Received: 19 April 2019 | Revised: 6 September 2019 | Accepted: 1 October 2019

DOI: 10.4218/etrij.2019-0190

FEATURED ARTICLE

ETRI Journal WILEY

Practical evaluation of encrypted traffic classification based on a combined method of entropy estimation and neural networks

Kun Zhou[1,2] | **Wenyong Wang**[1] | **Chenhuang Wu**[1,3] | **Teng Hu**[1,2]

[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Sichuan, China

[2]Institute of Computer Applications, China Academy of Engineering Physics (CAEP), Sichuan, China

[3]Putian University, Fujian, China

Correspondence
Wenyong Wang and Kun Zhou, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Sichuan, China.
Email: wangwy@uestc.edu.cn (W. W.) and zhoukun@std.uestc.edu.cn (K. Z.)

Funding information
This work was partially supported by National Science Foundation of China with the Grant Number: 2018YFB08040505

Encrypted traffic classification plays a vital role in cybersecurity as network traffic encryption becomes prevalent. First, we briefly introduce three traffic encryption mechanisms: IPsec, SSL/TLS, and SRTP. After evaluating the performances of support vector machine, random forest, naïve Bayes, and logistic regression for traffic classification, we propose the combined approach of entropy estimation and artificial neural networks. First, network traffic is classified as encrypted or plaintext with entropy estimation. Encrypted traffic is then further classified using neural networks. We propose using traffic packet's sizes, packet's inter-arrival time, and direction as the neural network's input. Our combined approach was evaluated with the dataset obtained from the Canadian Institute for Cybersecurity. Results show an improved precision (from 1 to 7 percentage points), and some application classification metrics improved nearly by 30 percentage points.

KEYWORDS
deep neural networks, encrypted traffic classification, entropy estimation, PCA

1 | INTRODUCTION

Research on traffic classification has become more challenging than ever, as the innovative applications and mechanisms to conceal the nature of traffic develop and mature rapidly. The accuracy and efficiency of traffic classification methods have attracted great research interest from academia and industry. For example, encrypted Voice over IP network (VoIP) traffic flow needs to be correctly picked out and labeled with appropriate transmission priorities—because of its time-delay sensitive nature—to preserve the quality of service (QoS). Old-fashioned strategies such as packet's header and payload inspection failed to discriminate traffic, because content checking on encrypted traffic was unsuccessful. Some inspiring methods based on the statistical behavior of traffic flow have been proposed. Although these methods achieved high accuracies in differentiating non-encrypted traffic flows, encrypted traffic classification is still in its initial development.

IP packets transmitted in plaintext can be easily recovered by network sniffer tools such as Wireshark. Thus, to ensure communication confidentiality, encryption methods must be employed, and that is becoming a fast-growing trend.

IPsec virtual private network (VPN), transport layer security (TLS)/secure socket layer (SSL), and secure real-time transport protocol (SRTP) dedicated to encrypted VoIP are three major protocols for encrypted network traffic. We selected the IPsec protocol to brief the principles and approaches

for traffic encryption. IPsec can be deployed to enable VoIP communication confidentiality at the IP network level. Some research concluded that a cryptographic engine could bring large overhead for voice traffic and was not perfect for VoIP encryption. This hot topic is worthy of investigation not only for its omnipresence in real-life VoIP scenarios but also for its academic value. Widely used in HTTPS Web traffic, TLS/SSL is undoubtedly one of the most important encryption mechanisms for packet transmission, as the NSS lab predicts that 75% of global web traffic will be encrypted by 2019.

We investigated and evaluated numerous flow features for encrypted traffic classification using four traditional machine learning methods—support vector machines (SVM), random forest (RF), naïve Bayes, and logistic regression—and a neural network (NN). An entropy-based method was used to first distinguish encrypted from non-encrypted traffic. For encrypted traffic, based on the results from the first phase, we designed a NN using three types of packet discriminators—packet length, inter-arrival time (IAT), and direction (forward and backward)—as input-layer parameters. For non-encrypted traffic, we employed principal component analysis (PCA) to cut dimensions by half to achieve high efficiency of classification while maintaining a certain degree of accuracy.

Contributions we made in this study are as follows:

- We investigated three network traffic encryption mechanisms to prepare for classification analysis and evaluation.
- Four traditional machine learning methods and a neural network were evaluated for network traffic classification.
- We proposed a combined approach to distinguish the encrypted traffic from the plaintext traffic using information entropy and a neural network, and we achieved improved results.

The remainder of the article is structured as follows. Section 2 describes related research. In Section 3, we cover our methodology with an emphasis on machine learning techniques, features, and datasets. Section 4 presents evaluation methods and results. Section 5 concludes the article and discusses future work.

2 | RELATED WORK

Some research demonstrated the feasibility of inferring the encrypted packets without decryption from the information leakage or the so-called "side-channel." Attacks by traffic analysis usually materialize at the application or transport/network level for traditional methods. General traffic analysis is based on application/traffic flow-level features, such as correlation statistics between flows at the transceiver ends, which is especially the case for encrypted traffic, because the packet content inspection methods fail before encryption. VoIP service provider, Skype, protects users' privacy by applying mechanisms such as stronger encryption, proprietary protocols, unknown codecs, dynamic path selection, and the constant packet rate. However, researchers have demonstrated how to compromise users' privacy according to a novel traffic analysis attack that extracted application-level features from VoIP call traces. In an interesting study, Wright and others [1] suggested that VoIP packets compressed with variable bit rate (VBR) encoding schemes and encrypted with a fixed-length cipher were susceptible to the attacks of using only the packet length feature to infer the spoken language of the encrypted conversation. Wright and others [2] showed possibilities of spotting the phrases within encrypted VoIP calls under specific circumstances. The article proposed profile hidden Markov models (pHMM) to model the packet sequences for the given phrases. Results indicated an average accuracy of 50% (greater than 90% for some phrases).

Traffic features are critical to the identification and classification of encrypted VoIP traffic for NNs. Anderson and others [3] introduced a complex set of observable data features and showed that these features could be exercised to detect malware communication while preserving the privacy of benign users. Wright and others [4] investigated the extent to which average application protocols could be differentiated with features such as packet size, timing, and direction that remained unchanged after encryption. The accuracy of their traffic classifier was greater than 80% for most protocols. Sherry and others [5] inspected the packet directly in the encrypted traffic. This article elaborated on the approach through an original protocol and encryption scheme.

Sun and others [6] focused on the inference of sensitive information from encrypted network connections with packet sizes and timing. Liberatore and Levine [7] reported their findings of identifying web pages encapsulated in encrypted HTTPs using only the number and size of the encrypted packets. Similarly, Schuster and others [8] proposed a method to spot videos played over an encrypted network channel using the total size of the packets transmitted in a short-time window. Packet inter-arrival times have been studied to infer keystrokes within SSH sessions [9]. Entropy estimation for real-time encrypted traffic identification [10] presented the approach. Results indicated that the encrypted VoIP traffic was detected correctly with a precision greater than 94%. Moore and others [11] made an impressive contribution by proposing more than 200 flow features for further analysis. Velan and others [12] compared the feature-based classification methods and pointed out their weaknesses and strengths. Daniel and others [13] evaluated three machine learning techniques—*k*-means clustering algorithm, MOGA, and C4.5—and concluded that C4.5 was the fastest. Nguyen and others [14] surveyed 18 significant works published from 2004 to 2007 and categorized them according to machine learning algorithms and primary

contributions. An inspiring result by Zhang and others [15] was that traffic classification performance could still be enhanced drastically even with very few training samples. Finamore and others [16] proposed KISS, an Internet classification engine, which achieved very good results with the average true positive rate of 99.6% and 98.1% for the worst case.

