Encrypted Traffic Detection: Beyond the Port Number Era

Hossein Doroud, Ahmad Alaswad, Falko Dressler
School of Electrical Engineering and Computer Science, TU Berlin
doroud@tu-berlin.de, alaswad@campus.tu-berlin.de, dressler@ccs-labs.org

***Abstract*—Internet service providers (ISP) rely on network traffic classifiers to provide secure and reliable connectivity for their users. Encrypted traffic introduces a challenge as attacks are no longer viable using classic Deep Packet Inspection (DPI) techniques. Distinguishing encrypted from non-encrypted traffic is the first step in addressing this challenge. Several attempts have been conducted to identify encrypted traffic. In this work, we compare the detection performance of DPI, traffic pattern, and randomness tests to identify encrypted traffic in different levels of granularity. In an experimental study, we evaluate these candidates and show that a traffic pattern-based classifier outperforms others for encryption detection.**

***Index Terms*—Deep Packet Inspection; Network Traffic Classification; Privacy; Ground Truth; Mobile Internet Ecosystem**

I. Introduction

Network monitoring is an essential tool for Internet Service Providers (ISP) to enhance their functionality such as Quality of Service (QoS) and Security [1]–[4]. Currently, Deep Packet Inspection-based (DPI) classifiers are the main tools in the hands of ISPs in this regard. These classifiers look into the user payloads to detect a string that is unique for an application protocol or an attack. The increased use of encrypted communication affects the performance of DPI classifiers as the user payload is not exposed clearly to the classifiers anymore.

Several parameters of traffic flows have been investigated to identify encrypted traffic [5], [6]. Reliance on the traffic pattern with the help of Machine Learning (ML) algorithms is among the most promising approaches which are applicable in the the field [7], [8]. However, Deri et al. [9] showed that it is not always suitable for mission-critical operations.

Randomness characteristics of the user payloads also can provide some information regarding the encryption property of a flow [10]. It relies on the fact that an efficient encryption algorithm should randomize the user payload significantly. Therefore, it can be considered an appropriate approach for splitting encrypted traffic from un-encrypted in the case that each traffic category should be treated differently.

Despite all the efforts, there is a lack of work that compares the performance of Deep Packet Inspection (DPI), traffic pattern, and randomness test classifiers to deal with encrypted traffic. To this end, first, we propose a randomness test-based classifier and then compare its performance with the well-selected candidates from DPI and traffic pattern classifiers. The selected classifiers compete with each other to classify our ground truth in three different levels of granularity: i) binary classification to distinguish encrypted from un-encrypted traffic, ii) application protocol classification to distinguish between encrypted application protocols and un-encrypted application protocols, and iii) classifying the flows according to the content and identifying encrypted content among un-encrypted contents. We also propose a traffic generator that can provide a ground truth with labels in three different levels of granularity.

Our main contributions can be summarized as follows:

- We propose a randomness test-based network traffic classification and make it is publicly available for the research community [1].
- We study the time consumption of several randomness tests which applicable in the field of network traffic classification.
- Also, we measure and compare the performance of DPI, traffic pattern, and randomness test classifiers to detect the encryption traffic in granularity levels binary, protocol, and content.

II. Related Work

Although distinguishing encrypted traffic based on the port number is the most straightforward approach, it is not very accurate in modern Internet traffic [11].

In the network monitoring community, DPI has become the main approach for identifying traffic, including encrypted traffic. nDPI[2] is an upgraded version of OpenDPI, which has been developed by Deri et al. [9]. Using the unencrypted part of packets (application payload, headers, etc), it can identify applications. As DPI is a resource-demanding process, Alcock et al. [12] proposed `Libprotoident`,[3] which processes only a specific amount of traffic. Despite the high accuracy of DPI classifiers, their detection performance is limited to the scope of their signature set. Also, the processing performance is rather limited.

