Analyzing HTTPS Encrypted Traffic to Identify User's Operating System, Browser and Application

Jonathan Muehlstein*, Yehonatan Zion*, Maor Bahumi†, Itay Kirshenboim*†
Ran Dubin‡, Amit Dvir*, Ofir Pele*†
* Center for Cyber Technologies, Department of Computer Science, Ariel University
† Center for Cyber Technologies, Department of Electrical and Electronics Engineering, Ariel University
‡ Department of Communication Systems Engineering, Ben-Gurion University of the Negev

***Abstract*—Desktops and laptops can be maliciously exploited to violate privacy. There are two main types of attack scenarios: active and passive. In this paper, we consider the passive scenario where the adversary does not interact actively with the device, but he is able to eavesdrop on the network traffic of the device from the network side. Most of the internet traffic is encrypted and thus passive attacks are challenging. In this paper, we show that an external attacker can identify the operating system, browser and application of HTTP encrypted traffic (HTTPS). To the best of our knowledge, this is the first work that shows this. We provide a large data set of more than 20000 examples for this task. Additionally, we suggest new features for this task. We run a through a set of experiments, which shows that our classification accuracy is 96.06%.**

***Index Terms*—Encrypted Traffic, HTTPS, Operating System, Browser, Application**

I. Introduction

There are two main types of attack scenarios: active and passive. Active adversaries try to physically or remotely control the user device. Passive adversaries may violate the privacy of the user by sniffing the network traffic of the devices from the network side. In this work we consider passive attacks.

If network traffic is not encrypted, the task of a passive attacker is simple: he can analyze the payload and read the content of each packet. User activities tracking on the web was proposed in [1]–[3]. This has been done by analyzing unencrypted HTTP requests and responses. A passive adversary may use this information for understanding user actions and revealing information regarding personal interests and habits.

However, most of the internet traffic today is encrypted. This happens both as users start to gain more familiarity with privacy threats and as Google encourages all website owners to switch from HTTP to HTTPS by taking into account whether sites use secure, encrypted connections as a signal in their ranking algorithms [4].

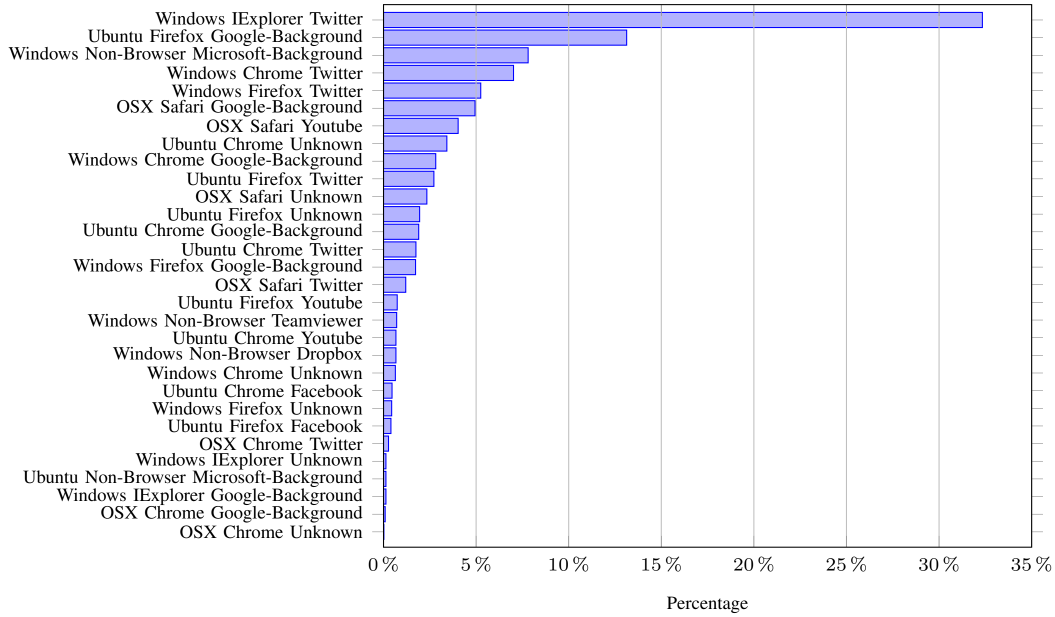
Many works have shown that encryption is not sufficient to protect confidentiality [5]–[39]. Bujlow et al. [27] presented a survey about popular DPI tools for traffic classification. Moore et al. [33] used a Naïve Bayes classifier which is a supervised machine learning approach to classify internet traffic applications. Williams et al. [35] conducted a comparison of five machine learning algorithms that were used to classify Internet traffic applications. Auld et al. [34] proposed to use a supervised Bayesian neural network to classify internet traffic applications. Alshamarri et al. [36] compared AdaBoost, Support Vector Machines, Naïve Bayes, RIPPER and C4.5 in order to classify Skype traffic. Donato et al. [39] presented a method for application classification called the Traffic Identification Engine.

