Computers, Materials & Continua
DOI: 10.32604/cmc.2022.025103
Article

Tech Science Press

VPN and Non-VPN Network Traffic Classification Using Time-Related Features

Mustafa Al-Fayoumi[1], Mohammad Al-Fawa'reh[2] and Shadi Nashwan[3,*]

[1]King Hussein School of Computing Sciences, Princess Sumaya University for Technology (PSUT), Amman, Jordan
[2]College of Information Technology and Computer Science, Yarmouk University, Amman, Jordan
[3]College of Computer and Information Sciences, Jouf University, Aljouf, Saudi Arabia
*Corresponding Author: Shadi Nashwan. Email: shadi_nashwan@ju.edu.sa
Received: 11 November 2021; Accepted: 25 January 2022

Abstract: The continual growth of the use of technological appliances during the COVID-19 pandemic has resulted in a massive volume of data flow on the Internet, as many employees have transitioned to working from home. Furthermore, with the increase in the adoption of encrypted data transmission by many people who tend to use a Virtual Private Network (VPN) or Tor Browser (dark web) to keep their data privacy and hidden, network traffic encryption is rapidly becoming a universal approach. This affects and complicates the quality of service (QoS), traffic monitoring, and network security provided by Internet Service Providers (ISPs), particularly for analysis and anomaly detection approaches based on the network traffic's nature. The method of categorizing encrypted traffic is one of the most challenging issues introduced by a VPN as a way to bypass censorship as well as gain access to geo-locked services. Therefore, an efficient approach is especially needed that enables the identification of encrypted network traffic data to extract and select valuable features which improve the quality of service and network management as well as to oversee the overall performance. In this paper, the classification of network traffic data in terms of VPN and non-VPN traffic is studied based on the efficiency of time-based features extracted from network packets. Therefore, this paper suggests two machine learning models that categorize network traffic into encrypted and non-encrypted traffic. The proposed models utilize statistical features (SF), Pearson Correlation (PC), and a Genetic Algorithm (GA), preprocessing the traffic samples into net flow traffic to accomplish the experiment's objectives. The GA-based method utilizes a stochastic method based on natural genetics and biological evolution to extract essential features. The PC-based method performs well in removing different features of network traffic. With a microsecond per-packet prediction time, the best model achieved an accuracy of more than 95.02 percent in the most demanding traffic classification task, a drop in accuracy of only 2.37 percent in comparison to the entire statistical-based machine learning approach. This is extremely promising for the development of real-time traffic analyzers.

Keywords: Network traffic-flow; traffic classification; time-based features; machine learning; VPN traffic analysis

1 Introduction

The continued expansion of the Internet and mobile networks has increased data security concerns. Nowadays, the constant expansion of the Internet and wireless technologies allows millions of people and devices to access the internet, with network capacity expanding as a result of the ever-growing number of machines and applications that are being used. In addition, this ever-growing access has resulted in a huge volume of data flow on the Internet. For example, the launch of the Internet of Things (IoT) requires both internet access and a high level of security to ensure privacy and functionality. Therefore, the need for more advanced techniques to enhance security services and cover all aspects of data assets and network quality of service (QoS) has arisen [1]. For instance, in recent years, much effort has been devoted to studying and analyzing network traffic by characterizing it through the different protocols adopted, such as Transmission Control Protocol/Internet Protocol (TCP/IP). Accordingly, most web-based applications rely on an application layer according to the TCP/IP design nature. In the case of network traffic, there may have been additional issues about the data security being transmitted [2].

This led to the adoption of a more advanced technique for tunneling IP traffic data, as in the case of VPN connectivity. This also led to the need to maintain a special mechanism for routing traffic between nodes on public networks, in addition to hiding the identity of the senders and the receivers in terms of the IPs.

Furthermore, encryption is already ubiquitous on the Internet, making it possible to have secure conversations. As a result, the process of traffic classification becomes more difficult. Traffic classification can be divided into several categories depending on its final purpose: traffic classification based on encryption of traffic using HTTPS, traffic classification based on protocol encapsulation using IP Security protocol (IPSec) or VPN tunneling, traffic classification based on application port numbers, and traffic classification based on specific applications such as Facebook and Gmail. Through the use of multiple services, it becomes possible to identify the application and the tasks associated with it.

Individuals, small businesses, and huge corporations now deal with network security, in the same manner, they deal with physical security. The global pandemic in the past years heightened the obstacles in facing multiple attacks and raised public awareness. Due to COVID 19, remote work requires a VPN. That is, employees had to work from home and then connect to a VPN to assure encrypted connections and avoid hazards. Understanding potential threats are critical to preventing assaults. The Internet is particularly vulnerable to threats and attacks. VPN, firewalls, and intrusion detection systems (IDS) are often utilized as preventative technologies [3].

VPNs are a technology that allows safe communication over unprotected networks (such as the Internet). There are various sorts of VPN systems, each with its security strategy, pros and cons, and reliance on different protocols and standards. This evolution of VPN technology has made it the dominant method of routing Internet traffic between different internet-connected destinations. One of the most significant features of a VPN is the already encrypted IP traffic tunnel. It is practically hard to detect which program is running via the VPN tunnel ends since the IPSec protocol maintains packet-level encryption.

Classification of VPN traffic is critical for network security and data tracking, as well as improving internet security. It is also a vital aspect in identifying networks such as VPN and non-VPN traffic. We can use bandwidth more efficiently by sorting packets before sending them across the network. For performance and usefulness, Internet Service Providers (ISPs) must classify traffic. It is also the first step in prioritizing network traffic. Packet classification is an important part of QoS. QoS is a set of policies that give network traffic priority over other traffic. In particular, real-time traffic stands out, as faults can be recognized instantly (VoIP). Assuring that key applications execute smoothly is the goal of QoS [4].