Traffic encryption has also gained research interest from the industry. IEEE industry connections established the security subgroup on encrypted traffic inspection (ETI), which aims to standardize an accepted way of traffic inspection, on top of encrypted transport standards. The group also covers requirements for traffic inspection mechanisms based on different use cases and explores proofs of concepts of implementations. Network giant Cisco's latest networking gear, encrypted traffic analytics (ETA) [17] unleashed in 2017, provides advanced traffic analytics including machine learning to identify encrypted network threats.

3 | METHODOLOGY

3.1 | Entropy-based methods

Shannon's entropy theory measures information uncertainty. Given m possible events $A_1, \ldots, A_m$ with probabilities of occurrence $p_1, \ldots, p_m$, entropy H is defined by (1).

$$H = -\sum_{i=1}^{m} p_i \log\left(p_i\right). \tag{1}$$

Equal probabilities have the maximum value of entropy. Unpredictable behavior means uncertainties and, thus, increases entropy. An appropriate estimator for the entropy is needed for small datasets, but entropy estimation based on a small sample becomes harder [18], especially for $N < m$. Motivated by the problems of estimating the entropy with small length N, Olivain and Goubault-Larrecq [19] presented the N-truncated entropy method. The N-truncated entropy $H_N(p)$ is defined as for words w of length N using a maximum likelihood estimator (MLE) to estimate the entropy. Average MLE estimates the number of words to formulate $H_N(p)$. If $p_i = 1/m$ for all i (where p follows the uniform distribution U), then $H_N(U)$ can be computed by (2).

$$H_N(U) = \frac{1}{m_N} \sum_{n_1+\cdots+n_m=N} \left[\binom{N}{n_1+\cdots+n_m} \times \left(\sum_{i=1}^{m} -\frac{n_i}{N} \log \frac{n_i}{N} \right) \right]$$
$$\text{where} \binom{N}{n_1+\cdots+n_m} = \frac{N!}{n_1! \cdots n_m!}. \tag{2}$$

We used MLE as an unbiased estimator of H_N. The estimated value of similarity to $H_N(U)$ reflects how much it resembled the uniform distribution. A Monte Carlo method for estimating $H_N(U)$ with corresponding confidence intervals works for the empirical evaluation test. Paninski [20] concluded that at least $N > \sqrt{m}$ samples were needed for the uniformity test. Inspired by the aforementioned methods, we propose the following information entropy-based algorithm.

Using the traffic sniffer tool's data processing functionality, we selected randomly 64 bytes in TCP protocol contents (TLSv1.2 for encrypted traffic) from the experimental pcap files (ISCX-VPN-NonVPN-2016) to compute the entropy. We used Monte Carlo pseudorandom sequence to mimic the encrypted traffic and compared it with the experiment data. See Algorithm 1 in Table 1 for entropy estimation details.

3.2 | Artificial neural networks

The research demonstrated the efficacy of artificial neural networks, especially the deep learning technologies [21], in traffic classification. Computational power, such as graphical processing units (GPUs) and Google's tensor processing unit (TPU), grows exponentially, and a deep NNs design and training have become more feasible. There are reports that NN architectures such as multilayer perceptrons (MLP), convolutional neural networks (CNN), recurrent neural networks (RNN), autoencoders, and generative adversarial networks (GAN) have all been used for traffic classification. In this article, we used a traditional NN with three layers (input, hidden, and output layer) to evaluate our combined approach.

$$\begin{aligned} h^{(1)} &= \sigma(W^{(1)}x), \\ h^{(2)} &= \mathrm{ReLu}(W^{(2)}x), \\ h^{(3)} &= \sigma(W^{(3)}h^{(1)}), \\ y &= \sigma(W^{(3)}h^{(1)} + W^{(5)}h^{(2)}. \end{aligned} \tag{3}$$

TABLE 1 Algorithm using entropy-based methods

Algorithm 1 Distinguish encrypted traffic from plaintext traffic using entropy-based methods
1. Generate Monte Carlo pseudorandom sequence between 0 and 256 with the length of 64 bytes for 10 000 times; for example, (*a*3 *b*4 *f*1 23 48 90 *ab* 34…), where "*a*3" stands for one character.
2. Using (2) to calculate $H_N(U)$, where $N = 64$, $m = 256$, and N_i stands for the frequency of character i, and $H_N(U)$ represents the average information entropy using the MLE method.
3. Collect and save the experimental pcap dataset using the sniffer tool's functionality.
4. Select randomly 64 bytes protocol contents in 100 TCP flow data and store them into 100*64 dataset matrix (test dataset matrix);
5. For every row of the test dataset matrix, calculate information entropy for every character using (1), that is $H_k(P)$.
6. Calculate average variance, $\sigma = \sum_{i=1}^{k} (H_i(p) - H_u(p))^2/(k-1)$, determine encryption = true if actual information entropy value falls within three times confidence range, otherwise encryption = false.

The NN architecture is shown in Figure 1.

The NN uses modular backpropagation algorithms, linear module plus softmax (activation function), and cross-entropy (loss function). The basic principle for each module was to (i) calculate "forward_pass," "backward_pass," and "para_gradients," (ii) chain together "forward_pass" and "backward_pass," (iii) compute "para_gradients," and (iv) apply gradients and iterate. For the NN, online stochastic gradient descent (SGD) was used to construct our model. See Table 2 for the main modular processes.

We proposed a combined approach to distinguish the encrypted and plaintext traffic and then to further classify the encrypted traffic into eight types of applications. The NN is comprised of three layers: the input layer has 23 neurons for 23 traffic features, the hidden layer has 100 neurons, and the output layer has eight neurons for eight classes (which are VoIP, audio streaming, browsing, chat, email, file transfer, P2P, and video streaming). Figure 2 depicts the high-level workflow of our approach.

3.3 | Dataset

The dataset from the Canada Institute for Cybersecurity is widely used by researchers worldwide. It includes network traffic such as IDS, Tor/Non-Tor, VPN/Non-VPN, and an Android malware dataset. For this study, we selected two relevant traffic datasets, "ISCX-Tor/Non-Tor" and "ISCX VPN/Non-VPN." These two datasets store captured traffic packets in the pcap format files, which we can open with Wireshark to inspect the packet details. The CSV files are traffic flow statistics typically used for supervised learning with assigned training labels. For "ISCX VPN-NonVPN" datasets, packets are captured over virtual private network (VPN) sessions, which are generally considered to be encrypted. The "ISCX-Tor/Non-Tor" dataset is generated by the Tor browser. The basic principle for Tor is to build encrypted connections in a way that no individuals ever know the complete path. Complex port obfuscation algorithms also improve the privacy and anonymity of Tor traffic. As the complete datasets were very huge and diversified, we chose the aforementioned two datasets for our experiments. VPN-NonVPN datasets were used to classify the encrypted and non-encrypted traffic, while Tor/Non-Tor was for application classification. We separated the training and test sets (no validation set) using random sampling. About 2000 packets randomly selected from the original dataset were used to feed to the NN. The repeat times for the train/test were set to 10. The training set size was 90% and 10% was for the test set.