Niemczyk et al. [38] suggested to divide the session to time buckets (10 seconds). The features that were used for each bucket are packet size counts and the time differences between packets. They found the recognition rate of Skype was almost perfect. However, their method was not able to differentiate between browsers and between joint application and browser usage.

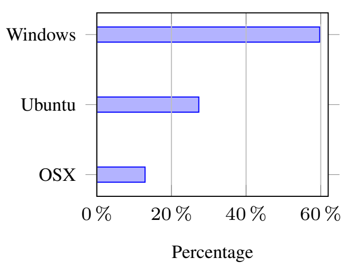
Feature extraction methods for traffic classification include session duration [36], number of packets in a session [32], [40], minimum, maximum and average values of inter-arrival packets time [32], [36], payload size information [32], bit rate [41], [42], round-trip time [41], packet direction [43], and server sent bit-rate [44] that has the advantage of overcoming communication problems such as packet loss and retransmissions.

Liberatore and Levine [45] showed the effectiveness of two traffic analysis techniques for the identification of encrypted HTTP streams. One is based on a naïve Bayes classifier and one on the Jaccards coefficient similarity measure. They also proposed several methods for actively countering the techniques, finding these methods effective, albeit at the cost of a significant increase in the size of the traffic stream. Panchenko et al. [46] showed that a Support Vector Machine (SVM) classifier is able to correctly identify web pages, even when the user used both encryption and anonymization networks such as Tor [47]. Cai et al. [48] presented a web page fingerprinting attack and showed that it is able to overcome defenses such as the application-level defenses HTTPOS [49] and randomized pipelining over Tor.

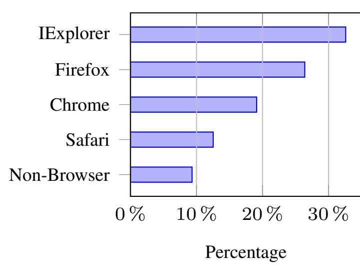
Matousek et al. [50] presented a technique for the identification of the operating system based on TCP parameters and traffic fingerprinting. Husak et al. [51] proposed real-time lightweight identification of HTTPS clients based on network monitoring and SSL/TLS fingerprinting. The fingerprinting is based on pairing HTTP traffic with SSL parameters of encrypted HTTP traffic. In these works, the systems have to identify the SSL parameters and are not robust to changes in the SSL parameters such as cipher suite.



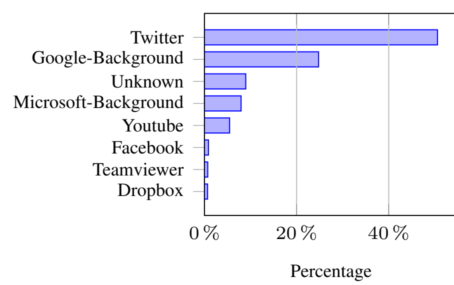
(a) Dataset labels (tuple) statistics



(b) Dataset OS statistics



(c) Dataset browser statistics



(d) Dataset application statistics

Fig. 1: Dataset statistics

Exploiting traffic features for gaining information has been applied not only with the HTTP protocol but also with other protocols. For example, Song et al. [8] showed that some SSH variants are not secure. They showed that simple statistical analysis was able to reveal sensitive information such as login passwords. Additionally, they showed that advanced statistical analysis on timing information can reveal what users type. Another example of a protocol that was shown to be vulnerable is Voice Over IP (VoIP). Wright et al. [14] showed that it is possible to identify spoken phrases by using encrypted VoIP packet length, when variable bit rate (VBR) encoding is used. They used a Hidden Markov model that achieved more than 90% recall and precision. Conti et al. [5] presented various action classifications for a range of applications for mobile devices that achieve high accuracy.