However, analyzing such traffic data would require complex techniques for understanding the traffic type and its nature, and in particular VPN traffic and the applications using this type of network traffic [5,6]. Managing and optimizing network capacity for QoS has grown more complicated. Traditionally, Network Monitoring Service (NMS) analyzes network and service traffic behavior. This system's data can be used to identify the most essential links or apps that use the most bandwidth on the network, allowing network managers to quickly resolve issues and offer high-quality service to customers [7].

In contrast to most researches in this field, which focuses on studying specific application types through machine learning and statistical analysis, we propose in this paper an analytic approach to classify the VPN and non-VPN network traffic using time-related variables. This classification analysis aids in understanding the most important thing in the network visibility, namely the traffic type of the network. The focus of our work was on differentiating encrypted traffic from un-encrypted traffic; i.e., VPN and non-VPN traffic, respectively. This is a challenging issue since features always rely on the IP packet header. In addition, since encrypted traffic is the main concern, three network implementations fall within the scope of this work, namely the IPSec, P2P tunneling, and Secure Socket Layer (SSL). The machine learning classification technique used to achieve the modeling and analysis results is the random forest algorithm, which is a special and advanced implementation of decision trees to data classification.

This paper's major goal is to classify encrypted network traffic into appropriate traffic categories. The study's objectives are as follows:

- Proposing a flow-based method for categorizing encrypted traffic into distinct categories, a flow-based method for detecting VPN traffic using time-related features, this will be achieved by utilizing three algorithms: statistical features (SF), Pearson Correlation (PC), and a Genetic Algorithm (GA).
- Proposing an ML analysis approach with the aid of a random forest algorithm to classify the traffic after considering the features reduction technique based on the variance and the correlated features with normalized data, i.e., standardized features.
- Studying the ROC rate between the different traffic timeout datasets to prove its efficiency.
- Examining the proposed model's strength and ability to classify network traffic on a real dataset.
- evaluating the proposed model by measuring the value of its accuracy on the ISCX-VPN2016 dataset.

The remainder of the paper is organized as follows. Section 2 presents the literature review of network traffic classifications using machine learning. Section 3 introduces the dataset and the methodology used for performing a classification analysis of network traffic. The experimental setup and results for the adopted methodology are presented in Section 4. Section 5 presents the discussion of findings. Finally, the conclusions and recommendations for future work can be found in Section 6.

2 Literature Review

Many approaches for classifying network traffic have been proposed. Some of these approaches depend on the statistical features or characteristics extracted from the packet flow, whereas others are based on the signature of the application itself using the rule-based, black list, and white approaches. Given those approaches, research regarding the network packet attributes and flow classification began 30 years ago by Paxson [8]. The authors assumed that some statistical features are suitable to classify the protocols. Such features are flow duration, packet length, and the time required for packets' arrival. A traffic classification model suggested by Gómez Sena et al. [9] was based on the statistical features of traffic flows. Real-time traffic classification is made possible by using statistics gathered from the first few packets of data traveling in both directions of flow. They proved that only 5 or 6 first packets are enough to get accurate results. Aceto et al. [10] proposed a payload signature-based classification method (PortLoad) based on header and application signatures. It is used to improve efficiency and the likelihood of real-time traffic classification. The high processing costs of this approach burden the network. This is because deep packet inspection (DPI) uses regular expressions to match packet payload data. Due to encryption and dynamic ports, DPI-based methods can only identify traffic encoded in the expression library. The amount of processing required by this method puts a lot of strain on the network. DPI also can't decrypt encrypted traffic. Yeganeh et al. [11] proposed a new technique to identify the traffic based on signatures; this approach reduced the time needed to classify the traffic flow. Nevertheless, it caused an issue in the detection process of unknown flow or customized signatures. It is noticeable that all those studies were based on packets size and they looked at the first few packets of the network flow to extract the statistics to enhance the performance of the classification process with high-scale and high-speed networks. However, despite the diversity of those approaches, for all of them, the statistics were the main concern rather than adopting data modeling to identify the patterns in the network traffic. In addition, all of them failed to detect unknown or manually generated signatures.

From another perspective, mostly related to data aggregations. Rao et al. in [12] focused in their research on characterizing video streams, namely Netflix and YouTube. Wang et al. [13] focused in their research on P2P traffic by extracting the features from many flows, which were then aggregated into clusters to detect P2P application behavior. From another point of view, others focused on the iMessage protocol and relied on it to extract the device type [14]. Furthermore, the authors in [15] proposed a new technique to handle the privacy of connections through deep packet inspection. This was a proposal that can inspect the payload of encrypted traffic without decryption; however, there is was a limitation to this proposal, which is that it handled HTTPS traffic only. At the same time, others presented a technique to detect Web-Real-Time communication encrypted traffic [16]. Mamun et al. [17] focused on identifying encrypted traffic by calculating the entropy of the payload of the packets. However, the novelty in those approaches relied on the communication type and the assets communicated.