An alternative is the use of Deep Learning (DL) to identify encrypted traffic. Rezaei and Liu [13] overviewed the capability of DL to identify encrypted traffic. They discussed the implementation of DL in the field from different perspectives such as data collection, data preprocessing, and feature selection. Aceto et al. [6] proposed DISTILLER as a multimodal multitask deep learning architecture [14] to classify encrypted traffic. Their

[1] https://github.com/tkn-tub/encryption_detection

[2] https://www.ntop.org/products/deep-packet-inspection/ndpi/

[3] https://github.com/wanduow/libprotoident

measurements on human-generated ground truth data showed a performance boost of up to 7.9% better in comparison with the state of the art for multi-task architectures.

Finally, the entropy of data can be used to identify encrypted traffic. Cheng et al. [5] target the encrypted part of traffic to do the identification. They proposed a classifier based on N-gram entropy and cumulative sum test (cumsum) [15]. Their measurements show that their proposal identified encrypted data in embedded files, such as pictures and compressed files. Casino et al. [16] proposed HEDGE (High Entropy DistinGuishEr) for classifying the high entropy files such as MPEG-1 Audio Layer 3 (MP3) and Portable Document Format (PDF) as well as encrypted files. This method was based on the evaluation of the randomness of the data streams and by using a tree-based threshold system.

Despite all of the above studies, it is still unclear which approach performs best in different scenarios. We assess and compare the performance of DPI, traffic pattern, and randomness tests to shed light on this question.

III. Detection Methods

We select a well-representative instance from DPI traffic pattern and randomness test-based approaches to compare their performance in different scenarios. We particularly emphasize on the randomness tests as these have found little attention from the network monitoring community so far.

1) Deep Packet Inspection: DPI classifier is well-known because of its high accuracy and low false-positive rate [17]. It is equipped with a set of signatures of either application protocols, or attacks and looks for them in the network traffic with the help of a string matching algorithm. This limits the detection scope of DPI to its signature sets and makes it a resource-demanding approach.

Encryption protocols randomize the bitstream of user payloads and hide the content from the DPI eye. Therefore, processing the encryption traffic is challenging for a DPI classifier. Some research works [18], [19] proposed to apply decryption techniques before DPI. Other works [9], [12] considered a different approach and investigated the unencrypted part of flows (e.g. headers, SSL certificate).

Several open-source classifiers were developed based on DPI principle (i.e., L7, nDPI). However, Bujlow et al. [20] indicated the superiority of `Libprotoident` [12] as a lightweight and highly accurate DPI classifier. The key factor that makes `Libprotoident` very fast is that it processes only the first four bytes of payloads in each direction instead of the entire packet payload. We choose `Libprotoident` as the representative of DPI and apply it with its default setting.

2) Traffic Pattern: Different application protocols including encryption protocols generate traffic with different patterns. Therefore, it is possible to identify the application protocol of a flow from its pattern.

Detection based on traffic pattern mainly relies on ML techniques and follows a training and testing paradigm. An ML algorithm learns to distinguish between traffic from different applications in the training phase and classifies the rest of the traffic accordingly in the testing phase. The classifiers developed based on traffic pattern cannot process the raw traffic directly. Therefore, in the preprocessing phase, the feature that represents the pattern characteristics of flows is extracted from the raw traffic. The required information for extracting the features determines the offline or online operational capability of the classifier [21]. E.g. if the duration of flow is among the features in the defined feature set, then the feature extraction cannot be completed before the end of the flow and consequently, the classifier cannot process the flow before its end.

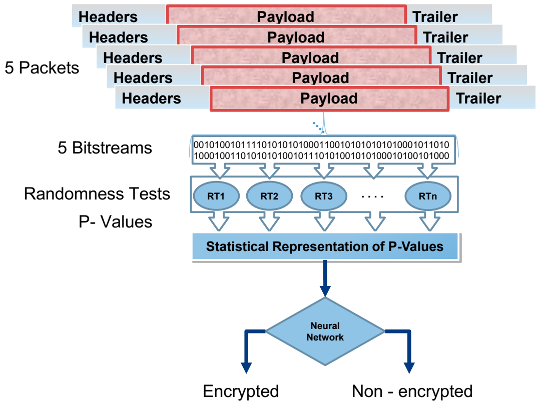

Fig. 1. Work flow of the proposed Randomness Test-based classifier

Since the online performance of the classifier is our point of interest, we consider only the features that can be extracted from the first packets of a flow. Our feature set consists of the statistics (min, max, mean, standard deviation) of the packet size and the inter-arrival time of the first five packets. TIE [22] is used to extract the mentioned features from a flow in each direction as stated by several works like [23].