This paper's main contributions are:

- This is the first work that shows how to identify the user's operating system, browser and application from his HTTPS traffic. Inspired by other works presented above, we exploit traffic patterns. Additionally, we present new features that exploit browsers' bursty behavior and SSL behavior. Using the baseline features, the accuracy is 93.51% , while using a combination of baseline and new features achieves accuracy of 96.06%.
- We provide a comprehensive dataset that contains more than 20000 labeled sessions. The operating systems in the dataset are: Windows, Linux-Ubuntu and OSX. The browsers are: Chrome, Internet Explorer, Firefox and Safari. The applications are: YouTube, Facebook and Twitter. The dataset is available for download at [52].

II. Identification of User's Operating System, Browser and Application

The goal of this paper is to identify user operating system, browser and application. To achieve this goal, we use supervised machine learning techniques. Supervised machine learning techniques learn a function that given a sample returns a label. The learning is carried out using a dataset of labeled samples. In our case, we chose to use sessions as samples where a session is the tuple <Protocol, IP source, IP destination, Port source, Port destination> and the label is the tuple <OS, Browser, Application>. Thus, our task is inherently a multiclass learning with 30 classes (see Figure 1a for the labels and their statistics in the dataset).

The rest of this section is organized as follows: In section II-A we describe how we collected the dataset and the dataset characteristics. In section II-B we describe and discuss our feature extraction scheme. Finally, in section II-C we describe the machine learning methodology we used.

A. Dataset

We used the Selenium web automation tool [53] to develop crawlers for gathering the dataset. We gathered all the traffic that passed through port 443 (TLS/SSL). Finally, we split the traffic into sessions using SplitCap [54].

For YouTube and Facebook traffic, we used the crawler on a standard internet connection over various operating systems and various browsers and combinations thereof. For Facebook, the same account was used both for sending and receiving posts. For Twitter, we had one sending account and several receiving accounts (followers) where they ranged over various operating systems and various browsers and combinations thereof. Teamviewer's traffic was generated by us actively without a crawler. In addition to our active traffic, we also observed background traffic that operating systems, browsers and applications created (Google-Background, Microsoft-Background). Dropbox traffic was composed both of active (no crawler) and background traffic.

Any traffic that we could not recognize was labeled as unknown. The browser label part of the tuple of stand alone applications which do not work under a browser (*e.g.* Dropbox, Teamviewer) were labeled as Non-Browser.

The dataset was collected over the period of more than two months in our research lab over diverse connections (wired and WiFi) and networks conditions (over workdays and weekends 24/7).

Our dataset contains more than 20000 sessions. The tuple labels statistics can be seen in Figure 1a. Operating system, browser, application statistics can be seen in Figures 1b,1c,1d correspondingly.

B. Feature Extraction

This section describes how we extract features from a session and the feature characteristics. Encrypted traffic generally relies on SSL/TLS for secure communication. These protocols are built on top of the TCP/IP suite. The TCP layer receives encrypted data from the above layer and divides data into chunks if the packets exceed the Maximum Segment Size (MSS). Then, for each chunk it adds a TCP header creating a TCP segment. Each TCP segment is encapsulated into an Internet Protocol (IP) datagram. As TCP packets do not include a session identifier, we identify a session using the tuple <Protocol, IP source, IP destination, Port source, Port destination>.

Forward packets
Forward total Bytes
Min forward inter arrival time difference
Max forward inter arrival time difference
Mean forward inter arrival time difference
STD forward inter arrival time difference
Mean forward packets
STD forward packets
Backward packets
Backward total Bytes
Min backward inter arrival time difference
Max backward inter arrival time difference
Mean backward inter arrival time difference
STD backward inter arrival time difference
Mean backward packets
STD backward packets
Mean forward TTL value
Minimum forward packet
Minimum backward packet
Maximum forward packet
Maximum backward packet
Total packets
Minimum packet size
Maximum packet size
Mean packet size
Packet size variance

(a) base features

TCP initial window size
TCP window scaling factor
SSL compression methods
SSL extension count
SSL chiper methods
SSL session ID len
Forward peak MAX throughput
Mean throughput of backward peaks
Max throughput of backward peaks
Backward min peak throughput
Backward STD peak throughput
Forward number of bursts
Backward number of bursts
Forward min peak throughput
Mean throughput of forward peaks
Forward STD peak throughput
Mean backward peak inter arrival time diff
Minimum backward peak inter arrival time diff
Maximum backward peak inter arrival time diff
STD backward peak inter arrival time diff
Mean forward peak inter arrival time diff
Minimum forward peak inter arrival time diff
Maximum forward peak inter arrival time diff
STD forward peak inter arrival time diff
Keep alive packets
TCP Maxiumu Segment Size
Forward SSL Version

(b) new features

TABLE I: The two sets of features used in this paper. The base features are features that are used in many traffic classification methods. The new features are proposed in this paper.

