Characterization of Encrypted and VPN Traffic Using Time-Related Features

Conference Paper · February 2016

DOI: 10.5220/0005740704070414

4 authors:

Arash Habibi Lashkari
University of New Brunswick
95 PUBLICATIONS **1,514** CITATIONS
SEE PROFILE

Gerard Draper Gil
European Commission
16 PUBLICATIONS **232** CITATIONS
SEE PROFILE

Mohammad Saiful Islam Mamun
National Research Council Canada
25 PUBLICATIONS **351** CITATIONS
SEE PROFILE

Ali A. Ghorbani
University of New Brunswick
263 PUBLICATIONS **7,216** CITATIONS
SEE PROFILE

Characterization of Encrypted and VPN Traffic using Time-related Features

Gerard Draper-Gil, Arash Habibi Lashkari, Mohammad Saiful Islam Mamun and Ali A. Ghorbani
University of New Brunswick, Fredericton NB E3B 5A3, New Brunswick, Canada
{*gerard.draper, msi.mamun, a.habibi.l, ghorbani*}*@unb.ca*

Keywords: Traffic Classification, Encrypted Traffic Characterization, Flow Time-based Features, VPN Traffic Characterization, Flow Timeout Value.

Abstract: Traffic characterization is one of the major challenges in today's security industry. The continuous evolution and generation of new applications and services, together with the expansion of encrypted communications makes it a difficult task. Virtual Private Networks (VPNs) are an example of encrypted communication service that is becoming popular, as method for bypassing censorship as well as accessing services that are geographically locked. In this paper, we study the effectiveness of flow-based time-related features to detect VPN traffic and to characterize encrypted traffic into different categories, according to the type of traffic e.g., browsing, streaming, etc. We use two different well-known machine learning techniques (C4.5 and KNN) to test the accuracy of our features. Our results show high accuracy and performance, confirming that time-related features are good classifiers for encrypted traffic characterization.

1 INTRODUCTION

Traffic classification technologies have received increased attention over the last decade due to the implementation of mechanisms for network quality of service (QoS), security, accounting, design and engineering. The networking industry as well as the research community have dedicated many efforts to the research of these technologies and came up with several classification techniques (Callado et al., 2009). However, the continuous expansion of Internet and mobile technologies are creating a dynamic environment where new applications and services emerge every day, and the existing ones are constantly evolving. Moreover, encryption is becoming pervasive in today's Internet, serving as a base for secure communications. This constant creation, evolution, and securization of applications makes traffic classification a great challenge for the Internet research community.

Traffic classification can be categorized based on its final purpose: associating traffic with encryption (e.g., encrypted traffic), protocol encapsulation (e.g., tunneled through VPN or HTTPS); according to specific applications, (e.g., Skype), or according to the application type (e.g., Streaming, Chat), also called traffic characterization. Some applications (e.g., Skype, Facebook) support multiple services like chat, voice call, file transfer, etc. These applications require identifying both the application itself and the specific task associated with it. Very few traffic classification techniques in the literature address this challenging trends (Wang et al., 2014; Rao et al., 2011; Coull and Dyer, 2014).

In early 90's, the initial traffic classification techniques associated transport layer ports with specific applications, a simple and fast technique. But, its low accuracy and unreliability rendered the development of Deep Packet Inspection (DPI) approaches. The DPI approach analyzes packets and classifies them according to some stored signature or pattern. However, DPI techniques that require payload examination are not computationally efficient, specially over high-bandwidth network. Moreover, they are often circumvented by encapsulated, encrypted, or obfuscated traffic that precludes payload analysis.

Selecting effective and reliable features for traffic analysis is still a serious challenge. Generally speaking the classification of network traffic falls mainly into two categories: flow-based classification, using properties such as flow bytes per second, duration per flow, etc. and packet-based classification, using properties such as size, inter-packet duration of the first (or n) packets, etc.