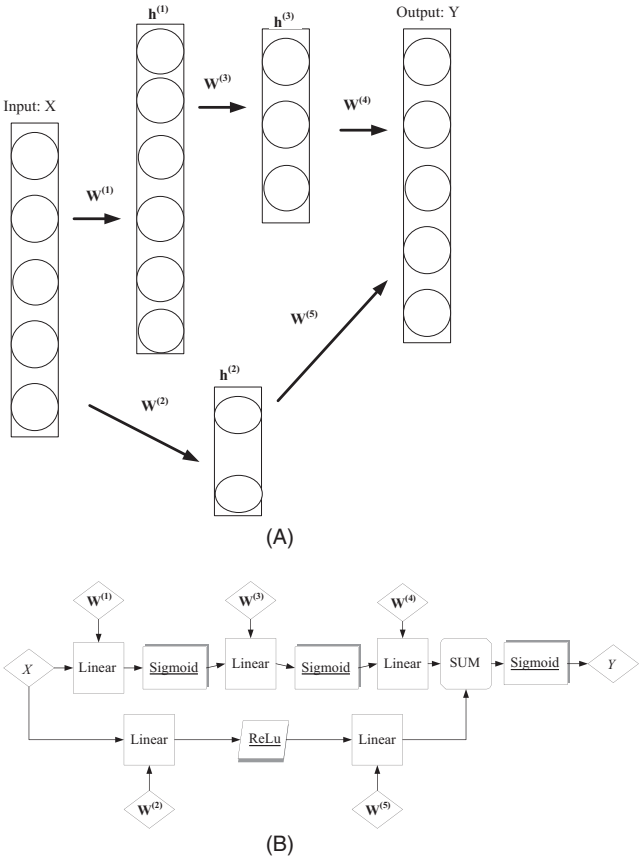


FIGURE 1 (A) Traditional neural network representation and (B) neural network computational graph

3.4 | Methods for encrypted and non-encrypted traffic classification

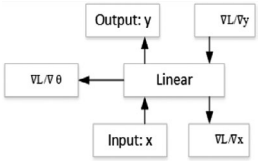
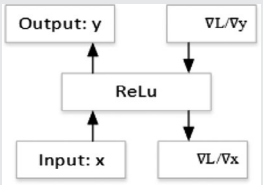
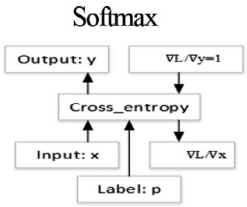
We investigated correlations between the 23 different statistic features and the target class. The following figures show the density distribution of the features "duration" and "total_fiat" with class type (VPN/Non-VPN), respectively. There were two spikes for "duration" of Non-VPN and one spike for "total_fiat" of Non-VPN. The differences between the two types of distributions were obvious, which indicated these two features could be used for classification. Figure 3 demonstrates the distribution of the two selected features.

We selected VPN/Non-VPN dataset and studied the distribution of every feature and class type. The heat map for the dataset and box plot for the two chosen statistic features (duration, total_fiat) are illustrated in Figure 4.

Heat map represents data in the matrix with changing colors. Darker colors mean the larger data. Some features showed they might be useful for classification. Box plot depicts the numeric data value of "duration" and "total_fiat" through quartiles.

We ranked the top 10 among 23 features using three metrics: information gain (IG), Gini decrease, and χ^2. IG for a specified attribute is defined using Shannon entropy $H(T)$. $\mathrm{IG}(T, a) = H(T) - H(T \mid a)$ is the difference between the a priori Shannon entropy $H(T)$ of the training set and the conditional entropy $H(T \mid a)$. The mean decrease in Gini coefficient

TABLE 2 Neural network modular and processes

Modular	Processes
	forward_pass: $Y = Wx + b$; backward_pass: $\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} W$ para_gradients: $\frac{\partial L}{\partial w} = \left(\frac{\partial L}{\partial y}\right)^{\mathrm{T}} x^{\mathrm{T}}$, $\frac{\partial L}{\partial b} = \frac{\partial L}{\partial y}$
	forward_pass: $Y = \mathrm{ReLu}(x)$; backward_pass: $\frac{\partial L}{\partial x_i} = (y_i > 0)$
	$Y = \mathrm{Softmax}(x)$; $y_n = \frac{e^{x_n}}{\sum_m e^{x_m}}$ forward_pass: $Y = L$ $= \mathrm{Cross\_entropy}(p, x)$; $Y = -\sum_i p_i \log x_i$ backward_pass: $\frac{\partial L}{\partial x_i} = \frac{p_i}{\partial x_i}$

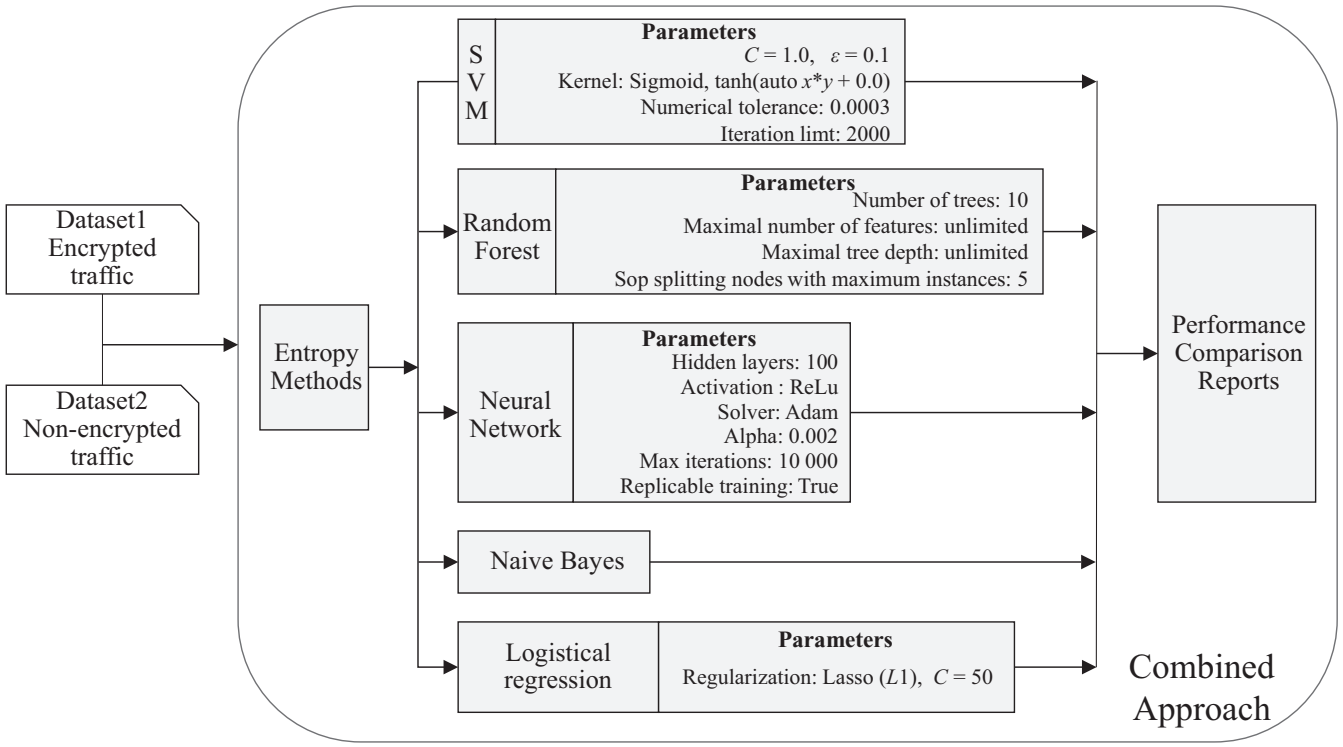


FIGURE 2 Workflow of the combined approach

measures each variable's contributions to the homogeneity of the nodes from scale 0 (homogeneous) to 1 (heterogeneous). Chi-squared test calculates the difference between the distribution of plaintext and decrypted ciphertext. A lower value of the test means a higher probability of successful decryption. Figure 5 illustrates the results of the top 10 rankings.