Since Artificial Neural Network (ANN) has been heavily applied to the field of network classification in the recent years [6], [24], it has been selected as the ML algorithm for learning the pattern of encrypted traffic.

3) Randomness Tests: Encryption protocols are intended to convert the user payload to a randomized bitstream. Consequently, several tests have been developed to evaluate the quality of randomness of bitstreams that an encryption protocol generates. As we could not find any implementation of a Randomness Test-based classifier in the public domain, we propose and implement a novel Randomness Test-based classifier. Figure 1, depicts the structure of our proposal.

Our proposal processes only the payloads of the first five captured packets of a flow. Theoretically, any bitstream with high randomness characteristics can be considered an encrypted stream. However, there are several contents like MP3 or PDF that generate highly randomized bitstreams. To this end, the proposed classifier includes multiple Randomness Tests to measure the randomness of a bitstream from different perspectives such as distribution of 1 and 0 or pattern repetition. By taking the required size of a bitstream and the complexity of the Randomness Test into account, we find the

TABLE I
THE SIZE OF ANN IN DIFFERENT EXPERIMENTS, (#HIDDEN LAYER, #NEURONE,)

	Binary	Protocol	Content
Traffic Pattern	(6, 100)	(2, 60)	(2, 100)
Randomness Test	(5, 100)	(4, 60)	(4, 80)

following tests appropriate for processing the network traffic: Monobit, Frequency Test within a Block (BlockFreq), Runs Test, Approximate Entropy Test (AprEnt), Test for the Longest Run of Ones in a Block (LRuns), Cusum, Serial Test (Ser), DFT, Topological Binary Test (TBT), Greatest Common Divisor test (GCD) and Book Stack test (BckStack) [15].

Each Randomness Test processes the five payloads separately and generates a P-Value for the correspondent payload. The P-Value is in the range of [0,1] [15]. Following, the mean value and the standard deviation of P-Values derived from the payloads of a flow are calculated and considered as the final output of each Randomness Test. Finally, a trained ANN classifies the flow according to the Randomness Tests outputs.

IV. EVALUATION

We design three experiments to compare the performance of the considered classifiers in different granularity levels. At the highest level, we do a *binary classification* to distinguish the encrypted flows from unencrypted ones. At the next level, the classifiers classify the flows according to their application protocol or by doing *Protocol classification*. Here, there are encrypted protocols (e.g., Secure Shell (SSH)) as well as unencrypted protocols (e.g., File Transfer Protocol(FTP)). At the lowest level of the granularity, we run a *content classification* and measure their performance to detect encrypted content among others such as MP3 and text file (txt).

The size of ANN is defined according to the try and error strategy and separately for each measurement. Each hidden layer has the same number of neurons. Table I reports the number of hidden layers and their neurons in the tuple format. Softmax and ReLu [25] are selected as the activation functions of neurons in hidden layers and I/O layers, respectively. The number of neurons for the input layer is 16, correspondent to the number of features that we have in both traffic pattern and randomness test-based classifiers. The number of output classes defines the number of output neurons that varies across different experiments. We follow 2-fold cross-validation for testing and training the ANN.

The classifiers performance is measured based on *Recall* (equation 1), *Precision* (Equation 2) and *F1* (Equation 3) [23]. Equations 1, 2, and 3 explain the definition of the parameters where TP, FN and FP represent true positive, false negative and false positive, respectively. The values of the considered parameters are in the range of [0, 1] where 1 indicates the maximum performance of a classifier.