A session contains two flows: forward and backward. A flow is defined as time ordered sequence of TCP packets during a single TCP session. The forward flow is defined as a time series bytes transported by incoming packets only, while the backward flow is defined as a time series bytes transported by

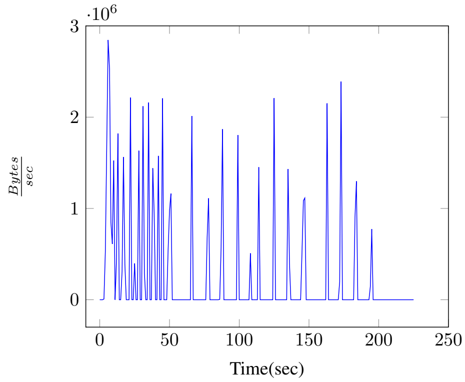


Fig. 2: An example of the bursty behavior of browser traffic.

outgoing packets only. We use the forward, backward and their combination as a representation of a connection. Additionally we also use time series features such as inter arrival time differentials between different packets on the same flow.

The feature extraction takes as an input the session network traffic and extracts features from it. In this paper, we consider two sets of features and their combination. First, a typical feature set, used in many traffic classification methods [33]–[38], [50], which we call "base features" is presented in Table Ia.

Second, we present a new set of features in Table Ib, which we call "new features". This set of features is based on a comprehensive network traffic analysis, in which we tried to identify traffic parameters that differentiate between different operating systems and browsers. Our new set is robust to changes (not based on a small set of strong parameters or fingerprint). The set of features include new SSL features, new TCP features and the bursty behavior of the browsers (peaks) which is defined as a section of traffic where there is silence before and after. This section of traffic is called a peak. An example of the bursty behavior of browsers is depicted in Figure 2. Our previous works also used these peak features to classify video titles [55] and quality representation [56]. Note that, the bursty behavior of browser traffic was observed for YouTube traffic in [57], [58].

C. Learning

In this section we describe our machine learning methodology. We chose to use the Support Vector Machine (SVM) [59] with Radial Basis Function (RBF) as the kernel function because it has excellent performance in many machine learning applications. We trained and tested on 70% train and 30% test splits five times and accuracy is reported as the average of these experiments.

First, features were scaled between zero and one at training and the same scaling factors were used for the test set. 5-fold cross validation was used for choosing both the regularization parameter of SVM, C, over the set $\{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ and for the gamma parameter of RBF, over the set $\{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$. We used LIBSVM [60] to train and test our data set.

III. Results

The accuracy for the tuple <OS, Browser, Application> classification with the three feature sets: base features,new features,base + new features is presented in Figure 3.

There are three main observations: First, the tuple <OS, Browser, Application> classification of encrypted classification is possible with high accuracy. Second, using our new features the results are comparable. Finally, in all experiments, using the base + new features achieved the best results. For tuple classification the addition of our new features increased accuracy from 93.52% to 96.06%.

Confusion matrices are depicted in Fig 4. As above, it can be seen that the classification is almost perfect where most of the mistakes are due to the unknown label which can actually be a correct answer that we cannot verify.

IV. Conclusions and Future Work

The framework proposed in this paper is able to classify encrypted network traffic and to infer which operating system, browser and application the user is using on his desktop or laptop computer. We show that despite the use of SSL/TLS, our traffic analysis approach is an effective tool. An eavesdropper can easily leverage the information about the user to fit an optimal attack vector.

A passive adversary may also collect statistics about groups of users for improving their marketing strategy. In addition, an attacker may use tuples statistics for identifying a specific person.

An interesting extension of this work would be to add action classification (*e.g.* send a tweet, receive a post) to the tuple as has been done for application and action for mobile devices [5]. Another interesting extension would be to extend operating system and browser classification to the mobile world.

Acknowledgment

This research was supported by the InfoMedia consortium.

References

[1] Richard Atterer, Monika Wnuk, and Albrecht Schmidt. Knowing the user's every move: User activity tracking for website usability evaluation and implicit interaction. In *Proceedings of the 15th International Conference on World Wide Web*, pages 203–212, 2006.