3.5 | Methods for application classification

Dataset ISCX-Tor-Non-tor2017 was used for the classification of different types of applications. We use eight applications and 23 features (see Table A1 in Appendix for definitions). The distribution of application type with frequency and probabilities is

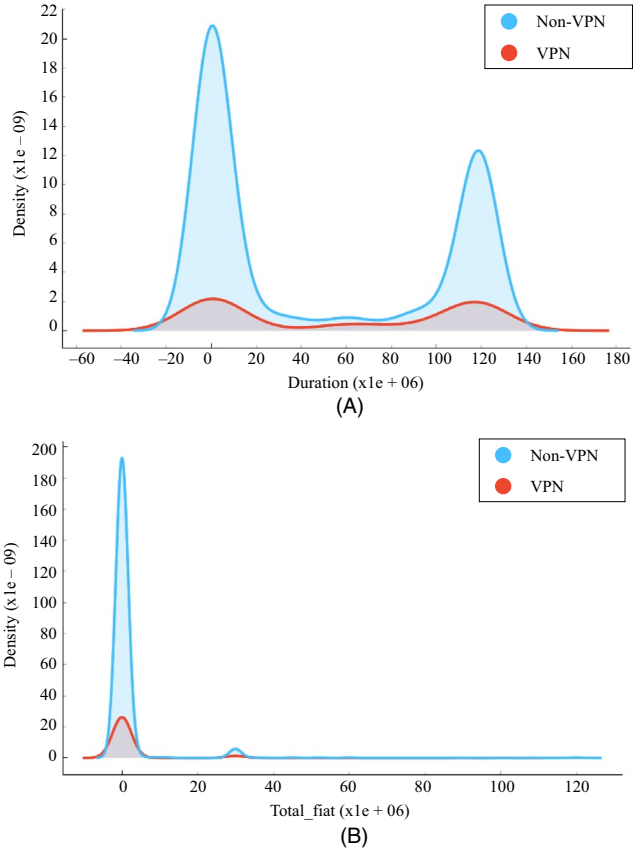


FIGURE 3 (A) Distribution for feature "duration" and Class and (B) distribution for feature "total_fiat" and Class

shown as a bar chart and freeviz in Figure 6. Freeviz shows that points in the same class are attracted to each other, whereas points in different classes repel each other.

	#	Info. gain	Gini	χ^2
mean_fiat		0.104	0.041	50.637
std_flowiat		0.087	0.034	62.462
max_flowiat		0.083	0.028	44.879
flowBytesPerSecond		0.079	0.028	42.226
min_fiat		0.077	0.029	41.466
mean_biat		0.070	0.029	69.111
min_biat		0.068	0.025	58.108
min_idle		0.058	0.025	152.566
mean_active		0.057	0.025	160.418
std_active		0.057	0.027	189.316

FIGURE 5 Input data features ranking by relevance: top 10

We first employed principal component analysis (PCA) to preprocess this type of dataset. This method efficiently reduces the dimensions while retaining the variance of the data and, thus, reduces the sizes of input layer. The dataset contains 23 features. After PCA processing, 10 principal components were computed at the variance greater than 98%. We selected the top 10 principal components to preserve the data variance at 0.984 (Figure 7).

After analyzing 23 features using PCA, we found that half of the features could be dropped to improve efficiency while preserving accuracy at a variance of 98.4%. Figure 8 demonstrates the top 10 principal components.

4 | RESULTS AND DISCUSSION

Numerous machine learning approaches have been proposed for traffic classification. Gil and others [22] adopted features such as the flow's duration, bps, and inter-arrival time for both

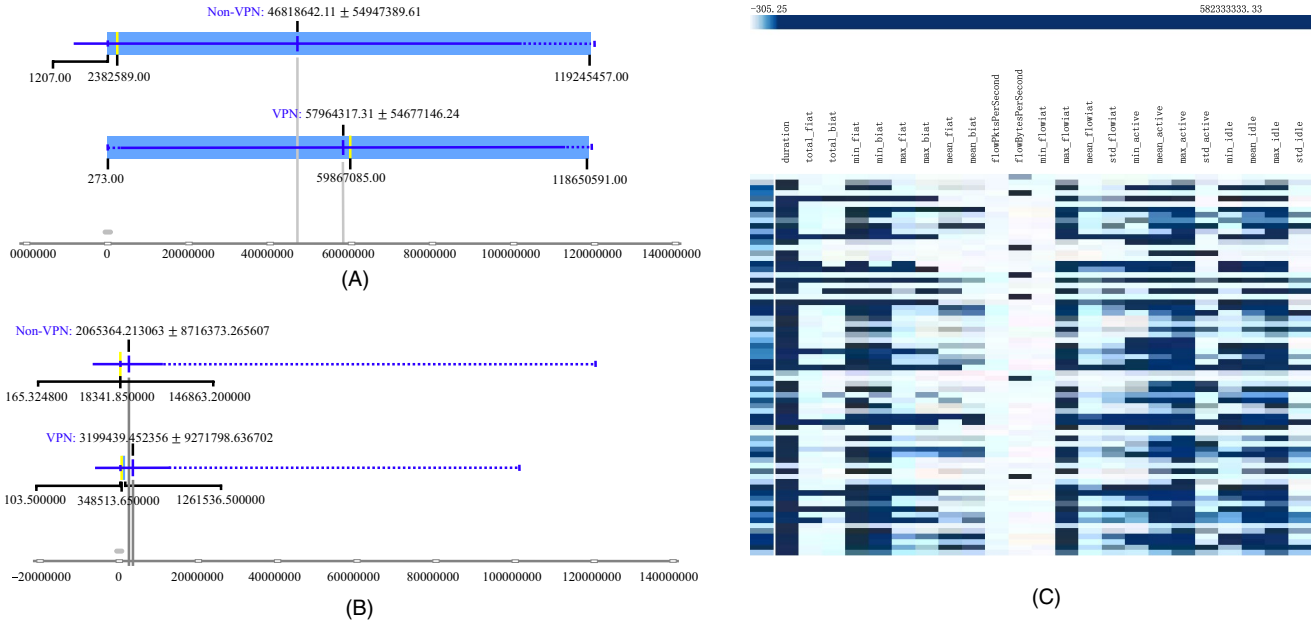


FIGURE 4 (A) Feature "duration" at student' t: 3.453 ($p = 0.001$), (B) feature "max_fiat" at student' t: 2.095 ($p = 0.037$), and (C) heat map for 23 data features