In this paper, we focus on analyzing regular encrypted traffic and encrypted traffic tunneled through a Virtual Private Network (VPN). The characteriza-

Draper-Gil, G., Lashkari, A., Mamun, M. and Ghorbani, A.
Characterization of Encrypted and VPN Traffic using Time-related Features.
In *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016)*, pages 407-414
ISBN: 978-989-758-167-0

tion of VPN traffic is a challenging task that remains to be solved. VPN tunnels are used to maintain the privacy of data shared over the physical network connection holding packet-level encryption, therefore making very difficult to identify the applications running through these VPN services.

Our Contribution in this paper is twofold. First, we propose a flow-based classification method to characterize encrypted and VPN traffic using only *time*-related features. Moreover, we reduce the computational overhead by reducing the set of features to a set that can be extracted with low computational complexity (Kim et al., 2008; Li et al., 2009). And second, we generate and publish an extensive labeled dataset of encrypted traffic, with 14 different labels (7 for regular encrypted traffic and 7 for VPN traffic). We choose only *time*-related features to expedite the efficiency and to ensure an encryption independent traffic classifier.

The remainder of this paper is organized as follows: Section 2 presents an overview of encrypted traffic classification. In Section 3 we describe the dataset. In Section 4 describes the experiments executed on the captured dataset, while Section 5 presents and discusses the results obtained. Finally, Section 6 presents the conclusions and future work.

2 RELATED WORK

Studies on packet size and flow based traffic classification were started in early 90's by Paxson *et al.* in (Paxson, 1994; Paxson and Floyd, 1995), where some statistical features like packet length, inter-arrival times and flow duration were supposed to be suitable to trace protocols. Later Belzarena *et al.* in (Gómez Sena and Belzarena, 2009) and Li *et al.* in (Li et al., 2009) used the statistics from the first few packets of the flow to gain efficiency. Moreover, in order to expedite the classification efficiency in a high-scale, high speed network, Nucci *et al.* in (Yeganeh et al., 2012) and Pescap *et al.* in (Aceto et al., 2010) proposed a signature based traffic identification scheme. Although they reduced the time to classify the flows, they failed to detect unknown or manually created signatures.

Traffic characterization techniques are not widely addressed in the current literature. Moreover, most of them focus on specific application type or devices. Wang *et al.* (Wang et al., 2014) proposed a model to characterize P2P traffic. They extracted features from multiple flows and aggregated flows into clusters to extract P2P application behaviour. Coull *et al* (Coull and Dyer, 2014) present a study on the iMessage protocol to identify the type of device. In (Rao et al., 2011), Rao *et al.* propose a network characteristics model for two of the most most popular video streaming services, Netflix and YouTube. In (Mauro and Longo, 2015), Mauro and Longo propose a method to detect encrypted WebRTC traffic. Mamun *et al.* (Mohammad S.I. Mamun and Ghorbani, 2015) proposed a method to identify enrypted traffic by measuring the entropy of the packet's payload. Sherry *et al.* (Sherry et al., 2015) propose a DPI system that can inspect encrypted payload without decrypting it, therefore maintaining the privacy of the communications, but it can only process HTTPS traffic.

A number of machine learning classification methods based on flow (Bernaille and Teixeira, 2007; Moore and Zuev, 2005) and packet-based (Iliofotou et al., 2007; Karagiannis et al., 2005) features have been proposed in the literature to identify traffic accurately. However, traffic classification for the encapsulated protocols (e.g., using Proxy server or VPN tunnels) that are mainly used for hiding the identities of the users for privacy reasons, are challenging and hence are not widely explored in the literature. However, recently, Heywood et al. in (Aghaei-Foroushani and Zincir-Heywood, 2015) proposed a data driven classifier to identify traffic coming from clients behind a proxy server using traffic flow information.

To the best of our knowledge, we are the first to propose a method to characterize VPN traffic in a broad sense, identifying 7 different traffic categories.

3 DATASET GENERATION

To create a representative dataset we captured real traffic generated by our lab members. We created accounts for users Alice and Bob in order to use services like Skype, Facebook, etc. In Table 1 we provide the complete list of different types of traffic and applications included in our dataset. For each traffic type (VoIP, P2P, etc...) we captured a regular session and a session over VPN, therefore we have a total of 14 traffic categories: VOIP, VPN-VOIP, P2P, VPN-P2P, etc. Following, we give a detailed description of the different types of traffic generated:

Browsing: Under this label we have HTTPS traffic generated by users while browsing or performing any task that includes the use of a browser. For instance, when we captured voice-calls using hangouts, even though browsing is not the main activity, we captured several browsing flows.