On the other hand, several machine learning classification techniques consider the network packet flow [18–24], where they mainly consider investigating further data features to identify the packets in terms of telling apart the encrypted ones from those that are not. In 2015, Velan et al. [25] provided a complete comparison of different classification methods based on their features and showed their strengths and weaknesses. They described the well-known encryption protocols used throughout the Internet. Therefore, the structure of the protocol showed the initiation of an encrypted connection and provides a large amount of information for the classification and analysis of encrypted traffic. They surveyed different methods of classification of the payload and feature-based classification of encrypted traffic and categorized them using a set taxonomy. They stated that different methods could

be used to recognize the encrypted application protocol in addition to the encryption protocol. The majority of studies rely on a wired network that is managed by the same network management domain.

In 2016, Draper-Gil et al. [26] studied the effectiveness of flow-based time-related features for detecting VPN traffic and categorizing encrypted traffic into several categories, such as browsing and streaming. To determine the validity of their features, they used two widely used machine learning approaches (C4.5 and KNN). As a result of their research, they asserted that time-related factors are effective classifiers for encrypting communications, providing excellent accuracy and performance.

In 2017, Bagui et al. [27] worked on the same dataset used in this paper. They applied six (GBT, KNN, LR, NB, RF, and SVM) types of machine learning algorithms and compared the achieved results. In their results, a high accuracy rate was achieved with two shortlisted algorithms, namely GBT and RF; their accuracy was 0.946% and 0.944%, respectively.

In 2018, Leroux et al. [28] proposed an approach based on encrypted streams. They employed machine learning techniques including Naive Bayes, Logistic Regression, and Random Forest to predict the sort of traffic passing through an IPsec or TOR tunnel. So they got packet size, interarrival time, and burst time, burst size from the encrypted streams. Experiments have shown that the proposed approach can be used in real-time applications such as QoS systems. In 2018, Zion et al. [29] created a new set of network traffic ML features. They gathered a dataset consisting of about 28,000-time frame samples and used it to demonstrate the high-performance categorization of mobile and desktop main and secondary traffic and desktop application activities, with accuracies of 81%, 94%, and 93%, respectively. Furthermore, they demonstrated that, in contrast to existing approaches that rely on a small number of robust ML characteristics, their system is resilient and performs well even when intelligent users aggressively modify and conceal their traffic (e.g., using a VPN).

In 2020, Miller et al. [30] presented a computational approach to detect VPN traffic, where the flow statistics data found in captured network packets' TCP headers were extracted. A multi-layered perceptron neural network with 10-fold cross-validation was used. Experiments revealed that the suggested model could accurately categorize network traffic as VPN or non-VPN. The author focused on two types of VPN applications, namely OpenVPN and Stunnel OpenVPN. The overall accuracy of OpenVPN was 93.71% and Stunnel OpenVPN achieved 97.82% accuracy.

In 2020, Pacheco et al. [31] proposed a new satellite-based Internet traffic classification approach to improve QoS. They used machine learning and deep learning techniques to classify Internet traffic. A complete assessment of the components necessary to connect an ML/DL solution to recognized satellite communication and QoS management infrastructure. A large and diverse set of Internet traffic was necessary to develop this solution. In this context, the data were generated using an emulated satellite communication environment, where other Internet communications are launched and captured. The proposed model focused on encrypted, unencrypted, and tunneling communications. Some experiments were conducted on a cloud platform to validate the proposal and set guidelines for its deployment over a satellite architecture.

In 2020, Guo et al. [32] presented two deep learning models to categorize the traffic into VPN and non-VPN traffic. In this paper, the authors used Convolutional Auto-Encoding (CAE) and a Convolutional Neural Network (CNN) to distinguish encrypted network traffic. The traffic generated by six distinct applications was converted into session pictures to be suitable for the deep learning models; the CNN method used two-dimensional local features of the images. The CAE-based method focused on an unsupervised method to extract the hidden layer features. Learning the nonlinear link between input and output is achievable. Several experiments were conducted using these approaches, and the best accuracy achieved by the CCN was 92.92%, while the CAE model achieved

98.77% accuracy. The authors claimed that their models outperform more established methods of identification.

The authors of [33] introduced a novel framework to carry out traffic classification at any layer on the radio network stack. An RNN-based baseline architecture was described, and its performance was benchmarked on three TC workloads at different radio stack layers. To evaluate their model, they integrated a waveform generator that was compliant with standards for 802.11 radio technology with packet traces from genuine communications. According to the results of the performance evaluations, the top model obtained an accuracy of more than 92% in the most challenging TC job.

In 2021, Aswad et al. [34] used Apache Spark and an Artificial Neural Network (ANN) to identify VPN traffic based on time-related features. The proposed approach was based on a VPN with multiple node features. When a packet is received by a node, no further processing is required because all routing processing has already been completed; the node's only responsibility is to transfer the packet to the proper next hop when the time arrives and it is visible to contact. The proposed implementation of the ANN prevents unnecessary processing and flooding occurs in standard VPN traffic classification. The dataset was divided into two parts: 80% for training and 20% for testing, with 10 k folds with 50 epochs. ABN and the Apache Spark engine were used to implement VPN network traffic flow classification, to the best of our knowledge. Experiments showed that the precision for VPN classification was 96.76%, whereas for non-VPN it was 92.56%. The proposed model was evaluated using the CIC-Darknet2020 and ISCXVPN2016 datasets. The ANN + Spark Engine technique outperformed the convolutional neural network compared to the stacked auto-encoder network.

In 2021, in Lu et al.'s work [35], the difficulties of traffic feature extraction efforts were avoided by focusing on DL to categorize network traffic as encrypted or not. The authors proposed an Inception-LSTM (ICLSTM) service to detect encrypted traffic, in which they converted the traffic data to gray images and then built an ICLSTM neural network to extract the important features and conduct effective traffic classification. In the training phase, the authors avoided the category imbalance problem by setting different weighting parameters for each category, which makes it more realistic and symmetrical for different types of encrypted traffic, and thus the detection effect is more balanced and reasonable. The proposed module was validated using a public dataset called ISCX 2016, where the accuracy for both encrypted and non-encrypted traffic was 98%.