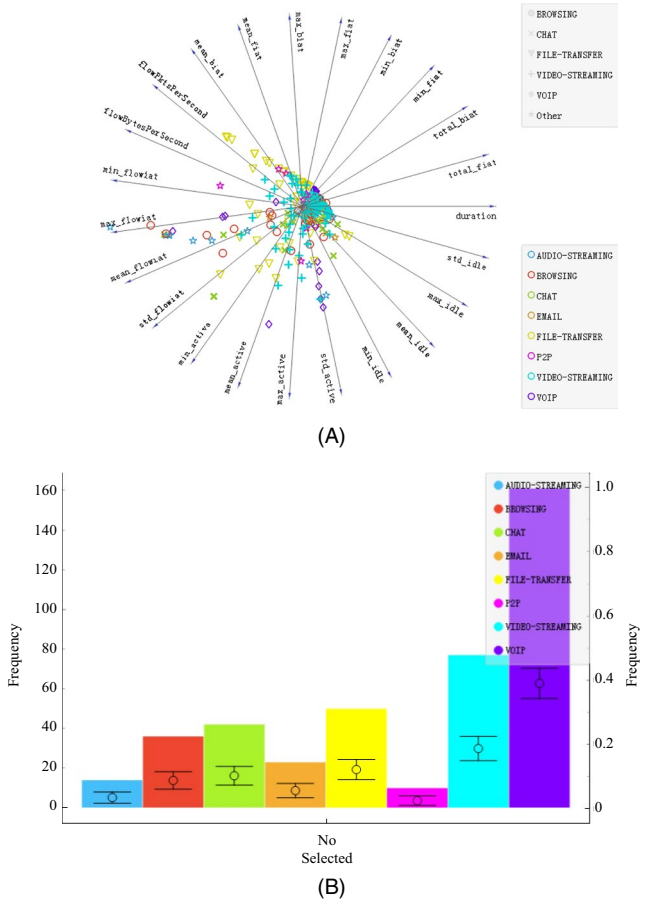


FIGURE 6 (A) Freeviz for different application distribution and (B) application distribution with frequency (left) and probabilities (right)

directions to characterize the network traffic. They experimented using *k*-nearest neighbor (*k*NN) and C4.5 decision tree algorithms and achieved approximately 92% and 88% recall, respectively, for the VPN-tunneled dataset. Yamansavascilar

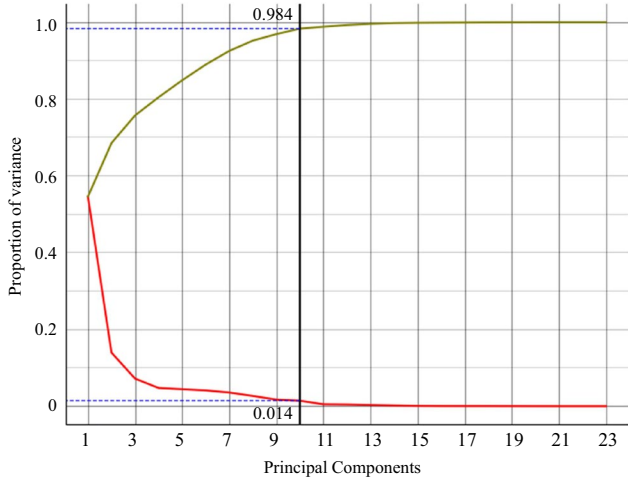


FIGURE 7 PCA analysis of 23 data features

class1	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
Non-VPN	15.1655	-14.5329	-8.31704	0.609728	2.11287	-2.2208	-2.82843	0.496161	1.39983	1.12375
Non-VPN	15.1655	-14.5329	-8.31703	0.609725	2.11345	-2.22066	-2.82828	0.496124	1.39976	1.12374
Non-VPN	11.848	-10.2438	-4.28449	0.187281	1.23975	-1.61743	-1.79819	-1.64557	1.31875	-0.263257
Non-VPN	10.3339	-9.56557	-4.63601	0.174459	1.05608	-1.15793	-1.16409	-0.02440...	0.986922	-1.51775
Non-VPN	9.67941	-9.49506	-5.23849	0.266708	1.15237	-1.1737	-1.3195	0.650077	0.79354	-1.12368
Non-VPN	9.64135	-9.48303	-5.26718	0.268242	1.15098	-1.16269	-1.3059	0.71072	0.782624	-1.15621
Non-VPN	11.7946	-8.96228	-3.45037	0.28945	1.26572	-1.53345	-1.99065	-0.981271	1.20711	1.36534
Non-VPN	6.64086	-7.33362	-5.04293	0.38333	1.04859	-0.830609	-1.22246	2.28705	0.182207	-0.74712
Non-VPN	6.35357	-7.09904	-5.15542	0.421848	0.996261	-0.688299	-1.09219	2.64195	0.06106...	-1.01652
VPN	6.66159	-6.81966	-3.81116	0.213478	0.900341	-0.910813	-1.15717	0.826318	0.509437	-0.561344

FIGURE 8 Top 10 features using the PCA method

and others [23] selected 111 discriminators for 14 classes of applications and achieved an accuracy of 94% with *k*-NN algorithm. However, in their report, they did not mention the specific details of their implementation, and the results need to be revalidated by independent third parties to increase their credibility.

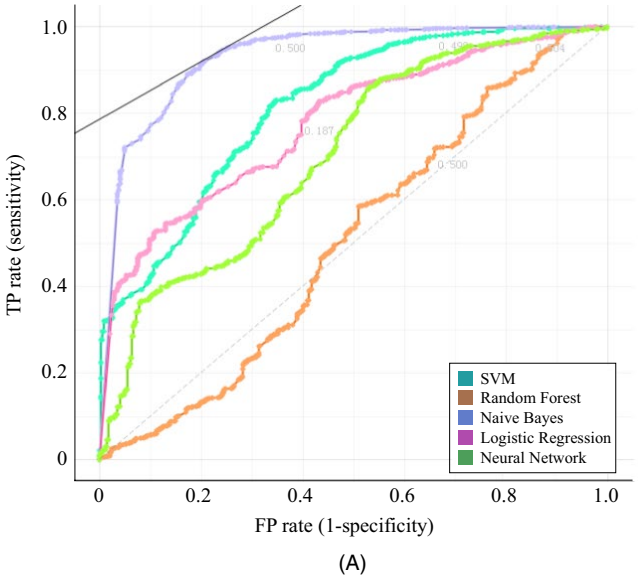
SVM [24] is used for data classification and regression. RF [25] mainly constructs decision trees to achieve the same functions as SVM does. Naïve Bayes model [26] is derived from applying Bayes' theorem with strong independence assumptions between the features. Logistic regression models the probability of output in terms of input and can be used for classification. Aceto and others [27,28] proposed many novel approaches to classify mobile applications. In the study by G. Aceto et al and others [29], their results showed that three anonymity networks (Tor, I2P, and JonDonym) can be easily distinguished with an accuracy of 99.99%. Taylor [30] studied smartphone apps from the encrypted traffic and achieved more than 99% accuracy. Rezaei and others [31] presented commonly used deep learning methods and their applications in traffic classification tasks. Lotfollahi and others [32] proposed a "deep packet" scheme to identify encrypted traffic and distinguish between VPN and non-VPN network traffic. Aceto and others [33] presented an inspiring contribution to encrypted TC: they used several state-of-the-art deep learning techniques to set a framework for comparisons and a performance evaluation workbench. They concluded that although DL had open issues and pitfalls, it was viable for the traffic classification.

Our method of entropy estimation for encrypted or non-encrypted traffic achieved an average accuracy of 98%. Dataset VPN stands for encrypted and non-VPN for plaintext (See Table 3).