Table 1: List of Captured protocols and applications.

Traffic	Content
Web Browsing	Firefox and Chrome
Email	SMPTS, POP3S and IMAPS
Chat	ICQ, AIM, Skype, Facebook and Hangouts
Streaming	Vimeo and Youtube
File Transfer	Skype, FTPS and SFTP using Filezilla and an external service
VoIP	Facebook, Skype and Hangouts voice calls (1h duration)
P2P	uTorrent and Transmission (Bittorrent)

Table 2: List of time based features.

Feature	Description
duration	The duration of the flow.
fiat	Forward Inter Arrival Time, the time between two packets sent forward direction (mean, min, max, std).
biat	Backward Inter Arrival Time, the time between two packets sent backwards (mean, min, max, std).
flowiat	Flow Inter Arrival Time, the time between two packets sent in either direction (mean, min, max, std).
active	The amount of time time a flow was active before going idle (mean, min, max, std).
idle	The amount of time time a flow was idle before becoming active (mean, min, max, std).
fb_psec	Flow Bytes per second.
fp_psec	Flow packets per second.

Email: The traffic samples generated using a Thunderbird client, and Alice and Bob Gmail accounts. The clients were configured to deliver mail through SMTP/S, and receive it using POP3/SSL in one client and IMAP/SSL in the other.

Chat: The chat label identifies instant-messaging applications. Under this label we have Facebook and Hangouts via web browser, Skype, and IAM and ICQ using an application called pidgin.

Streaming: The streaming label identifies multimedia applications that require a continuous and steady stream of data. We captured traffic from Youtube (HTML5 and flash versions) and Vimeo services using Chrome and Firefox.

File Transfer: This label identifies traffic applications whose main purpose is to send or receive files and documents. For our dataset we captured Skype file transfers, FTP over SSH (SFTP) and FTP over SSL (FTPS) traffic sessions.

VoIP: The Voice over IP label groups all traffic generated by voice applications. Within this label we captured voice-calls using Facebook, Hangouts and Skype.

P2P: This label is used to identify file-sharing protocols like Bittorrent. To generate this traffic we downloaded different .torrent files from a public a repository (*archive.org*) and captured traffic sessions using the uTorrent and Transmission applications.

The traffic was captured using Wireshark and tcpdump , generating a total amount of 28GB of data. For the VPN traffic, we used an external VPN service provider and connected to it using OpenVPN. To generate SFTP and FTPS traffic we also used an external service provider and Filezilla as a client.

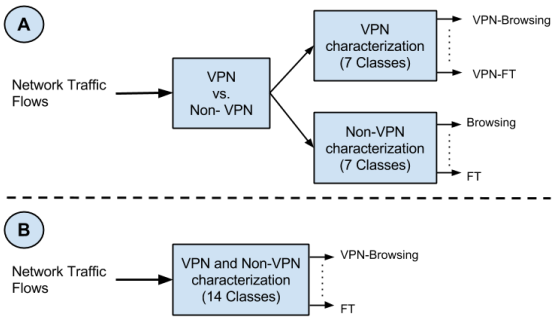


Figure 1: Characterization Scenarios.

4 EXPERIMENTS

We have defined two different scenarios A and B, depicted in Figure 1. As described in Section 3, we have used 4 different flow timeout values to generate our datasets, and we have chosen 2 machine learning algorithms (C4.5 and KNN). Therefore, we will have to execute each experiment 8 times. We have

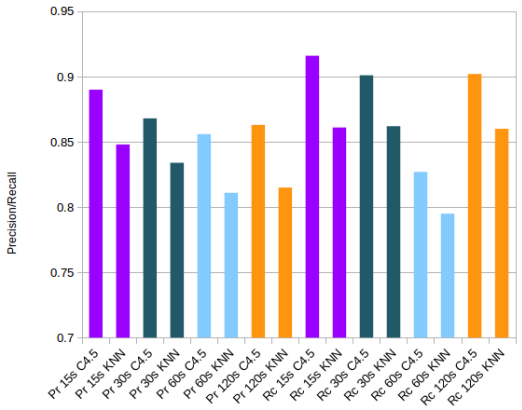
designed a total of 3 experiments, 2 for scenario A and one for scenario B:

Scenario A: The objective of this scenario is to characterize encrypted traffic with VPN identification, e.g. we will distinguish between voice-calls (VOIP) and voice-calls tunneled through VPN (VPN-VOIP). As a result we will have 14 different types of traffic, 7 regular types of encrypted traffic and 7 VPN types of traffic. In this Scenario we do the characterization in two steps. First, we distinguish between VPN and Non-VPN traffic and then we characterize each type of traffic separately (VPN and Non-VPN). In order to do this, we have divided our dataset in two different datasets: one with regular encrypted traffic flows and the other one with VPN traffic flows.

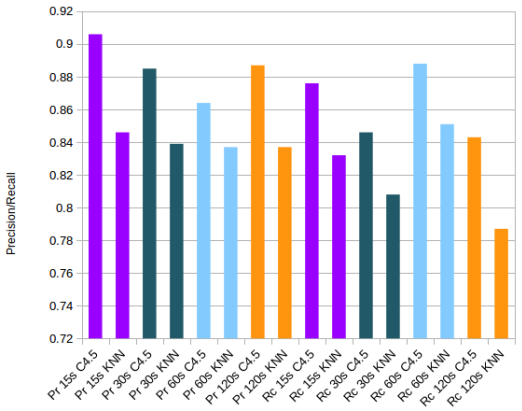
Scenario B: In this Scenario, we use a mixed dataset to do the characterization in one step. The input of our classifier is regular encrypted traffic and VPN traffic, and as output we have the same 14 different categories (Section 3).

4.1 Flow and Features Generation

We use a common definition of flow, where a flow is defined by a sequence of packets with the same values for {*Source IP*, *Destination IP*, *Source Port*, *Destination Port* and *Protocol (TCP or UDP)*}. Flows are considered to be bidirectional (forward and reverse directions) as in most of the reviewed papers (e.g.,(McGregor et al., 2004; Zander et al., 2005; Bernaille et al., 2006; Williams et al., 2006; Palmieri and Fiore, 2009)). Along with the flow generation we have to calculate the features associated with each flow. Many papers in the literature use a tool called NetMate to generate flows and features, but as part of our work we have developed our an application, ISCXFlowMeter. It is written in Java and gives us more flexibility in terms of choosing the features we want to calculate, adding new ones, and also having a better control of the duration of the flow timeout. ISCXFlowMeter generates bidirectional flows, where the first packet determines the forward (source to destination) and backward (destination to source) directions, hence the statistical *time*-related features are also calculated separately in the forward and reverse direction. Note that TCP flows are usually terminated upon connection teardown (by FIN packet) while UDP flows are terminated by a flow timeout. The flow timeout value can be assigned arbitrarily by the individual scheme e.g., 600 seconds for both TCP and UDP in (Aghaei-Foroushani and Zincir-Heywood, 2015). In this paper, we study several flow timeout (ftm) values with their corresponding classifier accuracy on the same dataset. In particular, we set the duration of flows to 15,30,60 and 120 seconds.



(a) Scenario A VPN Precission and Recall

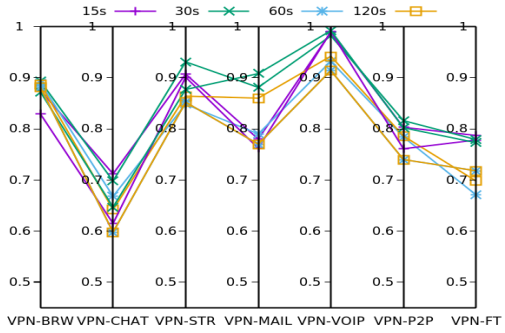


(b) Scenario A NON-VPN Precission and Recall

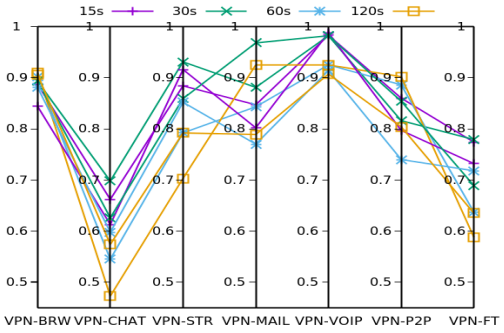
Figure 2: Scenario A-1: VPN detection.