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

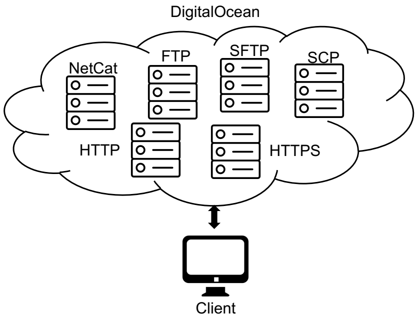

Fig. 2. Schematic of the ground-truth generator

TABLE II
THE DISTRIBUTION OF APPLICATION PROTOCOLS AVAILABLE IN GROUND-TRUTH

Protocol	# Flows	# Packets
FTP	5.1k	207.9k
NETCAT	5k	87.8k
HTTP	5k	343.7k
SCP	5k	423.6k
HTTPS	4.9k	373.7k
SFTP	4.9k	735.6k
Total	29.3k	2.3M

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

A. Ground-truth Generator

Although there are publicly available datasets, it is hard to obtain one with the user payload and labels in different resolution levels. Therefore we propose a testbed Ground-Truth Generator (GTG) which generates the required dataset for the measurements.

GTG consists of six servers for different protocols (Table II) that host an identical set of contents in different types (Table III). To this end, we use six servers from the public cloud platform DigitalOcean [4]. Also there is a client which plays a critical role in controlling the volume of traffic originating from different protocols and contents. It selects a pair of protocols and content, randomly. Following this, the client sends the request for the pair to the correspondent server and captures the following traffic. As the client has the information of the requests and their correspondent traffic, we can label the traffic with its protocol and the payload content.

Tables II and III report the distribution of the protocols and the contents exists in our ground-truth. Also, the ground-truth is publicly available for researchers [26].

B. Randomness Test Classifier Quality Performance

It is common to deploy an encryption detector in a high-speed network environment. In such a case, the processing speed is among the key factors to select an appropriate classifier. Despite some research works in [12], [23], [27] that reported the speed

[4] https://www.digitalocean.com/

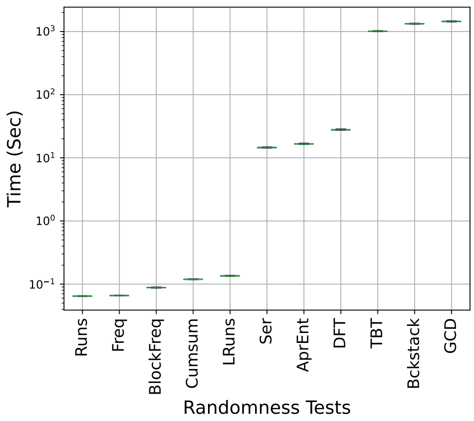

Fig. 3. Time consumption of randomness tests to process the ground truth

of DPI and traffic pattern classifiers, we could not find any measurement regarding the classification speed of a randomness test classifier. Therefore, we arrange an experiment to study the classification time required by the considered randomness tests for processing our dataset. The experiment is repeated five times for each randomness test to minimize the effect of any undesirable artifact.

Figure 3 reports the median, whiskers (upper and lower), upper quartile, and the lower quartile of the randomness tests' time consumption in five iterations. As shown in the Figure, there are slight differences in the time consumption among the different iterations.

Remarkably, the randomness tests TBT, GCD, and BckStack are drastically slower than the rest. The reason is that all the three randomness tests are implemented in R. Even though the R instance is initialized only once in the application, processing an R module for each randomness test is costly and increases the overall processing time significantly. We exclude the R implemented randomness tests (TBT, GCD and BckStack) from the rest of our measurements due to their low process speed.

TABLE III
THE DISTRIBUTION OF DIFFERENT CONTENTS IN GROUND-TRUTH

Content Type	# Flows	# Packets
au	1.2k	18k
.txt	1.1k	21,9k
.mp3	1.2k	25.8k
.pdf	1.2k	82.3k
.wav	1.2k	73.5k
.png	1.2k	33.8k
.xls	1.2k	66.7k
.csv	1.1k	25.6k
.webm	1.2k	35.7k
.mat	1.1k	59.5k
.zip	1.1k	33.5k
.jpg	1.2k	82.3k
.mp4	1.2k	25.9k
Encrypted	14.8k	1.5M
Total	29.3k	2.3M

