International Journal of
Intelligent Systems and
Applications in Engineering
ISSN:2147-6799 www.atscience.org/IJISAE
Advanced Technology and Science
Original Research Paper

Proposal of Machine Learning Approach for Identification of Instant Messaging Applications in Raw Network Traffic

Abdurrahman Pektaş*[1]

Accepted : 11/03/2018 Published: 29/06/2018

Abstract: Identification of Internet protocol from either raw network traffic or either network flows plays a crucial role at maintaining and improving the security of computer systems. A significant amount of research is carried out while exploiting a variety of identification techniques. Although certain level in success at detection of network protocols for unencrypted traffic has been achieved, accuracy and performance is rather poor for encrypted traffic. Considering technological trends, new and existing applications have been adopted to use encryption mechanism to protect information and privacy. Therefore, classification of encrypted network traffic is mandatory for ensuring security. Moreover, while performing network forensic investigation, labelling of network protocols/applications is a must to accomplish. In this study, we propose a method to automatically identify instant messaging applications from raw network traffic. To this end, we first extract flow based static features from network capture and then apply machine learning algorithms. The proposed method is evaluated with fairly large dataset. The dataset compromise of publicly available NISM dataset and the network traffic of 9 popular instant messaging applications collected in a controlled environment. The dataset overall contains 716607network flows belonging to 20 application categories. The proposed method classifies network flows of instant messaging applications into their corresponding application categories with the accuracy over 99 percent and F1-score of 99 percent.

Keywords: *Encrypted Traffic Identification, Network flow, Security, Machine Learning, Network Forensics.*

1. Introduction

Monitoring network traffic is an important task and mostly mandatory action to understand the behaviour of data communication processes and to maintain security in computer networks. When data transmission is encrypted, application along its underlying network protocol cannot be classified based on its port number and payload. Advanced classification schemes must be introduced to handle this issue. In particular, statistical features of network traces and features extracted from network flows might be used with machine learning methods to classify encrypted network traffic.

Machine learning has been exploited for Internet traffic classification and it has attracted a lot of attention recently. Research has been particularly focused on identification of the well-known and frequently used applications, and protocols such as DNS, Web, FTP, and database. In the open literature, Moore et al., presents one of the earliest results about classification of network flows into protocol categories by using supervised machine learning [1]. The authors study discriminators and attributes primarily derived from network flows. The author used a feature set consisting of 248 statistical characteristics. A level in classification accuracy is reached at 65% by applying Naive Bayes technique. However, the presented approach depends upon the network addresses (i.e., source and destination IPs) and port numbers used in connections. Thus, individual network traces cannot be classified with this approach.

Identification of application protocols using packet size, timing and direction of a connection is presented in [2]. k-Nearest Neighbour classifier (kNN) and Hidden Markov Model (HMM) are used. Several protocols are identified with 90% accuracy. But however, detection rate is not acceptable level and requires improvement of classification performance.

More recent works have proposed different methods for categorizing network traffic based on flow features without use of either IP address, port or payload information. For example, a binary classification of encrypted SSH or Skype network traffic is proposed in [3], [4]. The features are generated from network traces using open source NetMate tool [5].

A variety of machine learning algorithms including AdaBoost, Naive Bayesian, Support Vector Machine (SVM), C4.5 Decision Tree and RIPPER is applied with the derived feature set. Overall, C4.5 classifier gives better results with a detection rate of 95.9%.

Similarly, the binary identification of SSH versus non-SSH traffic is investigated in [6]. A total of 128 configurations on public network traces are used to generate training and testing set. Different feature sets are evaluated with 3 different supervised learning algorithms. C4.5 achieves more accurate results subject to test data in comparison with K-means and multi-objective genetic algorithm.

The impact of machine learning algorithm selection to the level in accuracy is studied in [7]. 29 statistical flow features are selected and three supervised learning methods SVM, Naive Bayes Kernel Estimation, and C4.5 decision tree are compared. As the most successful, SVM detects encrypted traffic with 97.2% accuracy.