Tab. 1 summarizes the related work regarding the classification of network traffic considering VPN traffic as one type of network traffic.

Table 1: Summary of related studies and their techniques

Author	Used techniques	Performance
[8] Paxson	Statistical-based	-
[9] Gómez Sena et al.	Statistical-based on first few first packet	-
[10] Aceto et al.	deep packet inspection	97.45%
[11] Yeganeh et al.	Signature-based	90%
[12] Rao et al.	Statistical-based	*
[13] Wang et al.	Behavior-based	90%
[14] Coull et al.	Signature-based	96%
[15] Sherry et al.	Deep packet inspection-based	-

(Continued)

Table 1: Continued

Author	Used techniques	Performance
[16] Di Mauro et al.	Machine learning-based	*
[17] Mamun et al.	Machine learning-based, through calculating the entropy of the payload	98.10%
[18] Bernaille et al.	Statistical-based	85%
[19] Moore et al.	Statistical, based on naïve Bayes algorithm	95%
[21] Iliofotou et al.	Behavior-based, Traffic dispersion graphs (TDGs)	*
[22] Karagiannis et al.	Behavior-based	95%
[25] Velan et al.	payload and feature-based classification	-
[26] Draper-Gil et al.	Machine learning-based	>80
[27] Bagui et al.	Machine learning-based	90
[28] Leroux et al.	Machine learning-based	83.87%
[29] Zion	Machine learning-based	93%
[30] Miller et al.	Deep learning-based	97.82%
[31] Pacheco et al.	Satellite architecture, deep learning, and machine learning	91.51
[32] Guo et al.	Deep learning-based	92.92%
[33] Camelo et al.	Deep learning-based	92
[34] Aswad et al.	Deep learning and apache spark based	96.76%;
[35] Lu et al.	Deep learning-based	98%

However, protocols used to hide users' identities and protect their privacy are challenging in terms of the classification for their encrypted properties. Nevertheless, as far as we know, few have proposed methods to classify VPN traffic based on the time-related features extracted from the network traffic and machine learning techniques. In addition, the previous experiments used Weka [36] and deployed KNN, C 4.5, and other supported algorithms in Weka. By contrast, this work is using the random forest algorithm considering the SCIKIT-LEARN python package [37], along with feature reduction coding practices based on correlations between the features and standardizing (normalization) of different packets' timeout, recorded in the given dataset and used to classify and obtain high score results. The only limitation or constraint that was experienced was the size of the dataset since this type of traffic analyzer is suitable for small to medium environments.

3 Methodology

Fig. 1 describes in detail the steps followed in this research paper. In addition, the following subsections detail each step in the figure.

3.1 Dataset

The ISCX-VPN2016 dataset was used to develop a model in this study [26]. The University of New Brunswick (UNB) made this dataset available for public use. It was collected by capturing experimental network traffic on a real-time basis in a customized lab. This data set takes into account seven different types of traffic: web browsing, email, chat, streaming, file transfer, VoIP, and P2P. In addition, the traffic that has been acquired has been divided into two types of traffic based on its characteristics. The first one is the normal traffic of these applications, and the second one is related to the traffic of these

applications over VPN. The traffic was captured with Wireshark and TcpDump [38]. The generated traffic was around 28 GB; other supporting tools were also used to construct this dataset, such as OpenVPN, which was used to generate VPN traffic, and FileZilla, for file transfer traffic. The dataset was then filtered to contain only the traffic that was just initiated by or targeted Alice and Bob IP addresses, which represent the two customized endpoints needed in the network to collect the required dataset. Finally, the ISCXFlowMeter tool [26] was used to extract the features from the dataset of traffic packets, representing the numerical inputs for the machine learning models' implementation in further stages of our experiments.