In our experiments, the classifier has a response time of $(FT + FE + ML)$ seconds, where FT is the customized flow-time, FE is the feature extraction time and ML is the machine learning algorithm time to perform classification. It has been observed that the maximum accuracy is achieved with $(FT = 15s)$ for all the classifiers. In the current implementation, we have found that the average delay attained is approx. $(FT + FE + ML = 15 + .001 + .01(kNN)$ *or* $1.26(C4.5) = 15.011$ *sec* (kNN) *or* 16.261 *sec* $(C4.5)$) for the VPN classifier and $(FT + FE + ML = 15 + .001 + .01(kNN)$ *or* $1.49(C4.5) = 15.011$ *sec* (kNN) *or* 16.491 *sec* $(C4.5)$) for the traffic type classifier.

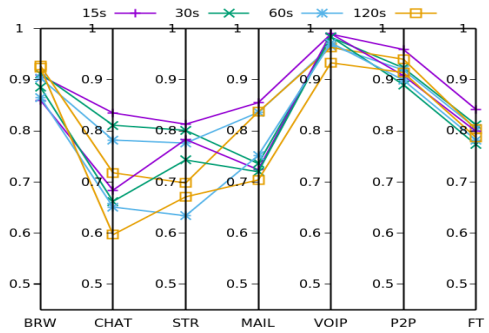
As previously mentioned, we focus on *time*-related features. When choosing *time*-related fea-