Our study is particularly focused on classification of well-known and popular instant messaging applications without using IP address, port number and payload data. As it is well known fact that instant messaging applications employ encryption methods to ensure security and privacy for users, payload inspection methods cannot work in categorizing network applications.

[1] *Galatasaray University, Department of Computer Engineering, Istanbul, Turkey-TR-3434.*
** Corresponding Author: Email: apektas@yandex.com*

To demonstrate the performance and computational effectiveness of the proposed system, we conduct experiments with fairly large dataset. The dataset is constituted by the public network traces provided by NISM research group [15], and network flow of instant messaging applications generated in our controlled network environment with a particular emphasis on accurate labelling of network traffic. Our contributions can be summarized as follows:

- We apply a feature selection method to improve classification accuracy and also provide greater insight into important features.
- We apply different machine learning techniques to achieve better classification accuracy. According to the evaluation results, decision forest classifies network traffic with more accuracy even subject to an imbalanced dataset. Fairly large dataset is used to test the reliability of the presented feature-based encrypted traffic identification.
- The experiment results show that our method classifies network traces to their corresponding network applications/protocols (i.e. instant messaging apps) with a level in accuracy of 99 percent. The dataset and source-code of this study will be available for academic and research communities, pending approvals [8].

The paper is structured as follows: Section 2 presents the related works on network traffic classification. Section 3 describes the details of the proposed method and the feature set extracted from network traces is elaborated. In Section 4, we briefly describe selected classification methods and elaborate the performance metrics, then plot confusion matrix for better illustration of levels in class-wise classification accuracy among the network protocol/applications. Finally, some conclusions are given.

2. Related Work

Currently, classification of encrypted network traffic while preserving user-privacy is a very challenging task to accomplish. Existing commercial products decrypts the network traffic by using a proxy to analyze it. Consequently, these tools destroy the right to privacy and Internet freedom for users. The privacy-preserving solutions allowing efficient and effective encrypted network classification have been extensively studied in the literature. The proposed methods are mainly focused on machine learning method. In general, these methods uses flow-based features (e.g. flow duration, sent and received bytes in the flow, the size of the first few packets, inter-arrival of the packets, etc.) or payload-based features. The flow-based features employed to identifying the type of network applications in encrypted traffic are more reliable than the payload-based methods. Because, when the encryption method is used (for example SSL/TLS) the payload-data becomes unintelligible byte sequences.

Shbair et al. [9] propose a technique to identify the HTTPS services, i.e. the name of the services, without relying upon particular fields that can be readily changed. The authors uses a set of 42 statistical TLS features for detecting the type of applications that run in SSL/TLS connection. Some of these features are total number of packets, packet size, inter arrival time. Correlation-based filter selection is performed to not only obtain a discriminative feature to increase classification accuracy but also to reduce over-fitting problem. After feature selection process, 18 features are selected. Naïve Bayes, Random Tree, C4.5 Decision Tree and Random Forest machine learning algorithms are evaluated over HTTPS dataset containing more than 288,901 belonging to 9 different services. Based on the evaluation results, The Random Forest achieves 93.5% F-measure.

Qazi et al. [10] introduce a method for identifying mobile application using network flows. The proposed method is tested on 40 Android popular applications in Google Play Store. To enhance the reliability of the evaluation, at least 200 flow records are collected for each application. The methods achieved 94% accuracy. However, as the number of applications to be classified increases, the method remains uncertain whether it sustain good performance.

In [11], the authors use TCP/IP headers for identifying mobile application. The dataset used in the study is collected on 4 different Android devices by running 1,595 applications. The authors employs supervised machine learning algorithms by using packet size of the first 64 packets generated by application. A major drawback with this work is that the accuracy decreases when training and testing devices are different.

Vu et al. [12] address the imbalanced property of network dataset in terms of the amount of encrypted and unencrypted traffic while identifying encrypted network traffic by using machine learning algorithms. They investigate the impact of three different techniques namely; over-sampling, under-sample and generating artificial data. The experimental results showed that the analyzed methods are useful for handling imbalanced network dataset.