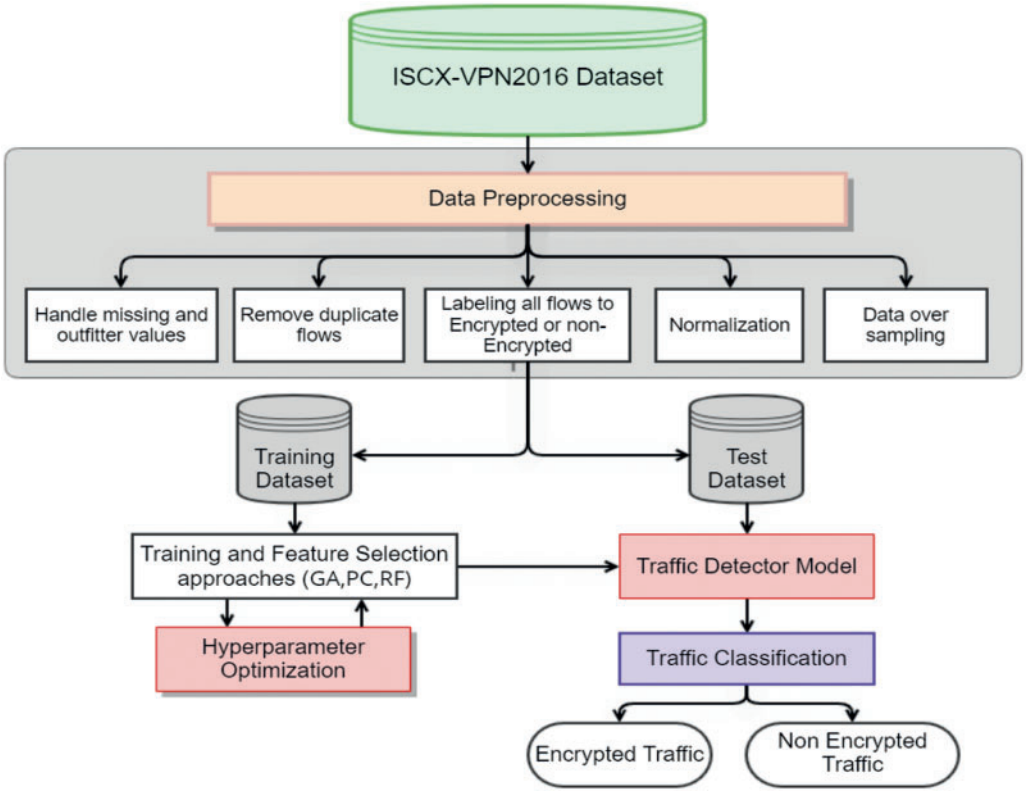


Figure 1: Work methodology

The final processed features dataset was then fed into two analysis scenarios: Scenario A was used to detect if the traffic is VPN or not and then tried to build classifiers to distinguish the type of encrypted traffic over VPN network connectivity. Scenario B used the labeled dataset, i.e., VPN and non-VPN instances, to form the second labeled dataset, in which the applications used to conduct the network-generated traffic were used as another variation of the instances' labeling.

This dataset gained extra value related to the type of the features, which were either packet-based or time-based. Packet-based are those features that rely more on the protocol of the networking concept conducted, such as IPs, ports, and the flow of bytes amount. By contrast, the time-based features are considered as another dimension for the analysis that does not rely only on the statistic elements of the data, but rather on the variability of the time-based features. Timeouts for the time-based features were captured according to the discrete values that were measured in seconds, including 15, 30, 60, and 120 s.

3.2 Data Preprocessing

In general, the work carried out concerning the data preprocessing started with and focused only on the 60 s dataset that has both labels; i.e., VPN and non-VPN. The data cleaning took place, followed by a data normalization process, to study its effects on the learning achieved through further experiments modeled on machine learning. The normalization took place in terms of calculating the Z score values; i.e., calculating the division of the values reduced from their mean by their standard deviation. The nominal features were dropped from the dataset to avoid tackling the models/performance. However, and for gaining a clear picture of the methodology adopted, the work on the dataset considered the time-based features in the later stages, but the cleaning and preparation for the data followed a standardized process that started loading the data and combined the classified files for VPN and non-VPN. Then, infinity literals and other literals that affect the modeling process were detected and fixed. Moreover, nulls were also checked to confirm the inexistence of the missed values in the given dataset. The final features set considered for the following analysis stages contained 24 features and 15545 instances. The features list included Class1, Duration, Flowbytespersecond, Flowpktspersecond, Label, Max_Active, Max_Biat, Max_Fiat, Max_Flowiat, Max_Idle, Mean_Active, Mean_Biat, Mean_Fiat, Mean_Flowiat, Mean_Idle, Min_Active, Min_Biat, Min_Fiat, Min_Flowiat, Min_Idle, Std_Active, Std_Flowiat, Std_Idle, Total_Biat, and Total_Fiat. The Class1 feature represents the applications used over the captured network traffic, while the Label feature is the binary one related to the traffic encryption state, i.e., VPN or non-VPN. It is important to mention here that this work replaced the given infinity values with the mean along with the negative values before ML modeling.

3.3 Exploratory Data Analysis

We next explored the data for the given dataset; for example, we generated the correlation matrix for all of the features as shown in Fig. 2 to study the effects of the features on each other using the Pearson correlation method. This correlation matrix shows that the Max_Flowiat–Max_Idle pair was highly correlated; other pairs were similarly correlated, thus necessitating dropping one of each pair to optimize the performance of the classifier by reducing the redundant features in the given dataset. Another important exploration step for understanding the classes' distribution within the given dataset instances was also conducted and found to be as shown in Fig. 3, where 55.3% of the data instances were classified as non-VPN and the remaining, i.e., 44.7%, were classified as VPN when the time interval 60 s. In addition, Fig. 3 also shows the distribution of the dataset in each class over different time intervals (15, 30, 120). Other exploration tasks were also conducted to better understand the dataset; those included understanding the outliers as shown in Fig. 4, to determine whether the class imbalance problem will affect the analysis.

3.4 Feature Engineering

This work focuses on features' reduction and selection more than upon performing features extraction processes, as the ISCXFlowMeter tool has already done the job during the network packets' early preprocessing presented by the authors [26].

Starting with feature reduction, two approaches were considered, where the first consists of using a small amount of variance to detect the weak features and hence pruning them. This approach yields no low variance-based features to reduce. The second approach consists in using the multicollinearity between the features.