[2] Fabrcio Benevenuto, Tiago Rodrigues, Meeyoung Cha, and Virglio Almeida. Characterizing user navigation and interactions in online social networks. *Information Sciences*, 195:1 – 24, 2012.

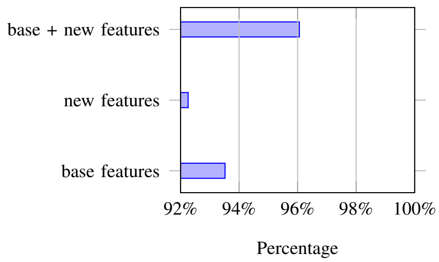
[3] Fabian Schneider, Anja Feldmann, Balachander Krishnamurthy, and Walter Willinger. Understanding online social network usage from a network perspective. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, pages 35–48, 2009.

[4] Google. Google webmaster central blog: Https as a ranking signal, 2014.

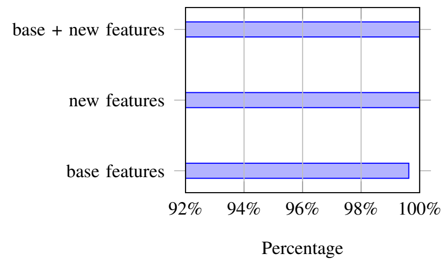
[5] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde. Analyzing android encrypted network traffic to identify user actions. *IEEE Transactions On Information Forensics and Security*, 2016.

[6] M. Crotti, F. Gringoli, P. Pelosato, and L. Salgarelli. A statistical approach to IP level classification of network traffic. In *International Conference on Communications*, pages 170–176, June 2006.

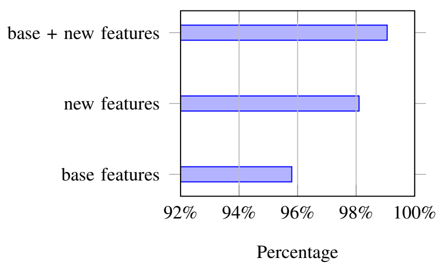
[7] T. Okabe, T. Kitamura, and T. Shizuno. Statistical traffic identification method based on flow-level behavior for fair VoIP service. In *IEEE Workshop on VoIP Management and Security*, pages 35–40, April 2006.



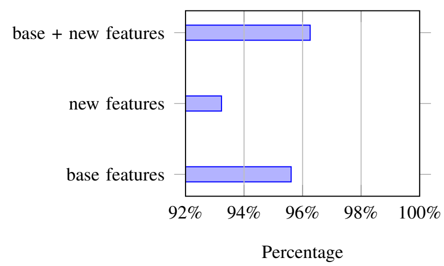
(a) Tuple Accuracy Results



(b) OS Accuracy Results



(c) Browser Accuracy Results



(d) Application Accuracy Results

Fig. 3: Accuracy results for SVM-RBF with different features set. This work is the first to show that the tuple <OS, Browser, Application> classification is possible with high accuracy. Note that, naive classification based on the data set statistics will have only 32.34% accuracy (<Windows, IExplorer, Twitter>). Moreover, adding our new features increase the accuracy to 96.06%.