In [13], WhatsApp voice calls are characterize via blind traffic detection in order to discriminate WhatsApp calls from other network applications. The authors have evaluated the proposed work on a small dataset by using different Android smartphones. The authors only takes into account the stream based statistical features, and not the payload data. The method precisely detects 86 WhatsApp calls.

It is clear that evaluation of the same machine learning method with different dataset produce different results. Namely, to compare the performance of the different network traffic classification approach, these approaches need to use same dataset. Otherwise, it is unfair to compare different methods. As the existing works predominantly uses private dataset, we cannot compare our work with other studies. Furthermore, to the best of our knowledge, there is currently no existing study focusing on the identification of instant messaging application.

3. Methodology and Feature Set

In this section, we present a classification method for encrypted network traffic. The proposed system considers flow based static features, including total network packets, flow duration, mean, between packets belonging to the same flow to model network.

The proposed methodology, as shown in Figure-1, consists of four major steps. The first step is extracting feature set from raw network capture by using open source NetMate tool [5]. For accurate labelling purposes, only the particular application is run on the phone and its network traffic is captured. Hence, the captured traffic is generated by that particular application's data transmission. To capture network traffic of instant messaging apps, the smart phone is connected to the deployed hotspot network which is hosted on a laptop. By this way, all network traffic originating from smart phone is monitored and captured. Since NetMate tool is capable of processing libpcap file format, the network traces are captured on the virtual hotspot interface by Wireshark sniffing tool on Windows 10 OS. After raw network traffic is saved into the file system, the NetMate tool is used to extract flow based statistical features. The set of extracted features is listed in Table-1.