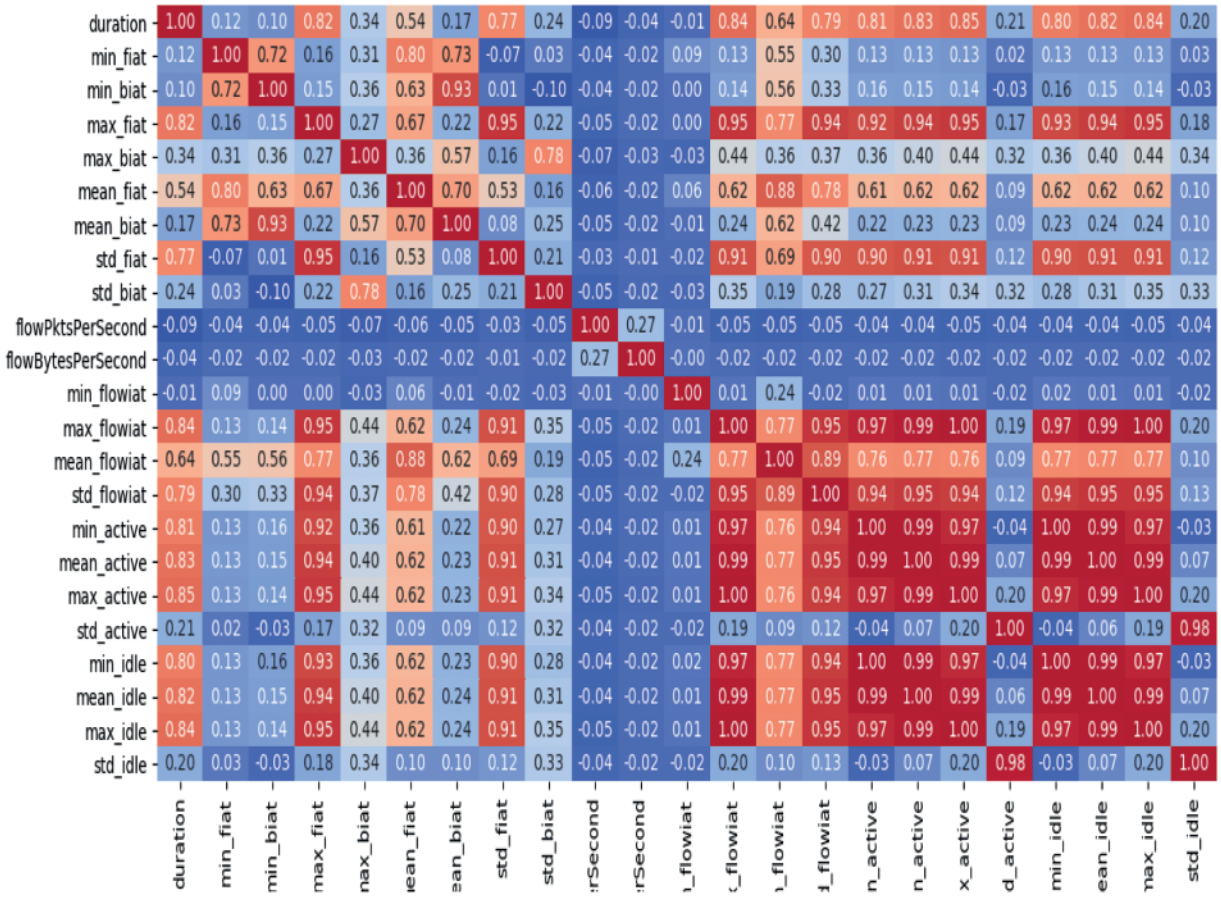


Figure 2: All features correlation matrix

Fig. 2 shows the multicollinear matrix found for the given dataset. Those mathematical and statistical relationships were discovered using the Spearman correlation coefficient. This approach yielded nine important and reduced features. These features are listed in Tab. 2. As can be seen in Tab. 2, after the feature reduction, we still have very important network traffic features. The first one is the duration of the network flow, which can give us an indicator of the network traffic type; for instance, TCP has an average duration time similar to UDP. Another feature related to the active session, which determines the time the flow was active before going idle, consists of three values, which can also indicate the type of network traffic in general and for VPN traffic in particular because in the case of VPN all of the traffic will be covered inside the VPN traffic. Another important feature is the flow byte per second; this value shows how many bytes are transmitted over each network flow and depends on the type of traffic itself. In addition, we have three values of the Biat feature, namely min, max, and mean; Biat is the time between two packets sent backward in the network flow. The resulting Spearman correlation matrix for those nine reduced features is shown in Fig. 5.

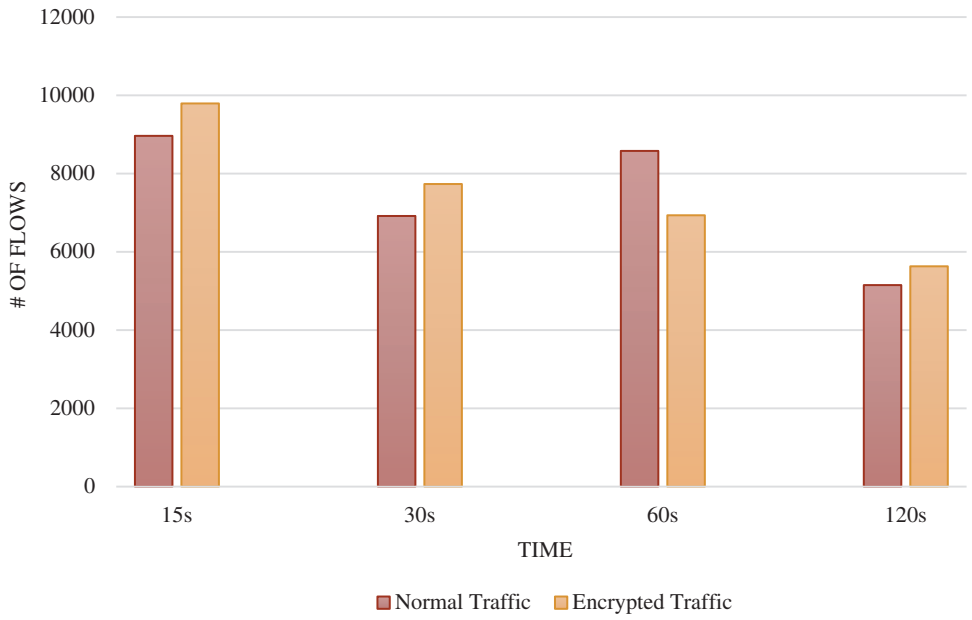


Figure 3: Classes distribution over different time intervals (15, 30, 60, 120)