Predicted labels

Real labels

	Windows IExplorer Twitter	Ubuntu Firefox Google-Background	Windows Non-Browser Microsoft-Background	Windows Chrome Twitter	Windows Firefox Twitter	OSX Safari Google-Background	OSX Safari Youtube	Ubuntu Chrome Unknown	Windows Chrome Google-Background	Ubuntu Firefox Twitter	OSX Safari Unknown	Ubuntu Firefox Unknown	Ubuntu Chrome Google-Background	Ubuntu Chrome Twitter	Windows Firefox Google-Background	OSX Safari Twitter	Ubuntu Firefox Youtube	Windows Non-Browser Teamviewer	Ubuntu Chrome Youtube	Windows Non-Browser Dropbox	Windows Chrome Unknown	Ubuntu Chrome Facebook	Windows Firefox Unknown	Ubuntu Firefox Facebook	OSX Chrome Twitter	Windows IExplorer Unknown	Ubuntu Non-Browser Microsoft-Background	Windows IExplorer Google-Background	OSX Chrome Google-Background	OSX Chrome Unknown
Windows IExplorer Twitter	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Firefox Google-Background	0	.97	0	0	0	0	0	0	0	0	0	0	.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Windows Non-Browser Microsoft-Background	0	0	.99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Windows Chrome Twitter	0	0	0	.99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.01	0	0	0	0	0	0	0	0	0
Windows Firefox Twitter	0	0	0	0	.98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.02	0	0	0	0	0	0	0
OSX Safari Google-Background	0	0	0	0	0	.92	.04	0	0	0	.02	0	0	0	0	.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OSX Safari Youtube	0	0	0	0	0	.02	.97	.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Chrome Unknown	0	0	0	0	0	0	0	.84	0	0	0	0	.07	.04	0	0	0	0	.01	0	0	.03	0	0	0	0	0	0	0	0
Windows Chrome Google-Background	0	0	.01	.03	0	0	0	0	.94	0	0	0	0	0	.02	0	0	0	0	0	.01	0	0	0	0	0	0	0	0	0
Ubuntu Firefox Twitter	0	0	0	0	0	0	0	0	0	.95	0	.03	0	0	0	0	.01	0	0	0	0	0	0	0	0	0	0	0	0	0
OSX Safari Unknown	0	0	0	0	0	.06	.01	0	0	0	.91	0	0	0	0	.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Firefox Unknown	0	.02	0	0	0	0	0	0	0	.08	0	.87	0	0	0	0	.01	0	0	0	0	0	0	.03	0	0	0	0	0	0
Ubuntu Chrome Google-Background	0	.07	0	0	0	0	0	.18	0	0	0	0	.73	0	0	0	0	0	.02	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Chrome Twitter	0	.02	0	0	0	0	0	.08	0	0	0	0	.03	.84	0	0	0	0	.01	0	0	.01	0	0	0	0	0	0	0	0
Windows Firefox Google-Background	0	0	0	.01	0	0	0	0	.01	0	0	0	0	0	.97	0	0	0	0	0	0	0	.01	0	0	0	0	0	0	0
OSX Safari Twitter	0	0	0	0	0	0	.06	0	0	0	.03	0	0	0	0	.91	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Firefox Youtube	0	.02	0	0	0	0	0	0	0	.02	0	.02	0	0	0	0	.93	0	0	0	0	0	0	0	0	0	0	0	0	0
Windows Non-Browser Teamviewer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Chrome Youtube	0	0	0	0	0	0	0	.07	0	0	0	0	.13	.04	0	0	0	0	.74	0	0	.02	0	0	0	0	0	0	0	0
Windows Non-Browser Dropbox	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Windows Chrome Unknown	0	0	.02	.09	0	0	0	0	.02	0	0	0	0	0	0	0	0	0	0	0	.86	0	0	0	0	0	0	0	0	0
Ubuntu Chrome Facebook	0	0	0	0	0	0	0	.3	0	0	0	0	.04	0	0	0	0	0	0	0	0	.67	0	0	0	0	0	0	0	0
Windows Firefox Unknown	0	0	.06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.94	0	0	0	0	0	0	0
Ubuntu Firefox Facebook	0	.06	0	0	0	0	0	0	0	.11	0	.28	0	0	0	0	0	0	0	0	0	0	0	.56	0	0	0	0	0	0
OSX Chrome Twitter	0	0	0	0	0	0	0	.13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.75	0	0	0	.06	.06
Windows IExplorer Unknown	.71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.29	0	0	0	0
Ubuntu Non-Browser Microsoft-Background	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Windows IExplorer Google-Background	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
OSX Chrome Google-Background	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
OSX Chrome Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fig. 4: Confusion matrices (rows are ground truth). For most tuples the classification is almost perfect. Exceptions happens mostly between similar tuples and the unknown classes (which can actually be a correct answer that we cannot verify). For example, “Ubuntu Chrome Google-Background” is mistakenly classified as “Ubuntu Chrome Unknown” in 18% of the cases and “Ubuntu Firefox Google-Background” in 7%. The total accuracy is to 96.06%

[8] Dawn Xiaodong Song, David Wagner, and Xuqing Tian. Timing analysis of keystrokes and timing attacks on ssh. In *Proceedings of the 10th Conference on USENIX Security Symposium - Volume 10*, pages 25–25, 2001.

[9] Brice Canvel, Alain Hiltgen, Serge Vaudenay, and Martin Vuagnoux. Password interception in a ssl/tls channel. In *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 583–599. Springer Berlin Heidelberg, 2003.

[10] T. Scott Saponas, Jonathan Lester, Carl Hartung, Sameer Agarwal, and Tadayoshi Kohno. Devices that tell on you: Privacy trends in consumer ubiquitous computing. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, pages 5:1–5:16, 2007.