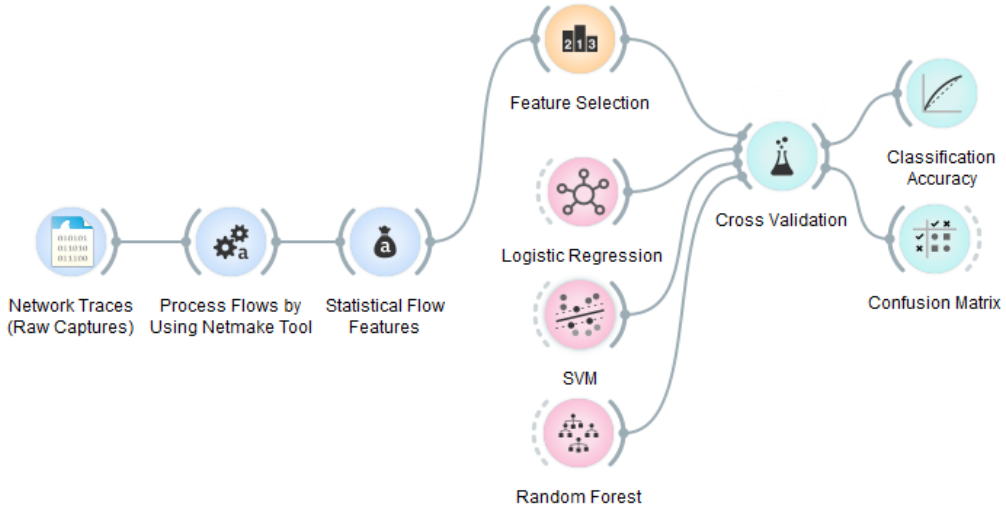


Fig. 1. Overview of the proposed methodology

Table 1. The list of statistical flow-based features

Feature	*Unit*	*Description*
total_fpackets	-	Total packets in the forward direction
total_fvolume	bytes	Total bytes in the forward direction
total_bpackets	-	Total packets in the backward direction
total_bvolume	bytes	Total bytes in the backward direction
min_fpktl	bytes	Minimum packet size in the forward direction
mean_fpktl	bytes	Mean packet size in the forward direction
max_fpktl	bytes	Maximum packet size in the forward direction
std_fpktl	bytes	Standard deviation of the packet length in the forward direction
min_bpktl	bytes	Minimum packet size in the backward direction
mean_bpktl	bytes	Mean packet size in the backward direction
max_bpktl	bytes	Maximum packet size in the backward direction
std_bpktl	bytes	Standard deviation of the packet length in the backward direction
min_fiat	microseconds	Minimum inter-arrival time in the forward direction
mean_fiat	microseconds	Mean inter-arrival time in the forward direction
max_fiat	microseconds	Maximum inter-arrival time in the forward direction
std_fiat	microseconds	Standard deviation of inter-arrival time in the forward direction
min_biat	microseconds	Minimum inter-arrival time in the backward direction
mean_biat	microseconds	Mean inter-arrival time in the backward direction
max_biat	microseconds	Maximum inter-arrival time in the backward direction
std_biat	microseconds	Standard deviation of inter-arrival time in the backward direction
duration	microseconds	Duration of the flow

variance, standard deviations of packet size and inter-arrival time
The second step is dedicated to the selection of appropriate features from the set of flow features. We use a meta-transformer (specifically SelectFromModel transformer in scikit-learn) along with Extra Trees Classifier (see for instance [14]) to select the most representative subset of features toward accurate classification of encrypted network traffic. As a result of this process, each instant messaging application's traffic is represented as a feature matrix and a class label indicating application category.
The third step includes the building classification model based on feature set. Since protocols and applications are vectorised into a sparse matrix, it is an input to the machine learning algorithms to build classification model. In our experiments, we examine three classification methods that are more suitable to high dimensional feature space, including Logistic Regression, Naive Bayes and meta-classifier Random Forest Classification. The final step is the evaluation of classification methods.

2.1. Feature Set

Network captures may provide valuable information about network activities. For example, visited sites, downloaded files, and other activities can be captured by analysing network captures. In this work, our particular focus is on extraction of features from network traces by tracing for both in forward and backward flow direction. Overall, 21 numeric features are obtained and for computational purposes, we convert these features into a feature vector.
For each flow, a feature vector is constituted by the features listed

in Table-1. For the sake of clarity, a definition is given for each feature. The forward direction refers to the direction of the first packet in a network flow, also known as client side or a request. The backward direction is the reverse side of the forward direction, i.e., it is the flow originated by the server side or a reply to a request. A sub-flow is the part of the main flow when protocol or application is inactive, for instance, when data transmission is not occurred for the time period over 1 second.

4. Evaluation and Experimental Study

In this section, we elaborate the performance of the selected algorithms with respect to the feature sets. Our focus in particular is the achievement of a high level in accuracy for classification of network traffic. For this purpose, we test the machine learning algorithms with fairly large dataset.

4.1. Dataset

The NISM dataset is provided by [15] and also used by Alshammari et al. in [4]. This public dataset consists of 713851 network flows belonging to 11 application categories such as DNS, HTTP, SSH, Telnet, SFTP, SCP, etc. We merged instant messaging application's network traffic collected in our controlled network into the NISM dataset. The dataset of instant messaging application consists of 2756 network flows belonging to 9 popular applications such as whatsapp, skype, wechat, etc. In total, the dataset includes 716607 flows belonging to 20 network application/protocol. For accurate labelling of network traffic, while capturing we run only that particular application. Therefore, we assume that captured traffic represents solely the focused application. The categories of the network flows in our dataset are listed in Table 2.

4.2. Evaluation Metrics

To evaluate the proposed classification method, the following metrics are used: ***precision***, ***recall*** (a.k.a. sensitivity), ***F1-score***, ***classification accuracy*** (the overall correctness of the model). In binary classification (positive and negative classes), true positives (*tp*) refer to the correctly predicted positive samples, while true negatives (*tn*) are the number of the correctly predicted negative samples. False positives (*fp*) refer to the incorrectly classified positive samples. Similarly, false negatives (*fn*) are the number of incorrectly classified negative samples. Briefly, the terms positive and negative imply the classifier's success while true and false indicates whether or not the prediction is matched with actual (i.e., ground truth) label.

