde, Analyzing android encrypted network traffic to identify user actions, *IEEE Trans. Inf. Forensics Secur.*, **11** (2016), 114–125.
41. Z. Liu, R. Wang, D. Tang, Extending labeled mobile network traffic data by three levels traffic identification fusion, *Future Gener. Comput. Syst.*, **88** (2018), 453–466.
42. G. Aceto, D. Ciuonzo, A. Montieri, A. Pescap, Multi-classification approaches for classifying mobile app traffic, *J. Network Comput. Appl.*, **103** (2018), 131–145.
43. K. L. Dias, M. A. Pongelupe, W. M. Caminhas, L. de Errico, An innovative approach for real-time network traffic classification, *Comput. Networks*, **158** (2019), 143–157.

44. A. J. Pinheiro, J. de M. Bezerra, C. A. Burgardt, D. R. Campelo, Identifying IoT devices and events based on packet length from encrypted traffic, *Comput. Commun.*, **144** (2019), 8–17.
45. Y. M. Choi, *On the accuracy of signature-based traffic identification technique in IP networks*, In 2007 2nd IEEE/IFIP International Workshop on Broadband Convergence Networks, 2007, 1–12.
46. B. C. Park, Y. J. Won, M. S. Kim, J. W. Hong, *Towards automated application signature generation for traffic identification*, In NOMS 2008-2008 IEEE Network Operations and Management Symposium, 2008, 160–167.
47. T. Okabe, T. Kitamura, T. Shizuno, *Statistical traffic identification method based on flow-level behavior for fair VoIP service*, In 1st IEEE Workshop on VoIP Management and Security, 2006, 35–40.
48. D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, P. Tofanelli, *Revealing skype traffic: When randomness plays with you*, SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, 2007, 37–48.
49. R. Alshammari, A. Nur Zincir-Heywood, *An investigation on the identification of VoIP traffic: Case study on Gtalk and Skype*, In 2010 International Conference on Network and Service Management, 2010, 310–313.
50. H. A. H. Ibrahim, S. M. Nor, A. Mohammed, A. B. Mohammed, Taxonomy of machine learning algorithms to classify real time interactive applications, *Int. J. Comput. Networks Wireless Commun.*, **2** (2012), 69–73.
51. D. Adami, C. Callegari, S. Giordano, M. Pagano, T. Pepe, Skype-Hunter: A real-time system for the detection and classification of Skype traffic, *Int. J. Commun. Syst.*, **25** (2012), 386–403.
52. L. A. Khan, M. S. Baig, A. M. Youssef, Speaker recognition from encrypted VoIP communications, *Digital Invest.*, **7** (2010), 65–73.
53. T. Yildirim, P. J. Radcliffe, *VoIP traffic classification in IPSec tunnels*, In 2010 International Conference on Electronics and Information Engineering, 2010, 151–157.
54. B. Li, M. Ma, Z. G. Jin, A VoIP traffic identification scheme based on host and flow behavior analysis, *J. Network Syst. Manage.*, **19** (2011), 111–129.
55. R. Alshammari, A. Nur Zincir-Heywood, Identification of VoIP encrypted traffic using a machine learning approach, *J. King Saud Univ. Comput. Inf. Sci.*, **27** (2015), 77–92.
56. T. Qin, L. Wang, Z. L. Liu, X. H. Guan, Robust application identification methods for P2P and VoIP traffic classification in backbone network, *Knowl. Based Syst.*, **82** (2015), 152–162.
57. M. M. Rathore, A. Ahmad, A. Paul, S. Rho, Exploiting encrypted and tunneled multimedia calls in high-speed big data environment, *Multimedia Tools and Appl.*, **77** (2018), 4959–4984.
58. G. Draper-Gil, A. H. Lashkari, M. S. 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, 407–414.
59. H. L. Arash, G. Draper-Gil, M. S. Mamun, Ali A. Ghorbani, CICFlowMeter: Network traffic flow generator and analyser, Available from: https://www.unb.ca/cic/research/applications.html, 2017.
60. J. R. Quinlan, *C4.5: Program for machine learning*, San Mateo, California, Morgan Kaufmann Publishers, 1993.

61. W. X. Sun, J. Chen, J. Q. Li, Decision tree and PCA-based fault diagnosis of rotating machinery, *Mech. Syst. Signal Process.*, **21** (2007), 1300–1317.

62. L. Breiman, Random forests, *Mach. Learn.*, **45** (2001), 5–32.

63. Y. Freund, R. E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, *J. Comput. Syst. Sci.*, **55** (1997), 23–27.

64. L. Breiman, Bagging predictors, *Mach. Learn.*, **24** (1996), 123–140.

AIMS Press