The test results showed that the RF outperformed the NN by a small margin. We consider that this is because the RF parameters (such as "fixed random seeds" and hyperparameters) happened to be well set. NN has also many important parameters

TABLE 3 Metrics for entropy methods

	Non-VPN	VPN	∑
Non-VPN	97.7%	1.0%	1651
VPN	2.3%	99%	348
∑	1686	313	1999



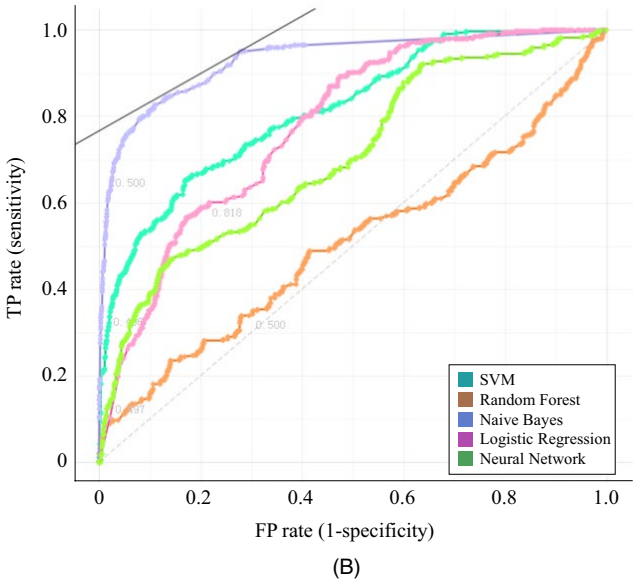


FIGURE 9 (A) FP rate for Non-VPN class using traditional methods. (B) FP rate for VPN class using traditional methods

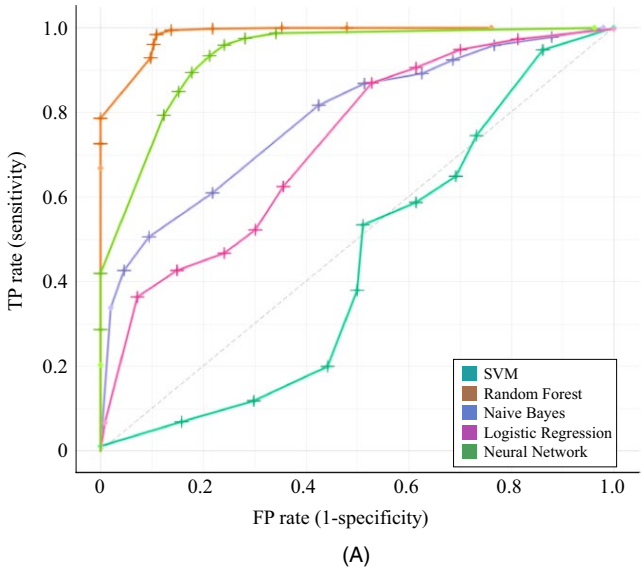
TABLE 4 Metrics using the traditional methods

Method	AUC	F1	Precision	Recall
SVM	0.500	0.274	0.657	0.270
Random forest	0.933	0.915	0.916	0.919
Neural network	0.819	0.846	0.858	0.868
Naïve Bayes	0.779	0.768	0.809	0.746
Logistic regression	0.717	0.787	0.792	0.830

to set, and we simply used the combination of ReLU activation function, Adam solver, regularization alpha = 0.002, and maximum 10 000 iterations, and the results were close to RF. Because we did not exhaust the combinations of the parameters of NN, we could not conclude that RF outperformed the NN methods, and it was highly possible that fine-tuning of the parameters of NN could produce better results.

4.1 | Encrypted traffic classification

We compared traditional machine learning methods with our combined approach for classification using the same dataset, NN, and parameters. The following two ROC curves (Figure 9) representing the FP rate for "VPN" and "Non-VPN," respectively, demonstrated the traditional machine learning methods used to distinguish encrypted from plaintext traffic. These traffic traces were directly classified without using the entropy method.



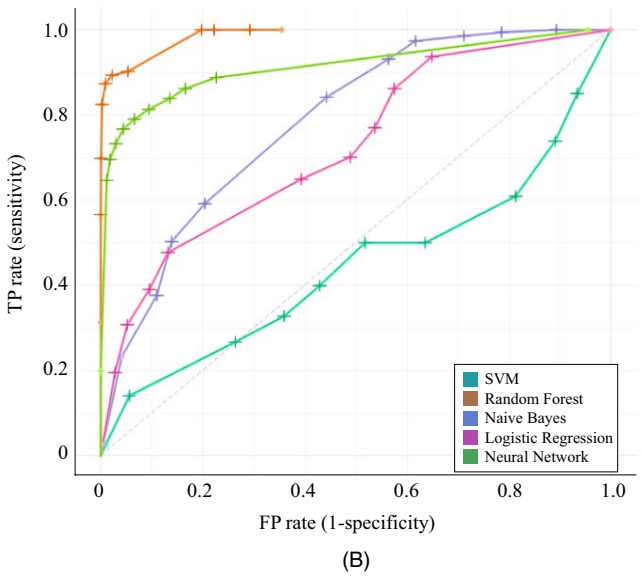


FIGURE 10 (A) FP rate for Non-VPN class using our combined approach. (B) FP rate for VPN class using our combined approach

TABLE 5 Metrics of our proposed methods

Method	AUC	F1	Precision	Recall
SVM	0.443	0.564	0.714	0.505
Random forest	0.986	0.971	0.971	0.971
Neural network	0.947	0.925	0.929	0.930
Naïve Bayes	0.784	0.769	0.811	0.751
Logistic regression	0.732	0.794	0.807	0.836

Metrics in Table 4 summarize the two kinds of class type results using five different methods with four criteria using the traditional approach.

Note that recall (Rc), precision (Pr), and F1 score (F1) are important metrics for classification performance. These metrics are calculated as (4).

$$\begin{aligned} \mathrm{F1} &= \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}, \\ \mathrm{Precision} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \\ \mathrm{Recall} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}. \end{aligned} \tag{4}$$

AUC represents the area the ROC curve occupies. The *x*-axis stands for the cumulative distribution of the false-alarm probability (FP rate) and the *y*-axis for the cumulative distribution of the detection probability (TP rate). Using our combined approach, we obtained the following results of the ROC curve for the two class types ("VPN" and "Non-VPN"). The same traffic traces were tested with our combined approach. They were first distinguished as encrypted or non-encrypted and then further classified if encrypted. Two kinds of class type results using our combined approach are summarized in Figure 10 and Table 5.

To compare the traditional and our combined approach, the same parameters as in "cross-validation" were used. We concluded that all five methods' performances have been improved, from 1 percentage point for the naïve Bayes method to 7 percentage points for the NN.

4.2 | Application of traffic classification

As stated in the methodology section, we used Tor/non-Tor dataset as plaintext traffic classified by the first stage of the entropy method to classify applications. Results in Table 6 were obtained by using traditional machine learning methods, and the confusion matrix (Figure 11) was specifically for NNs.

Metrics in Table 7 were for our combined approach, and the confusion matrix (Figure 12) was also for the same NN.

TABLE 6 Metrics for different classes of applications using traditional methods

TP Rate	FP Rate	Precision	Recall	F1	ROC Area	Class
0.963	0.040	0.939	0.963	0.951	0.968	VoIP
0.357	0.020	0.385	0.357	0.370	0.672	Audio
0.306	0.050	0.367	0.306	0.333	0.661	Browsing
0.571	0.065	0.500	0.571	0.533	0.803	Chat
0.478	0.023	0.550	0.478	0.512	0.735	Email
0.740	0.041	0.712	0.740	0.725	0.870	FTP
0.500	0.007	0.625	0.500	0.556	0.742	P2P
0.636	0.083	0.636	0.636	0.636	0.813	Video
0.719	0.049	0.713	0.719	0.715	0.855	Weighted Avg.