The precision is the proportion of true positives out of all positive instances, in other words, it is the probability for a positive sample to be classified correctly. The recall is the proportion of instances that are predicted positive and are actually positive (i.e., *tp*). The F1-score, also known as F-measure or F-score, is the weighted harmonic mean of the precision and recall. F1-score is reached at its best value at 1 and lowest value at 0. In binary classification problem, the precision and recall contributes equally to F1-score. However, in multi-class evaluation studies, F1-score is calculated by taking the weighted mean of F1-score of each class. F1-score is also highly referred in the evaluation tasks of natural language processing.

The formulas of the metrics are given as follows:

$$precision = \frac{tp}{tp+fp} \quad (1)$$

$$recall = \frac{tp}{tp+fn} \quad (2)$$

$$F1 - score = 2 \times \frac{precision \times recall}{precision+recall} \quad (3)$$

$$accuracy = \frac{correctly\ classified\ instances}{total\ number\ of\ instances} \quad (4)$$

4.3. Classification Methods

We evaluate three different classifiers; Logistic Regression, Naive Bayes and Random Forest. Our particular aim is to assess whether the statistical flow features can provide sufficient information for the accurate description of network applications. And also, we would like to distinguish the classification method with respect to the level in classification accuracy.

In machine learning, multinomial Logistic Regression algorithm ([16], [17]) is a classification technique introduced to learn and fit the data based on logit function (i.e., logistic curve). More clearly, multinomial logistic regression aims to predict the outcomes of a multi-class problem according to the given set of real-valued, binary-valued or categorical-valued feature. And this algorithm is scalable when a large number of features are used.

For binary classification, SVM classifier divides the n-dimensional feature space into two regions (i.e., positive and negative) on a hyperplane. The hyperplane also called maximum-margin hyperplane is to maximise the distance between the nearest points from both classes that are called support vectors [18]. SVM classifier is practically useful for solving various real world problems. For instance, SVM has been widely used to classify

Table 2. Categories of the network flow in the dataset and their class-specific measures

Class	***Count***	***Precision***	***Recall***	***F1-score***
DNS	38016	1.00	0.95	0.97
FTP	1728	1.00	1.00	1.00
HTTP	11904	0.99	1.00	0.99
TELNET	1251	1.00	1.00	1.00
hangout	620	0.98	1.00	0.99
lime	646271	1.00	1.00	1.00
line	153	0.78	0.93	0.85
localForwarding	2557	1.00	1.00	1.00
messenger	307	1.00	1.00	1.00
remoteForwarding	2422	1.00	0.99	1.00
scp	2444	1.00	1.00	1.00
sftp	2412	1.00	1.00	1.00
shell	2491	0.99	1.00	1.00
skype	221	0.83	0.86	0.84
tango	444	1.00	0.82	0.90
telegram	163	0.93	0.81	0.87
viber	292	0.82	0.93	0.87
wechat	342	0.97	0.94	0.96
whatsapp	214	0.89	0.76	0.82
x11	2355	1.00	1.00	1.00
Weighted avg/total	716607	0.99	0.99	0.99

images and hand-written characters. Moreover, SVM algorithm has been extensively applied in bioinformatics.

Random Forest is an ensemble method designed to increase the accuracy by using collection of decision tree [19]. Each tree is trained on randomly selected features, and each tree votes for the most popular class. Then, the output of the classifier is determined by integrating the votes of trees. Consequently, the Random Forest algorithm performs well subject to high dimensional feature space while being computationally less expensive versus the other ensemble methods. Moreover, making decision over a set of trees leads to a significant increase in classification accuracy.

4.4. Evaluation Results

We use 10-fold cross-validation approach. Since datasets are imbalanced in terms of sample number in each class, we adapt Stratified K-Folds method in the evaluation process. Stratified K-Folds validator splits the data into train and test set by preserving the percentage of samples for each class.