TABLE IV
THE PERFORMANCE OF LIBPROTOIDENT, RANDOMNESS TEST AND TRAFFIC PATTERN BASED CLASSIFIERS TO DETECT ENCRYPTED TRAFFIC

Metric	Randomness Test	LibProtoident	Traffic Pattern
Precision (%)	96.7%	100%	100%
Recall (%)	99.9%	100%	100%
F1 (%)	98.3%	100%	100%

TABLE V
DETECTION FLOW FROM SFTP CLASS

Metric	Randomness Test	LibProtoident	Traffic Pattern
Precision (%)	65.6%	0%	89.9%
Recall (%)	96.7%	0%	90.9%
F1 (%)	78.2%	0%	90.4%

V. MEASUREMENTS

We compare the performance of Randomness Test-based, Traffic Pattern-based, and DPI classifiers to classify our ground truth in the following experiments.

A. Binary Classification

We put all the flows generated from the encrypted protocols in one class and the rest in the other. According to the Table IV, all the classifiers successfully identify encrypted flows. However, it is notable that the randomness test classifier has a slightly lower performance than the rest. This experiment proves that the encryption footprint is identifiable in all flow aspects t at this level of granularity. Therefore, other criteria such as memory consumption and classification time can define the optimum approach. In the next experiments, we continue to compare the classifiers' performance to detect the encryption protocol level.

B. Protocol Classification

Within this experiment, we study the capability of each classifier to seperate the encrypted protocols from others.

Secure-FTP (SFTP) and Secure Copy Protocol (SCP) both use OpenSSH for encryption. It is interesting to study the performance of the classifiers for identifying such traffics. Tables V and VI indicate randomness tests and traffic pattern classifiers despite their different performances, they classify part of traffic successfully. However, DPI fails to classify any traffic from these application protocols. In fact, DPI classifies both classes as SSH.

Following, we change the label of SCP and SFTP flows to SSH and repeat the experiment. Table VII shows that merging these classes under the SSH class not only improves

TABLE VI
DETECTION FLOW FROM SCP CLASS.

Metric	Randomness Test	LibProtoident	Traffic Pattern
Precision (%)	92.4%	0%	90.8%
Recall (%)	50.1%	0%	89.8%
F1 (%)	65%	0%	90.3%

TABLE VII
DETECTING FLOW FROM CLASSES SCP AND SFTP AS SSH

Metric	Randomness Test	LibProtoident	Traffic Pattern
Precision (%)	95.8%	100%	100%
Recall (%)	98.6%	100%	100%
F1 (%)	97.2%	100%	100%

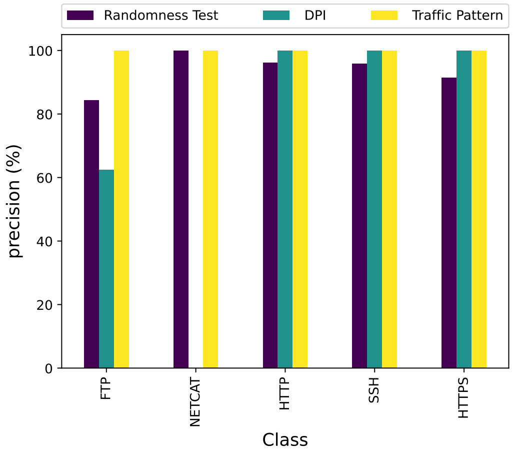

Fig. 4. Precision of considered classifiers to identify the protocol of flows

the performance of DPI significantly but also improves the performance of the other classifiers positively. Since this change does not affect the performance of the classifiers over the rest of the classes, we continue to keep the SSH class for the next experiment.

According to Figure 6, traffic pattern classifies all the flows with the maximum performance. DPI classifies approximately 50% of the NETCAT flows as FTP and the rest as Unknown_TCP. Although DPI detects all FTP traffic (Figure 5) correctly, the false positive generated from NETCAT reduces the precision of the FTP class drastically (Figure 4). The randomness test classifier does not reach the maximum performance. However, it classifies all the classes with an F1 score higher than 86%.