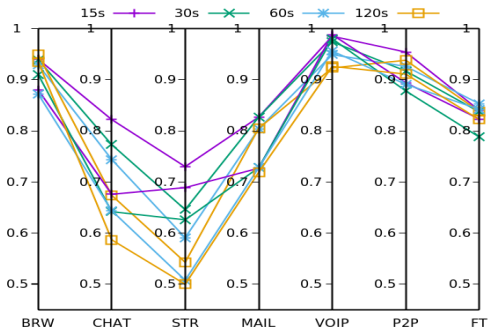
(a) ScenarioA VPN Precision



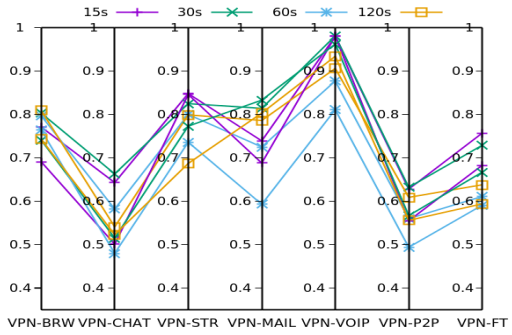
(b) ScenarioA VPN Recall



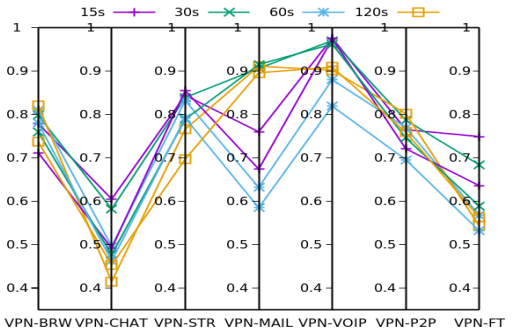
(c) ScenarioA Non-VPN Precision



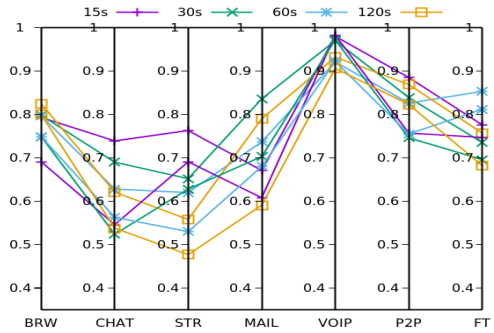
(d) ScenarioA Non-VPN Recall



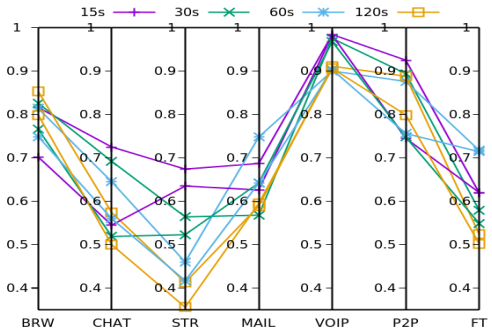
(e) ScenarioB VPN Precision



(f) ScenarioB VPN Recall



(g) ScenarioB Non-VPN Precision



(h) ScenarioB Non-VPN Recall

Figure 3: Precision and Recall of traffic characterization.

tures, we consider two different approaches. In the first approach we measure the time, e.g. time between packets or the time that a flow remains active. In the second approach, we fix the time and measure other variables, e.g., bytes per second or packets per second. In Table 2 we provide the complete list of features extracted in this work. As one can see form Table 2, except the duration, which shows the total time of one flow, there are six groups of features. The first three groups are namely: -fiat, -biat, and -flowiat, and are focused respectively on the forward, backward and bi-directional flows. The fourth and fifth groups of features, are calculated regarding to the idle-to-active or active-to-idle states and are named -idle and -active. Finally, the last group focuses on the size and number of packets per second and is named -psec feature.

4.2 Machine Learning Approaches

To execute the experiments we used Weka (Hall et al., 2009), a well known tool that implements different machine learning algorithms. We used its default settings with 10 fold cross validation. Although Weka includes many different algorithms for clustering and classifying, regarding to the previous research work and the human readability, we have selected two algorithms from the supervised and unsupervised families: C4.5 decision tree and KNN.

C4.5 Decision Tree: Developed by Ross Quinlan, this algorithm is one of the most popular classification techniques in machine learning and data mining. It is based on the concept of Information entropy. The algorithm requires a set of training pairs {inputs-output} where the output is the corresponding class. Both numerical and categorical data are supported, and the result is presented as a tree, making it readable for humans.

KNN: The K-Nearest Neighbors algorithm is one of the most simple algorithms in machine learning. It is based on similarity measures, thus it depends on the metric used to calculate the distance between examples. The output of the classification is a class membership, which is determined according to the majority vote of its K nearest neighbours.

To evaluate the quality of our classification processes, we will use two common metrics: Precision (Pr) or Positive Predictive value and Recall (Rc) or Sensitivity.

$$Pr = \frac{TP}{TP+FP} \qquad Rc = \frac{TP}{TP+FN}$$

Where the TP is the number of instances correctly classified as A, FP is the number of instances incorrectly classified as A, and FN is the number of instances incorrectly classified as Not-A.

5 ANALYSIS OF THE RESULTS

In the Figures 2 and 3 we can see the Precision and Recall of the different results. Overall C4.5 and KNN had similar results, although C4.5 performed a little better. But interestingly the results present a dependance on the flow-timeout value selected. Therefore we have chosen to focus the attention on these result. For each flow tiemout value we have two different representations (two lines) one of them corresponds to the C4.5 result and the other one to the KNN.

5.1 Analysis of Scenario A

In the Figure 2 we have the Precision (Pr) and Recall (Rc) results of the first part of the scenario A, where we classify traffic into VPN and Non-VPN. We can see that there is a direct relation between flow timeout (ftm) values and the performance of the classifiers. In particular, the Precision (Pr) of the C4.5 VPN traffic classifier decreases from 0.890 using 15 seconds to 0.86 using 120 seconds, and the Pr for Non-VPN traffic decreases from 0.906 to 0.887. We can see a similar behavior in the case of the KNN algorithm, where the Pr for VPN traffic decreases from 0.848 to 0.815, and from 0.846 to 0.837 in the case of Non-VPN traffic. The best results are achieved using the C4.5 algorithm and 15s ftm: 0.89 for VPN and 0.906 for Non-VPN. This means that, using *time*-related features we can distinguish VPN from Non-VPN with a 15s delay (the time it takes to build a flow). These results show that when using *time*-related features for VPN and Non-VPN traffic classification, using shorter timeout values improve the accuracy rate.