[11] Yali Liu, C. Ou, Zhi Li, C. Corbett, B. Mukherjee, and D. Ghosal. Wavelet-based traffic analysis for identifying video streams over broadband networks. In *IEEE Global Telecommunications Conference*, pages 1–6, Nov 2008.

[12] Yali Liu, Ahmad-Reza Sadeghi, Dipak Ghosal, and Biswanath Mukherjee. Video streaming forensic content identification with traffic snooping. In *Information Security*, volume 6531 of *Lecture Notes in Computer Science*, pages 129–135. Springer, 2011.

[13] AM. White, AR. Matthews, KZ. Snow, and F. Monrose. Phonotactic reconstruction of encrypted voip conversations: Hookt on fon-iks. In *SP*, 2011.

[14] CV. Wright, L. Ballard, F. Monrose, and GM. Masson. Language identification of encrypted voip traffic: Alejandra y roberto or alice and bob? In *USENIX Security*, 2007.

[15] A. Dainotti, A. Pescape, and KC. Claffy. Issues and future directions in traffic classification. *Network, IEEE*, 2012.

[16] Vern Paxson. Empirically derived analytic models of wide-area tcp connections. *TON*, 1994.

[17] R. Alshammari and AN. Zincir-Heywood. Unveiling skype encrypted tunnels using gp. In *CES*, 2010.

[18] S. Zander, T. Nguyen, and G. Armitage. Self-learning ip traffic classification based on statistical flow characteristics. In *PANM*, 2005.

[19] I. Paredes-Oliva, I. Castell-Uroz, P. Barlet-Ros, X. Dimitropoulos, and J. Sole-Pareta. Practical anomaly detection based on classifying frequent traffic patterns. In *INFOCOM WKSHPS*, 2012.

[20] D. Zhang, C. Zheng, H. Zhang, and H. Yu. Identification and analysis of skype peer-to-peer traffic. In *ICIW*, 2010.

[21] D. Bonfiglio, M. Mellia, M. Meo, and D. Rossi. Detailed analysis of skype traffic. *Multimedia, IEEE Transactions on*, 2009.

[22] KT. Chen, CY. Huang, P. Huang, and CL. Lei. Quantifying skype user satisfaction. In *CCR*, 2006.

[23] E. Hjelmvik and W. John. Statistical protocol identification with spid: Preliminary results. In *SNCNW*, 2009.

[24] R. Bar-Yanai, M. Langberg, D. Peleg, and L. Roditty. Realtime classification for encrypted traffic. In *EA*, 2010.

[25] S. Valenti, D. Rossi, A. Dainotti, A. Pescapè, A. Finamore, and M. Mellia. Reviewing traffic classification. In *DTMA*, 2013.

[26] Z. Cao, G. Xiong, Y. Zhao, Z. Li, and L. Guo. A survey on encrypted traffic classification. In *ATIS*, 2014.

[27] P. Barlet-Ros T. Bujlowa, V. Carela-Espaolb. Independent comparison of popular dpi tools for traffic classification. *Computer Networks*, 76:75–89, 2015.

[28] T. Nguyen and G. Armitage. A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys and Tutorials*, 10:56–76, 2008.

[29] R. Bar Yanai, M. Langberg, D. Peleg, and L. Roditty. Realtime classification for encrypted traffic. In *Experimental Algorithms*, volume 6049 of *Lecture Notes in Computer Science*, pages 373–385. Springer Berlin Heidelberg, 2010.

[30] M. Calisti, S. Meer, and J. Strassner. *Advanced Autonomic Networking and Communication (Whitestein Series in Software Agent Technologies and Autonomic Computing)*. Birkhäuser Basel, 1 edition, 2008.

[31] M. Korczynski and A. Duda. Classifying service flows in the encrypted skype traffic. In *ICC*, pages 1064–1068, June 2015.

[32] D. Zhang, C. Zheng, H. Zhang, and H. Yu. Identification and analysis of skype peer-to-peer traffic. In *Internet and Web Applications and Services*, pages 200–206, May 2010.

[33] Andrew W. Moore and Konstantina Papagiannaki. Toward the accurate identification of network applications. In *Proceedings of the 6th International Conference on Passive and Active Network Measurement*, PAM'05, pages 41–54, 2005.

[34] Andrew W. Moore and Denis Zuev. Internet traffic classification using bayesian analysis techniques. In *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 50–60, 2005.

[35] Nigel Williams, Sebastian Zander, and Grenville Armitage. A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification. *SIGCOMM Comput. Commun. Rev.*, 36(5):5–16, Oct 2006.