FIGURE 11 Matrix for different application types using traditional methods

	AUDIO-STREAMING	BROWSING	CHAT	EMAIL	FILE-TRANSFER	P2P	VIDEO-STREAMING	VOIP	Σ
AUDIO-STREAMING	63.6%	0.0%	8.3%	0.0%	2.3%	0.0%	2.3%	0.0%	14
BROWSING	18.2%	33.3%	18.8%	4.2%	2.3%	12.5%	11.6%	0.5%	36
CHAT	9.1%	30.3%	45.8%	0.0%	4.7%	0.0%	5.8%	1.2%	42
EMAIL	0.0%	6.1%	0.0%	66.7%	2.3%	0.0%	3.5%	0.5%	23
FILE-TRANSFER	9.1%	6.1%	6.2%	12.5%	79.1%	0.0%	7.0%	0.5%	50
P2P	0.0%	3.0%	0.0%	0.0%	0.0%	25.0%	8.1%	0.0%	10
VIDEO-STREAMING	0.0%	12.1%	14.5%	16.7%	7.0%	62.5%	61.6%	0.5%	77
VOIP	0.0%	9.1%	6.2%	0.0%	2.3%	0.0%	0.0%	96.2%	161
Σ	11	33	48	24	43	8	86	160	413

TABLE 7 Metrics for different classes of applications using our combined methods

TP Rate	FP Rate	Precision	Recall	F1	ROC Area	Class
0.994	0.004	0.994	0.994	0.994	1.000	VoIP
0.857	0.008	0.800	0.857	0.828	0.997	Audio
0.889	0.013	0.865	0.889	0.877	0.993	Browsing
0.976	0.011	0.911	0.976	0.943	0.997	Chat
0.826	0.000	1.000	0.826	0.905	0.996	Email
0.940	0.003	0.979	0.940	0.959	0.999	File-Transfer
0.800	0.000	1.000	0.800	0.889	0.998	P2P
0.974	0.015	0.938	0.974	0.955	0.995	Video
0.954	0.007	0.956	0.954	0.954	0.998	Weighted Avg.

	AUDIO-STREAMING	BROWSING	CHAT	EMAIL	FILE-TRANSFER	P2P	VIDEO-STREAMING	VOIP	Σ
AUDIO-STREAMING	93.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	14
BROWSING	0.0%	92.1%	2.3%	0.0%	0.0%	0.0%	0.0%	0.0%	36
CHAT	0.0%	0.0%	95.5%	0.0%	0.0%	0.0%	0.0%	0.0%	42
EMAIL	0.0%	0.0%	0.0%	100%	0.0%	0.0%	1.3%	0.0%	23
FILE-TRANSFER	0.0%	2.6%	0.0%	0.0%	94.1%	0.0%	1.3%	0.0%	50
P2P	6.7%	0.0%	0.0%	0.0%	0.0%	100%	0.0%	0.0%	10
VIDEO-STREAMING	0.0%	2.6%	2.3%	0.0%	3.9%	0.0%	96.1%	0.0%	77
VOIP	0.0%	2.6%	0.0%	0.0%	2.0%	0.0%	1.3%	100%	161
Σ	15	38	44	22	51	9	76	158	413

FIGURE 12 Matrix for different application types using our combined approach

The criteria of the TP rate, FP rate, precision, and recall for classification have been greatly improved, with some metrics nearly 30 percentage points up, which proved the effectiveness of our approach in traffic applications.

5 | CONCLUSIONS

We analyzed the computational complexities for the combined approach. For training a NN, we analyzed the time complexity that has three layers with i, j, and k nodes, respectively, with n training examples and m epochs. The result was $O(mn \times (ij + jk))$. For the SVM problem, the computational complexity was on the order of n^3 (n is the size of the training dataset). RF is an ensemble model of decision trees. The time complexity for building one decision tree is $O(v \times n \log(n))$, where n is the number of records and v is the number of variables. Therefore, for a RF with ntree number of trees, the complexity would be $O(\text{ntree} \times v \times n \log(n))$. For naïve Bayes, it is $O(N \times d)$, where N is the number of training examples, and d stands for the dimensionality of the features. For logistic regression, computational complexity with gradient-based optimization is $O(f \times c \times s \times e)$, with f features, c classes, s samples, and e epochs. Complexity for entropy estimation is at the same scale as naïve Bayes. The overall complexity of the combined approach is the maximum of all the aforementioned methods.

Finally, we concluded that our combined approach outperformed all the other naïve machine learning methods on the "ISCX VPN-NonVPN/ISCX-Tor-NonTor-2017" traffic dataset in the traffic classification. We envisage that our work can be viewed as a fusion of machine learning, especially deep learning, with traffic classification issues. Although deep learning methods for traffic classification were proposed and reported with high accuracy, open issues still exist: for example, features or models may be non-representative, or the approach may only work for a particular dataset. Moreover, our approach has only been studied on certain types of traffic. A comprehensive study of new encryption protocols, such as TLS 1.3, has not been conducted yet.

ACKNOWLEDGMENTS

The author would like to thank all the anonymous reviewers for valuable suggestions and the Canadian Institute for Cybersecurity for the datasets.

CONFLICT OF INTEREST

The author and co-authors have no conflict of interest to declare.

ORCID

Kun Zhou https://orcid.org/0000-0002-7926-6017
Chenhuang Wu https://orcid.org/0000-0001-8002-7630
Teng Hu https://orcid.org/0000-0002-8624-0210