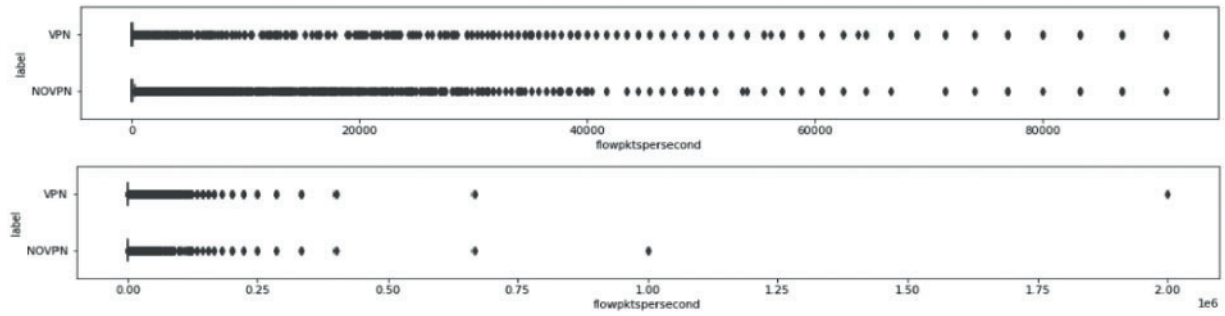


Figure 4: Outliers comparison

Another feature selection approach is GA-based feature selection, which picks optimum features by selecting a population at random and checking its fit to the environment using an objective function. It also employs a looped evolutionary process to enhance the population until it reaches the optimum [16]. When GA was applied to the processed data, which had 24 features, eight were chosen as important features and the rest were disregarded during the model's classification phase. The most important features after applying the GA were total_biat, max_fiat, max_biat, flowPktsPerSecond, flowBytesPerSecond, min_flowiat, max_flowiat, and mean_flowiat.

3.5 Dataset Training and Testing

Ten-fold cross-validation (CV) was utilized in this paper to evaluate the classification models. The data were randomly split into 10 sets of data using this approach; nine sets were utilized as training sets, while the tenth was kept aside for testing. With different training sets, this process was repeated 10 times. Cross-validation can be seen in Fig. 6.

Table 2: Features descriptions

Feature	Description
Duration	The duration of the flow
Flow bytes per second	Flow byte per second
Max_Active	The maximum time a flow was active before going idle
Max_Biat	Backward inter-arrival time, the time between two packets sent backward
Max_Flowiat	Maximum forward inter-arrival time, the time between two packets sent forwards
Mean_Biat	Average of backward inter-arrival time, the time between two packets sent backward
Min_Active	The minimum time a flow was active before going idle
Min_Biat	Minimum backward inter-arrival time, the time between two packets sent backward
Std_Active	The standard deviation of a value of time a flow was active before going idle