[36] R. Alshammari and A.N. Zincir-Heywood. Unveiling skype encrypted tunnels using gp. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8, July 2010.

[37] Anthony McGregor, Mark Hall, Perry Lorier, and James Brunskill. Flow clustering using machine learning techniques. In Chadi Barakat and Ian Pratt, editors, *Passive and Active Network Measurement*, volume 3015 of *Lecture Notes in Computer Science*, pages 205–214. Springer Berlin Heidelberg, 2004.

[38] B. Niemczyk and P.Rao. Identification over encrypted channels. In *blackHat*, Aug. 2014.

[39] W. de Donato, A. Pescap, and A. Dainotti. Traffic Identification Engine: An Open Platform for Traffic Classification. *IEEE Network*, 28(2):56–64, Mar 2014.

[40] I. Paredes-Oliva, I. Castell-Uroz, P. Barlet-Ros, X. Dimitropoulos, and J. Sole-Pareta. Practical anomaly detection based on classifying frequent traffic patterns. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 49–54, March 2012.

[41] Kuan-Ta Chen, Chun-Ying Huang, Polly Huang, and Chin-Laung Lei. Quantifying skype user satisfaction. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '06, pages 399–410, 2006.

[42] D. Bonfiglio, M. Mellia, M. Meo, and D. Rossi. Detailed analysis of skype traffic. *Multimedia, IEEE Transactions on*, 11(1):117–127, Jan 2009.

[43] E. Hjelmvik and W. John. Statistical protocol identification with spid: Preliminary results. In *Swedish National Computer Networking Workshop*, pages 399–410, 2009.

[44] Roni Bar Yanai, Michael Langberg, David Peleg, and Liam Roditty. Realtime classification for encrypted traffic. In Paola Festa, editor, *Experimental Algorithms*, volume 6049 of *Lecture Notes in Computer Science*, pages 373–385. Springer Berlin Heidelberg, 2010.

[45] Marc Liberatore and Brian Neil Levine. Inferring the source of encrypted http connections. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pages 255–263, 2006.

[46] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *ACM WP*, 2011.

[47] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *USENIX-SS*, 2004.

[48] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a distance: Website fingerprinting attacks and defenses. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 605–616, 2012.

[49] Xiapu Luo, Peng Zhou, Edmond W. W. Chan, Wenke Lee, Rocky K. C. Chang, and Roberto Perdisci. Httpos: Sealing information leaks with browser-side obfuscation of encrypted flows. In *In Proc. Network and Distributed Systems Symposium (NDSS)*, 2011.

[50] P. Matousek, O. Rysavy, M. Gregrand, and M. Vymlatil. Towards identification of operating systems from the internet traffic. In *5th International Conference on Data Communication Networking*, pages 21–27, July 2014.

[51] Martin Husak, Milan Čermák, Tomáš Jirsík, and Pavel Čeleda. Https traffic analysis and client identification using passive ssl/tls fingerprinting. *EURASIP Journal on Information Security*, (1):1–14, 2016.

[52] Dataset. The research dataset. http://www.ariel.ac.il/sites/amitd/ad-data-sets.

[53] Selenium. Selenium automates browsers. http://www.seleniumhq.org/. Accessed: 2016-02-28.

[54] SplitCap. Splitcap - pcap file splitter. http://www.netresec.com/?page=SplitCap. Accessed: 2016-02-28.

[55] Ran Dubin, Amit Dvir, Ofir Pele, and Ofer Hadar. I know what you saw last minute-encrypted http adaptive video streaming title classification. *arXiv preprint arXiv:1602.00490*, 2016.

[56] Ran Dubin, Amit Dvir, Ofir Pele, and Ofer Hadar. Real time video quality representation classification of encrypted http adaptive video streaming-the case of safari. *arXiv preprint arXiv:1602.00489*, 2016.

[57] Pablo Ameigeiras, Juan J. Ramos-Muoz, Jorge Navarro-Ortiz, and Juan M. Lpez-Soler. Analysis and modelling of youtube traffic. *TETT*, 2012.
[58] Ashwin Rao, Arnaud Legout, Yeon-sup Lim, Don Towsley, Chadi Barakat, and Walid Dabbous. Network characteristics of video streaming traffic. In *CoNEXT*, 2011.
[59] C. Cortes and V. Vapnik. Support vector machine. *Machine learning*, 1995.
[60] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *IST*, 2011.