REFERENCES

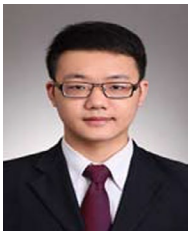
1. C. V. Wright et al., *Language identification of encrypted VoIP traffic: Alejandra y Roberto or Alice and Bob?*, in Proc. USENIX Security Symp. USENIX Security Symp., Moston, MA, USA, Aug. 2007, Article no. 4.
2. C. V. Wright et al., *Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations*, in IEEE Symp. Security Privacy, Oakland, CA, USA, May 2008, pp. 35–49, https://doi.org/10.1109/SP.2008.21.
3. B. Anderson, S. Paul, and D. McGrew, *Deciphering malware's use of TLS without decryption*, arXiv, 2016. https://arxiv.org/abs/1607.01639.
4. C. V. Wright, F. Monrose, and G. M. Masson, *On inferring application protocol behaviors in encrypted network traffic*, J. Mach. Learning Research **7** (2006), 2745–2769.
5. J. Sherry et al., *BlindBox: Deep Packet Inspection over Encrypted Traffic*, in Proc. ACM Conf. Special Interest Group Data Commun., London, UK, Aug. 2015, pp. 213–226.
6. Q. Sun et al., *Statistical identification of encrypted web browsing traffic*, in Proc.IEEE Symp. Security Privacy Berkeley, CA, USA, May 2002, pp. 1–12.
7. M. Liberatore and B. N. Levine, *Inferring the source of encrypted HTTP connections*, in Porc. ACM Confe. Comput. Commun. Security, Alexandria, VA, USA, 2006, pp. 255–263.
8. R. Schuster, V. Shmatikov, and E. Tromer, *Beauty and the burst: Remote identification of encrypted video streams*, in Proc.USENIX Security Symp., Vancuver, Canada, 2017, pp. 1357–1374.
9. D. X. Son, D. Wagne, and X. Tian, *Timing analysis of keystrokes and timing attacks on SSH*, in Proc. USENIX Security Symp., Washington, DC, USA, Aug. 2001, Article no. 25.
10. P. Dorfinger, G. Panholzer, and W. John, *Entropy estimation for real-time encrypted traffic identification*, in Proc. Int. Workshop Traffic Monitoring Analysis, Vienna, Austria, 2011, pp. 164–171. https://doi.org/10.1007/978-3-642-20305-3_14.
11. A. Moore, D. Zuev, and M. Crogan, *Discriminators for use in flow-based classification*, Department of Computer Science Research Reports; RR-05-13, 2005.
12. P. Velan et al., *A survey of methods for encrypted traffic classification and analysis*, Int. J. Netw. Manag. **25** (2015), 1–24.
13. D. J. Arndt and A. N. Zincir-Heywood, *A comparison of three machine learning techniques for encrypted network traffic analysis*, in IEEE Symp. Computat. Intell. Security Defense Applicat. Paris, France, Apr. 2011, https://doi.org/10.1109/CISDA.2011.5945941.
14. T. T. Nguyen and G. Armitage, *A survey of techniques for Internet traffic classification using machine learning*, IEEE Commun. Surveys Tutor. **10** (2008), 56–76. https://doi.org/10.1109/SURV.2008.080406.
15. J. Zhang et al., *Network traffic classification using correlation information*, IEEE Trans. Parallel Distrib. Syst. **24** (2012), 104–117. https://doi.org/10.1109/TPDS.2012.98.
16. A. Finamore et al., *KISS: Stochastic packet inspection classifier for UDP traffic*, IEEE/ACM Trans. Netw. **18** (2010), 1505–1515. https://doi.org/10.1109/TNET.2010.2044046.
17. B. Anderson, S. Paul, and D. McGrew, *Deciphering malware's use of TLS*, [without decryption] J. Comput. Virology Hacking Techn. **14** (2018), 195–211.
18. J. A. Bonachela, H. Hinrichsen, and M. A. Munoz, *Entropy estimates of small data sets*, J. Phys. A: Math. Theor. **41** (2008), 1–9.
19. J. Goubault-Larrecq and J. Olivain, *Detecting Subverted Cryptographic Protocols by Entropy Checking*, Research Report LSV-06-13, 2006, INRIA Futurs projet SECSI.
20. L. Paninski, *Estimation of entropy and mutual information*, Neural Computation, Neural Comput. **15** (2003), 1191–1253.
21. M. Lotfollahi et al., *Deep Packet: A novel approach for encrypted traffic classification using deep learning*, Soft Comput. (2019), 1–14, https://doi.org/10.1007/s00500-019-04030-2.
22. G. D. Gil et al., *Characterization of encrypted and VPN traffic using time-related features*, in Proc. Int. Conf. Inform. Syst. Security Privacy, 2016 pp. 407–414, https://doi.org/10.5220/0005740704070414.
23. B. Yamansavascilar et al., *Application identification via network traffic classification*, in Proc. Int. Conf. Comput., Netw. Commun., Santa Clara, CA, USA, Jan. 2017, https://doi.org/10.1109/ICCNC.2017.7876241.
24. B.-H. Asa et al., *Support vector clustering*, J. Mach. Learn. Research **2** (2001), 125–137.
25. T. K. Ho et al., *The random subspace method for constructing decision forests*, IEEE Trans. Pattern Anal. Mach. Intell. **20** (1998), 832–844.
26. I. Rish, *An empirical study of the naive Bayes classifier*, IJCAI Workshop Empirical Methods AI **3** (2001), 41–46.
27. G. Aceto et al., *Multi-classification approaches for classifying mobile app traffic*, J. Netw. Comput. Applicat. **103** (2018), 131–145.
28. G. Aceto et al., *Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges*, IEEE Trans. Netw. Service Manag. **16** (2019), 445–458.
29. G. Aceto et al., *Anonymity services Tor, I2P, JonDonym: Classifying in the Dark (Web)*, IEEE Trans. Dependable Secure Comput. (2018), Early Access.
30. V. F. Taylor et al., *Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic*, in Proc. IEEE Eur. Symp. Security Privacy (EuroS&P), Saarbrucken, Germany, Mar. 2016, pp. 439–454.
31. S. Rezaei and X. Liu, *Deep learning for encrypted traffic classification: An overview*, IEEE Commun. mag. **57**(2019), 76–81.
32. M. Lotfollahi et al., *Deep packet: A novel approach for encrypted traffic classification using deep learning*, Springer, Berlin Heidelberg, Soft Computing, 2019, pp. 1–14.
33. G. Aceto et al., *Mobile encrypted traffic classification using deep learning*, in Proc. Netw. Traffic Measurement Analysis Conf., Vienna, Austria, June 2018, pp. 1–8.

AUTHOR BIOGRAPHIES

Kun Zhou received his BS degree in information engineering from Southwest Normal University, China, in 2000, and his MS degree in computer applications from the School of Electronic Engineering, Beijing University of Posts and Telecommunications, China, in 2006. From 2000 to 2003 and from 2006 to 2016, he worked for the Institute of Computer Applications, China Academy of Engineering Physics, where he is now a senior research engineer. He is now pursuing his doctorate in computer science at the University of Electronic Science and Technology of China. His research interests are cybersecurity-related issues with artificial intelligence technologies, such as machine learning and deep learning.

Wenyong Wang received a BS degree in computer science from BeiHang University, Beijing, China, in 1988, and MS and PhD degrees from the University of Electronic Science and Technology (UESTC), Chengdu, China, in 1991 and 2011, respectively. He has been the pro fessor and supervisor for PhD students in computer science and engineering. He served as a director of the Information Center of UESTC and chairman of the UESTC-Dongguan Information Engineering Research Institute since 2003. He holds many academic titles, such as an IEEE member, senior member of the Chinese Computer Federation, director of China Internet Association, and member of China Next-Generation Internet Committee of Experts. He is now a visiting professor at Macau University of Technology. His main research interests include computer networks, next-generation Internet, software-defined networking, software engineering, and artificial intelligence.

Chenhuang Wu received his BS degree in mathematics from Minnan Normal University, China, in 2007. Since then, he has been with the School of Mathematics and Finance, Putian University. Currently, he is an associate professor at Putian University and is pursuing his doctorate at the University of Electronic Science and Technology of China. His research interests include elliptic curve cryp tography and digital signatures.



Teng Hu received his BS degree from Sichuan University in 2011 and his MS degree from Beijing University of Posts and Telecommunications in 2014. Currently, he is a Ph.D. candidate in the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu. His research interests include cyber-security, big-data security, and blockchain security.

APPENDIX A

TABLE A1 Definitions for ISCX VPN-Non-VPN dataset's 23 features

No	Feature	Definitions
1	duration	Flow transmitting time
2	total_fiat	Total inter-arrival time for forward communication
3	total_biat	Total inter-arrival time for backward communication
4	min_fiat	Minimum packet inter-arrival time for forward communication
5	min_biat	Minimum packet inter-arrival time for backward communication
6	max_fiat	Maximum packet inter-arrival time for forward communication
7	max_biat	Maximum packet inter-arrival time for backward communication
8	mean_fiat	Mean of inter-arrival time for forward communication
9	mean_biat	Mean of inter-arrival time for backward communication
10	flowPktsPerSecond	Flow packets per second, pps
11	flowBytesPerSecond	Flow bytes per second, Bps
12	min_flowiat	Minimum flow inter-arrival time
13	max_flowiat	Maximum flow inter-arrival time
14	mean_flowiat	Mean of flow inter-arrival time
15	std_flowiat	Standard deviation of flow inter-arrival time
16	min_active	Minimum flow active time
17	mean_active	Mean of flow active time
18	max_active	Maximum of flow active time
19	std_active	Standard deviation of flow active time
20	min_idle	Minimum flow idle time
21	max_idle	Maximum flow idle time
22	mean_idle	Mean of flow idle time
23	std_idle	Standard deviation of flow idle time