C. Content Classification

Within this experiment, we increase the granularity level of classification to its highest level that is reachable in the scope of our ground truth. We measure the performance of the considered classifiers to separate encrypted contents from plaintext. DPI is not applicable in this level of granularity as it does not have the required signatures. This is the reason why it is excluded from the experiments.

Figures 7 to 9 show that both classifiers can detect encrypted content with high level of performance. However, the F1 (figure 9) of unencrypted classes are mostly bellow 60%. Although the traffic pattern classifier outperforms the randomness test classifier, none of them perform well. Considering that performance of none is 0, each identifies part of the information which is required for the classification. Thus, we expect the combination of extracted information from traffic pattern and randomness tests can improve the overall performance.

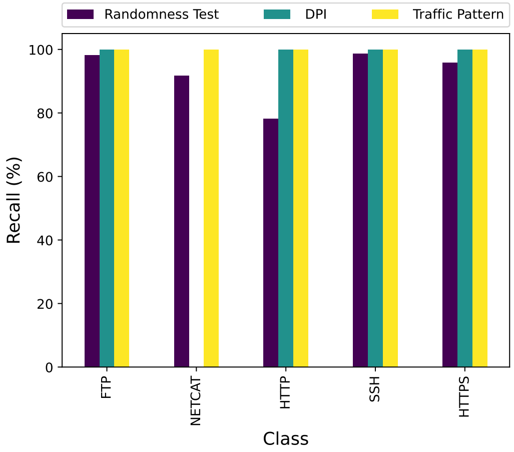

Fig. 5. Recall of considered classifiers to identify the protocol of flows

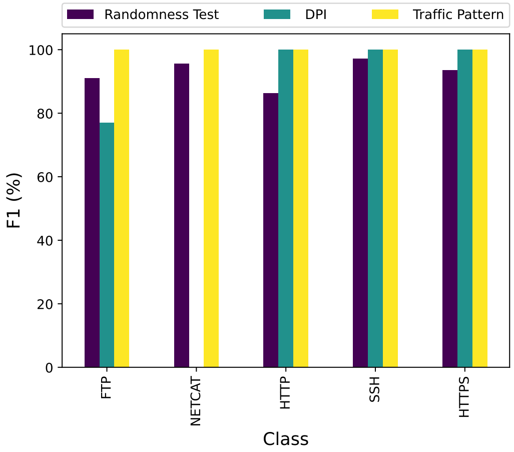

Fig. 6. F1 of considered classifiers to identify the protocol of flows

VI. CONCLUSION

We focused on encryption detection problems in network traffic classification and compared the performance of classifiers based on DPI, traffic pattern, and randomness test. We selected `Libprotoident` and an ANN as the representative instances from DPI and traffic pattern respectively. We also proposed a novel classifier to classify the network traffic according to their payload bitstream randomness characteristics. Moreover, we compare the performance of the selected classifiers to detect encryption among our generated traffic in three different granularity levels. The results showed the classifier which is designed based on traffic pattern detected the encrypted classes in all the three levels with the highest performance metrics. The performance of Deep Packet Inspection (DPI) reduced drastically with increasing the granularity level. Although the randomness test classifier never reached the highest performance, it preserved its performance (F1 > 90%) to detect the encryption traffic at different levels.

As future work, we plan to extend this line of research by comparing the performance of traffic pattern and randomness