The second part of scenario A focuses on the characterization of VPN and Non-VPN traffic (see Figure 3 parts a,b,c,d), separately. The input is classified according to the traffic categories defined in Section 3. Again, the results for shorter ftm values are better than the results for larger values, although with a few exceptions in the case of the VPN classifier (Figures 3a, 3b), like VPN-MAIL where the best result is obtained with an ftm of 30s. In the case of the Non-VPN classifier (Figures 3c, 3d) this trend can be clearly seen.

The best results (average Pr) are obtained with C4.5 and 15s of ftm: 0.84 and 0.89 for the VPN and Non-VPN classifiers respectively. Moreover, the average Pr for all traffic categories is higher than 0.84,

which means that *time*-related features are good classifiers to characterize encrypted and VPN traffic.

5.2 Analysis of Scenario B

In this Scenario all encrypted and VPN traffic are mixed together in one dataset, and the objective is to characterize the traffic without previously dividing VPN from Non-VPN traffic, therefore we will have 14 types of traffic: 7 encrypted and 7 VPN traffic categories. The results are shown in Figure 3 (parts e,f,g,h).

In this case, we cannot see the pattern *'shorter timeout - better accuracy'* as clear as in the previous scenario (5.1). For example using the C4.5 algorithm the Pr of VPN-Browsing, VPN-Mail, and Mail with 15 sec is 0.771, 0.739, 0.671 respectively, values lower than the 0.809, 0.786, 0.79 obtained with 120 sec. The KNN results are similar, the Pr of VPN-Browsing, VPN-Chat, and VPN-Mail traffic categories is (0.691, 0.501, 0.688) for 15s. ftm, smaller than the Pr obtained with 120 sec (0.743, 0.501, 0.688). On the other hand, the highest average Pr from the different ftm values is around 0.783 for C4.5 and 0.711 for KNN algorithms, around 0.5 points lower that the best values from Scenario A.

6 CONCLUSIONS

In this paper we have studied the efficiency of time-related features to address the challenging problem of characterization of encrypted traffic and detection of VPN traffic. We have proposed a set of time-related features and two common machine learning algorithms, C4.5 and KNN, as classification techniques. Our results prove that our proposed set of time-related features are good classifiers, achieving accuracy levels above 80%. C4.5 and KNN had a similar performance in all experiments, although C4.5 has achieved better results. From the two scenarios proposed, characterization in 2 steps (scenario A) vs. characterization in one step (scenario B), the first one generated better results. In addition to our main objective, we have also found that our classifiers perform better when the flows are generated using shorter timeout values, which contradicts the common assumption of using 600s as timeout duration. As future work we plan to expand our work to other applications and types of encrypted traffic, and to further study the application of time-based features to characterize encrypted traffic.

REFERENCES

Aceto, G., Dainotti, A., de Donato, W., and Pescape, A. (2010). Portload: Taking the best of two worlds in traffic classification. In *IEEE Conference on Computer Communications Workshops*, INFOCOM 2010, pages 1–5. IEEE.

Aghaei-Foroushani, V. and Zincir-Heywood, A. (2015). A proxy identifier based on patterns in traffic flows. In *IEEE 16th International Symposium on High Assurance Systems Engineering*, HASE 2015, pages 118–125. IEEE.

Bernaille, L. and Teixeira, R. (2007). Early recognition of encrypted applications. In *Proceedings of the 8th International Conference on Passive and Active Network Measurement*, PAM'07, pages 165–175, Berlin, Heidelberg. Springer-Verlag.

Bernaille, L., Teixeira, R., Akodkenou, I., Soule, A., and Salamatian, K. (2006). Traffic classification on the fly. *ACM SIGCOMM Computer Communication Review*, 36(2):23–26.

Callado, A., Kamienski, C., Szabo, G., Gero, B., Kelner, J., Fernandes, S., and Sadok, D. (2009). A survey on internet traffic identification. *Communications Surveys & Tutorials, IEEE*, 11(3):37–52.