Table-3 gives the general classification accuracy and average recall, precision and F1-score of each machine learning algorithm. Following the numerical results for each metric, meta-classifier Random Forest outperforms the other two classifiers and achieves the highest accuracy about %99.9. Random Forest classifier achieves almost perfect classification accuracy for each application category, see Table-2. In other words, the classification model correctly predicts the categories of the test samples.

Figure-2 shows the class-wise classification accuracy of the network applications. Concerning the metric results in Table-2, the accuracy of the classifier at recognising instances of different classes is illustrated with the confusion matrix in Figure-2. The confusion matrix displays the number of correct and incorrect predictions made by the classifier with respect to ground truth (actual classes). The confusion matrix illustrates that the model is successful at determination of the majority of the network application families. Exceptionally, some network flows excited by skype application are wrongly classified into the scp class and local Forwarding classified as line. This misclassification is mainly due to the imbalanced dataset, the classification model tends to classify network flows into the category which includes most members.

Table 3. . Classification accuracies of the machine learning algorithms

Algorithm	*Accuracy*	*Precision*	*Recall*	*F1-score*
Logistic Regression	0.92	0.87	0.92	0.89
SVM	0.96	0.95	0.96	0.95
Random Forest	0.99	0.99	0.99	0.99

Classification experiments are carried out on a 2.5GHz Intel 4-Core i-7 processor with 8 GB physical memory, using scikit learn ([20], [21]) and MS Windows 10. Processing time costs 80 seconds on average to train about 716K network flows. Overall, 0.14 seconds average processing time is required for classification of a given network flow.

5. Conclusion

In this paper, we present a classification method for network applications, especially instant messaging applications, by extracting their flow based static features. The deployed framework extracts features from a given raw traffic capture. Then, feature selection is performed to distinguish the most important features. Machine learning algorithms are applied to the benchmark dataset. Random Forest algorithm achieves the highest accuracy in classifying network applications into their respective families.

Our experimental results on real-world network captures illustrate the capabilities of the proposed method at identifying the instant messaging applications. The proposed system can be used while performing a forensic analysis on some raw data captured by law enforcement or network security units. We are planning to test the classification system while extending the dataset in terms of sample size and application categories.

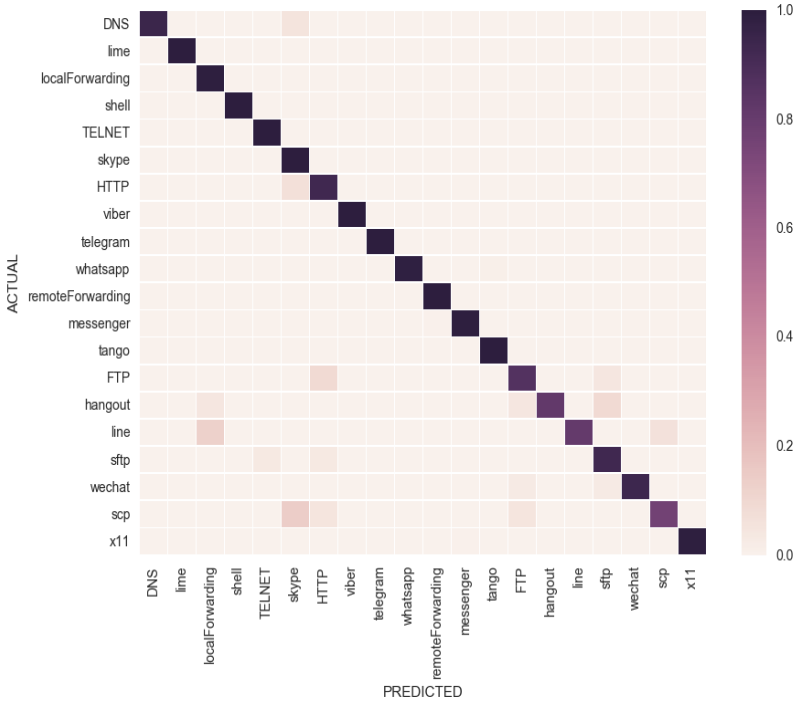


Fig. 2. Normalized confusion matrix for Random Forest classifier

References

[1] A. W. Moore and D. Zuev, “Internet traffic classification using bayesian analysis techniques,” ACM SIGMETRICS Performance Evaluation Review., vol. 33, pp. 50-60, 2005.