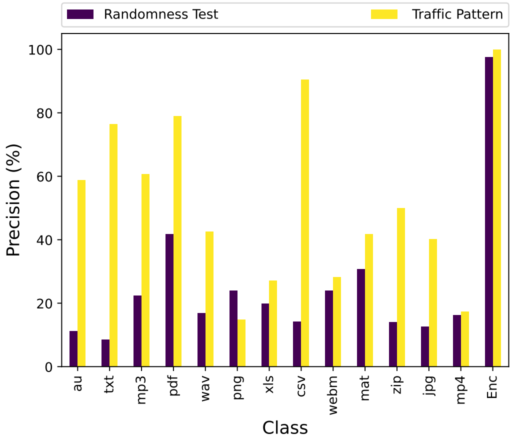

Fig. 7. Precision of considered classifiers to identify the content of flows

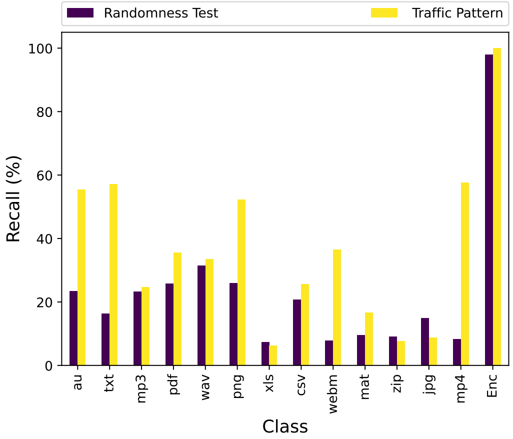

Fig. 8. Recall of considered classifiers to identify the content of flows

test classifiers from resource consumption and time of classification points of views. We will also study the significance of considering both randomness and traffic pattern features to detect encryption on the overall performance of a classifier.

Acknowledgment

This work has been supported in part by the German Research Foundation (DFG) under grant no. DR 639/20-1.

References

[1] S. Lee, K. Levanti, and H. S. Kim, "Network monitoring: Present and future," *Elsevier Computer Networks (COMNET)*, vol. 65, pp. 84–98, Jun. 2014.

[2] G. Carle, F. Dressler, R. A. Kemmerer, H. Koenig, C. Kruegel, and P. Laskov, "Network attack detection and defense - Manifesto of the Dagstuhl Perspective Workshop," *Springer Computer Science - Research and Development (CSRD)*, vol. 23, no. 1, pp. 15–25, Mar. 2009.

[3] G. Nguyen, S. Dlugolinsky, V. Tran, and A. L. Garcia, "Deep Learning for Proactive Network Monitoring and Security Protection," *IEEE Access*, vol. 8, pp. 19 696–19 716, Jan. 2020.

[4] A. Zaalouk, R. Khondoker, R. Marx, and K. Bayarou, "OrchSec: An orchestrator-based architecture for enhancing network-security using Network Monitoring and SDN Control functions," in *IEEE Network Operations and Management Symposium (NOMS 2014)*, Kraków, Poland, May 2014.

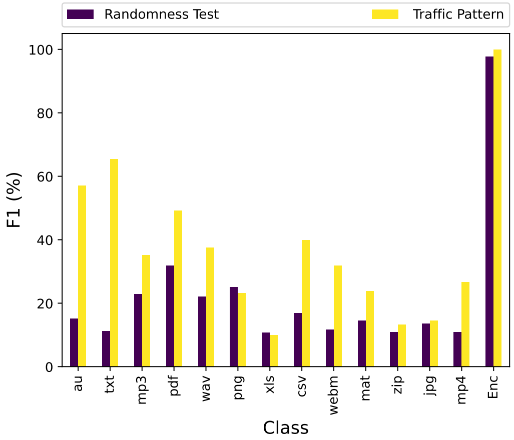

Fig. 9. F1 of considered classifiers to identify the content of flows

[5] G. Cheng and Y. Hu, "Encrypted Traffic Identification Based on N-gram Entropy and Cumulative Sum Test," in *13th International Conference on Future Internet Technologies (CFI 2018)*, Seoul, South Korea, Jun. 2018.

[6] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescape, "DISTILLER: Encrypted traffic classification via multimodal multitask deep learning," *ACM Journal of Network and Computer Applications*, vol. 183-184, Jun. 2021.

[7] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[8] V. Carela-Español, P. Barlet-Ros, M. Solé-Simó, A. Dainotti, W. De Donato, and A. Pescapé, "K-Dimensional Trees for Continuous Traffic Classification," in *Traffic Monitoring and Analysis*, 2010, pp. 141–154.