Coull, S. E. and Dyer, K. P. (2014). Traffic analysis of encrypted messaging services: Apple imessage and beyond. *ACM SIGCOMM Computer Communication Review*, 44(5):5–11.

Gómez Sena, G. and Belzarena, P. (2009). Early traffic classification using support vector machines. In *Proceedings of the 5th International Latin American Networking Conference*, LANC '09, pages 60–66, New York, NY, USA. ACM.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Iliofotou, M., Pappu, P., Faloutsos, M., Mitzenmacher, M., Singh, S., and Varghese, G. (2007). Network monitoring using traffic dispersion graphs (tdgs). In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 315–320, New York, NY, USA. ACM.

Karagiannis, T., Papagiannaki, K., and Faloutsos, M. (2005). Blinc: Multilevel traffic classification in the dark. In *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '05, pages 229–240, New York, NY, USA. ACM.

Kim, H., Claffy, K., Fomenkov, M., Barman, D., Faloutsos, M., and Lee, K. (2008). Internet traffic classification demystified: Myths, caveats, and the best practices. In *Proceedings of the 2008 ACM CoNEXT Conference*, CoNEXT '08, pages 11:1–11:12, New York, NY, USA. ACM.

Li, W., Canini, M., Moore, A. W., and Bolla, R. (2009). Efficient application identification and the temporal and spatial stability of classification schema. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 53(6):790–809.

Mauro, M. D. and Longo, M. (2015). Revealing encrypted webrtc traffic via machine learning tools. In *Proceedings of the 12th International Conference on Security and Cryptography*, SECRYPT '15, pages 259–266. SciTePress.

McGregor, A., Hall, M., Lorier, P., and Brunskill, J. (2004). Flow clustering using machine learning techniques. In *Passive and Active Network Measurement*, volume 3015 of *Lecture Notes in Computer Science*, pages 205–214. Springer Berlin Heidelberg.

Mohammad S.I. Mamun, N. S. and Ghorbani, A. A. (2015). An entropy-based encrypted traffic classification using machine learning. In *Proceedings of the 17th International Conference on Information and Communication Security*, ICICS 2015, Berlin, Heidelberg. Springer-Verlag.

Moore, A. W. and Zuev, D. (2005). Internet traffic classification using bayesian analysis techniques. In *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '05, pages 50–60, New York, NY, USA. ACM.

Palmieri, F. and Fiore, U. (2009). A nonlinear, recurrence-based approach to traffic classification. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 53(6):761–773.

Paxson, V. (1994). Empirically derived analytic models of wide-area tcp connections. *IEEE/ACM Transactions on Networking*, 2(4):316–336.

Paxson, V. and Floyd, S. (1995). Wide area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244.

Rao, A., Legout, A., Lim, Y.-s., Towsley, D., Barakat, C., and Dabbous, W. (2011). Network characteristics of video streaming traffic. In *Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies*, CoNEXT '11, pages 25:1–25:12, New York, NY, USA. ACM.

Sherry, J., Lan, C., Popa, R. A., and Ratnasamy, S. (2015). Blindbox: Deep packet inspection over encrypted traffic. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, pages 213–226, New York, NY, USA. ACM.

Wang, D., Zhang, L., Yuan, Z., Xue, Y., and Dong, Y. (2014). Characterizing application behaviors for classifying p2p traffic. In *International Conference on Computing, Networking and Communications*, ICNC'14, pages 21–25. IEEE.

Williams, N., Zander, S., and Armitage, G. (2006). A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification. *ACM SIGCOMM Computer Communication Review*, 36(5):5–16.

Yeganeh, S., Eftekhar, M., Ganjali, Y., Keralapura, R., and Nucci, A. (2012). Cute: Traffic classification using terms. In *21st International Conference on Computer Communications and Networks*, ICCCN '12, pages 1–9. IEEE.

Zander, S., Nguyen, T., and Armitage, G. (2005). Automated traffic classification and application identification using machine learning. In *Proceedings of the The IEEE Conference on Local Computer Networks 30th Anniversary*, LCN '05, pages 250–257, Washington, DC, USA. IEEE Computer Society.