[2] C. V Wright, F. Monrose, and G. M. Masson, “On inferring application protocol behaviors in encrypted network traffic,” Journal of Machine Learning Research, vol. 7, pp. 2745-2769, 2006.

[3] R. Alshammari and A. N. Zincir-Heywood, “Machine learning based encrypted traffic classification: Identifying ssh and skype”, CISDA, vol. 9, pp. 289-296, 2009.

[4] R. Alshammari and A. N. Zincir-Heywood, “Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?” Computer networks, vol. 55, no.6, pp. 1326-1350, 2011.

[5] Calculating Flow Statistics Using NetMate, 2017. [Online], Available: https://dan.arndt.ca/nims/calculating-flow-statistics-using-netmate/ . Accessed on: Jan15, 2017.

[6] D. J. Arndt and A N. Zincir-Heywood, “A comparison of three machine learning techniques for encrypted network traffic analysis,” In Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2011, pp. 107-114.

[7] Y. Okada, S. Ata, N. Nakamura, Y. Nakahira, and I. Oka, “Comparisons of machine learning algorithms for application identification of encrypted traffic,” In Proc. Machine Learning and Applications and Workshops (ICMLA), 2011, pp. 358-361.

[8] Github repo containing the source code and the dataset of this work, 2017, [Online], Available: https://gitlab.com/apektas/instant_messaging_app_identification. Accessed on: Feb-12, 2017.

[9] W. M. Shbair, T. Cholez, J. Francois, I. Chrisment, "A multi-level framework to identify HTTPS services,". In Proc. Network Operations and Management Symposium (NOMS), 2016, pp. 240-248.

[10] Z. A. Qazi, J. Lee, T. Jin, G. Bellala, M. Arndt, G. Noubir, "Application-awareness in SDN," ACM SIGCOMM computer communication review, vol. 43, no. 4, pp. 487-488, 2013.

[11] H. F. Alan, J. Kaur, "Can Android Applications Be Identified Using Only TCP/IP Headers of Their Launch Time Traffic?," in Proc. 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks, 2016, pp. 61-66.

[12] L. Vu, D. Tra Van, Q. U, Nguyen, "Learning from imbalanced data for encrypted traffic identification problem," in Proc. Seventh Symposium on Information and Communication Technology, 2016, pp. 147-152.

[13] A. Cuadra-Sanchez, J. Aracil, "A novel blind traffic analysis technique for detection of WhatsApp VoIP calls," International Journal of Network Management, vol. 27, no. 2, 2017.

[14] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” Machine learning, vol. 63, no. 1, pp. 3-42, 2006.

[15] NIMS1 data set, 2017, [Online], Available: https://projects.cs.dal.ca/projectx/data/NIMS.arff.zip. Accessed on: Jan-15, -2017.

[16] H. Yu, F. Huang, and C. Lin, “Dual coordinate descent methods for logistic regression and maximum entropy models,” Machine Learning, vol. 85, no.1, pp.41-75, 2011.

[17] M. Schmidt, N. L. Roux, and F. Bach, “Minimizing finite sums with the stochastic average gradient,” Mathematical Programming, pp. 1-30, 2013.

[18] T. Wu, C. Lin, and R. C. Weng, “Probability estimates for multiclass classification by pairwise coupling,” Journal of Machine Learning Research, vol. 5, pp.975-1005, 2004.

[19] L. Breiman,. “Random forests,” Machine learning, vol. 45, no. 1, pp. 5-32, 2001.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, “Scikit-learn: Machine learning in python,”. Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[21] Scikit-learn: machine learning in Python, 2017, [Online], Available: http://scikit-learn.org/stable/index.html., Accessed on: Mar-15, 2017.