[9] L. Deri, M. Martinelli, T. Bujlow, and A. Cardigliano, "nDPI: Open-source high-speed deep packet inspection," in *10th International Wireless Communications and Mobile Computing Conference (IWCMC 2014)*, Nicosia, Cyprus, Aug. 2014.

[10] M. S. I. Mamun, A. A. Ghorbani, and N. Stakhanova, "An Entropy Based Encrypted Traffic Classifier," in *Information and Communications Security*, 2016, pp. 282–294.

[11] C. Williamson and A. Madhukar, "A Longitudinal Study of P2P Traffic Classification," in *IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2006)*, Monterey, CA, Sep. 2006.

[12] S. Alcock and R. Nelson, "Libprotoident: traffic classification using lightweight packet inspection," WAND Network Research Group, Technical Report, Jan. 2012.

[13] S. Rezaei and X. Liu, "Deep Learning for Encrypted Traffic Classification: An Overview," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 76–81, May 2019.

[14] D. Ramachandram and G. W. Taylor, "Deep Multimodal Learning: A Survey on Recent Advances and Trends," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 96–108, Nov. 2017.

[15] P. L'Ecuyer and R. Simard, "TestU01: A C library for empirical testing of random number generators," *ACM Transactions on Mathematical Software*, vol. 33, no. 4, pp. 1–40, Aug. 2007.

[16] F. Casino, K.-K. R. Choo, and C. Patsakis, "HEDGE: Efficient Traffic Classification of Encrypted and Compressed Packets," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 11, pp. 2916–2926, Nov. 2019.

[17] R. T. El-Maghraby, N. M. A. Elazim, and A. M. Bahaa-Eldin, "A survey on deep packet inspection," in *12th International Conference on Computer Engineering and Systems (ICCES 2017)*, Cairo, Egypt, Dec. 2017.

[18] J. Sherry, C. Lan, A. P. Raluca, and S. Ratnasamy, "BlindBox: Deep Packet Inspection over Encrypted Traffic," *ACM SIGCOMM Computer Communication Review*, vol. 45, pp. 213–226, Oct. 2015.

[19] B. Xu, G. He, and H. Zhu, "ME-Box: A reliable method to detect malicious encrypted traffic," *Journal of Information Security and Applications*, vol. 59, p. 102823, Jun. 2021.

[20] T. Bujlow, V. Carela-Español, and P. Barlet-Ros, "Independent comparison of popular DPI tools for traffic classification," *Elsevier Computer Networks*, vol. 76, pp. 75–89, Jan. 2015.

[21] H. A. Jamil, A. Mohammed, A. Hamza, S. M. Nor, and M. N. Marsono, "Selection of On-line Features for Peer-to-Peer Network Traffic Classification," in *Advances in Intelligent Systems and Computing*, 2014, vol. 235, pp. 379–390.

[22] W. De Donato, A. Pescape, and A. Dainotti, "Traffic identification engine: an open platform for traffic classification," *IEEE Network*, vol. 28, no. 2, pp. 56–64, Mar. 2014.

[23] H. Doroud, G. Aceto, W. De Donato, E. Alizadeh Jarchlo, A. M. Lopez, C. D. Guerrero, and A. Pescape, "Speeding-Up DPI Traffic Classification with Chaining," in *IEEE Global Communications Conference (GLOBECOM 2018)*, Abu Dhabi, United Arab Emirates, Dec. 2018.

[24] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescape, "Mobile Encrypted Traffic Classification Using Deep Learning: Experimental Evaluation, Lessons Learned, and Challenges," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 445–458, Sep. 2019.

[25] B. Asadi and H. Jiang, "On Approximation Capabilities of ReLU Activation and Softmax Output Layer in Neural Networks," arXiv, cs.LG 2002.04060, Feb. 2020.

[26] H. Doroud, A. Alaswad, and F. Dressler, "Ground truth of encrypted traffic detection," 2022. [Online]. Available: https://zenodo.org/record/6817198

[27] M. Dicks and J. Chavula, "Deep Learning Traffic Classification in Resource-Constrained Community Networks," in *2021 IEEE AFRICON Conference*, Arusha, Tanzania, Sep. 2021.