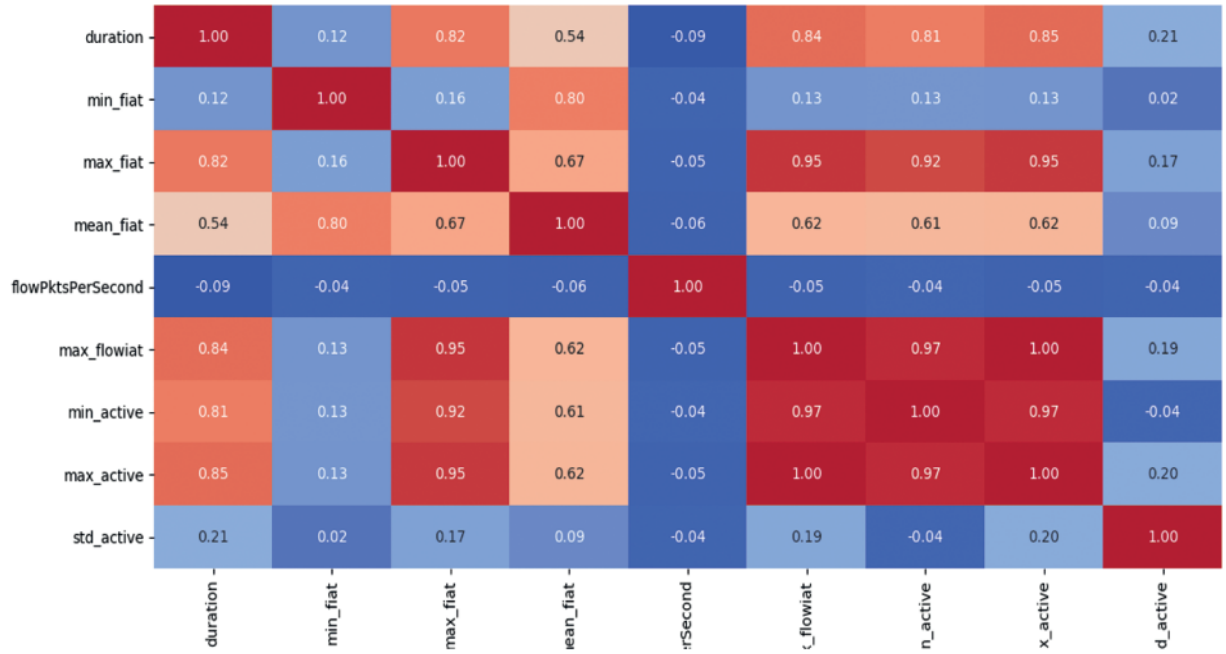


Figure 5: Correlation matrix for reduced features set

4 Experimental Practice and Setup

Random forests [39] is an ensemble learning algorithm that is used to perform a binary classification process on the presented dataset. The dataset is either processed in the normal state or using a normalization process for the values given to the reduced features among all instances. Tab. 5 summarizes the results of the classification experiments conducted, whereas Fig. 7 summarizes

the results achieved in terms of a receiver operating characteristic curve (ROC). In the proposed architecture, RF as a machine learning model is utilized, which is described in Algorithm 1. The general configurations for the RF classifier are proposed. Tab. 3 summarizes the proposed model parameters configurations that are used to conduct the experiments for detecting network traffic.

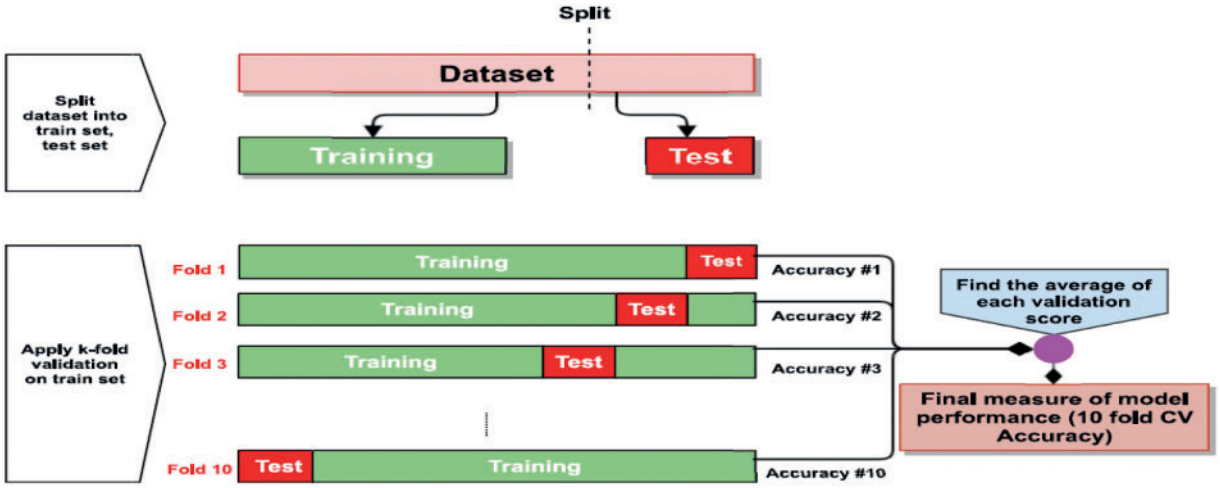


Figure 6: Cross-validation process

Algorithm 1: Random Forest pseudocode

Input Parameters: A training set $\boldsymbol{S}$, set of Features $\boldsymbol{F}$
Output: Subset of features $\boldsymbol{f}$
Precondition: a training set $\boldsymbol{S} :== (x_1, y_1), \cdots, (x_n, y_n)$, features F, and the number of trees in forest B.

Function ***RandomForset*** $(\boldsymbol{S}, \boldsymbol{F})$
{
  $\boldsymbol{H} \leftarrow 0$
  for $\boldsymbol{i} \in 1, \cdots, \boldsymbol{B}$
  {
    $\boldsymbol{S^i} \leftarrow$ ***A bootstrap sample from*** $\boldsymbol{S}$
    $\boldsymbol{h_i} \leftarrow$ ***RandomizedTreeLearn*** $(\boldsymbol{S^i}, \boldsymbol{F})$
    $\boldsymbol{H} \leftarrow \boldsymbol{H} \cup \boldsymbol{h_i}$
  }
  return $\boldsymbol{H}$
}
Function ***RandomizedTreeLearn*** $(\boldsymbol{S}, \boldsymbol{F})$
{
  At each node:
  $\boldsymbol{f} \leftarrow$ ***very small subset of*** $\boldsymbol{F}$
  Split on best feature in $\boldsymbol{f}$
  return learned tree
}

Table 3: Proposed RF classifier model hyperparameters settings

Hyperparameter	Value
Size of each bag	100
Number of iterations	100
Number of execution slots	1
Number of attributes to randomly investigate	0
Minimum number of instances	1
Minimum variance for split	1e−3
The maximum depth of the tree	0
Batch-size	100
Seed	1

To get the best result from the model, the hyperparameter values of the genetic algorithm were built, tested, and set. They will be used to select the optimal features in the proposed final structure for classifying network traffic. Therefore, the general configurations for the GA algorithm are proposed. Tab. 4 summarizes the GA algorithm parameters settings that are used to conduct the experiments for detecting network traffic.

Table 4: Proposed GA algorithm hyperparameters settings

Hyperparameter	Value
Cross over Probability	0.6
Maximum generations	20
Mutation probability	0.0.33
Population size	20
Report to frequency	1
Seed	1

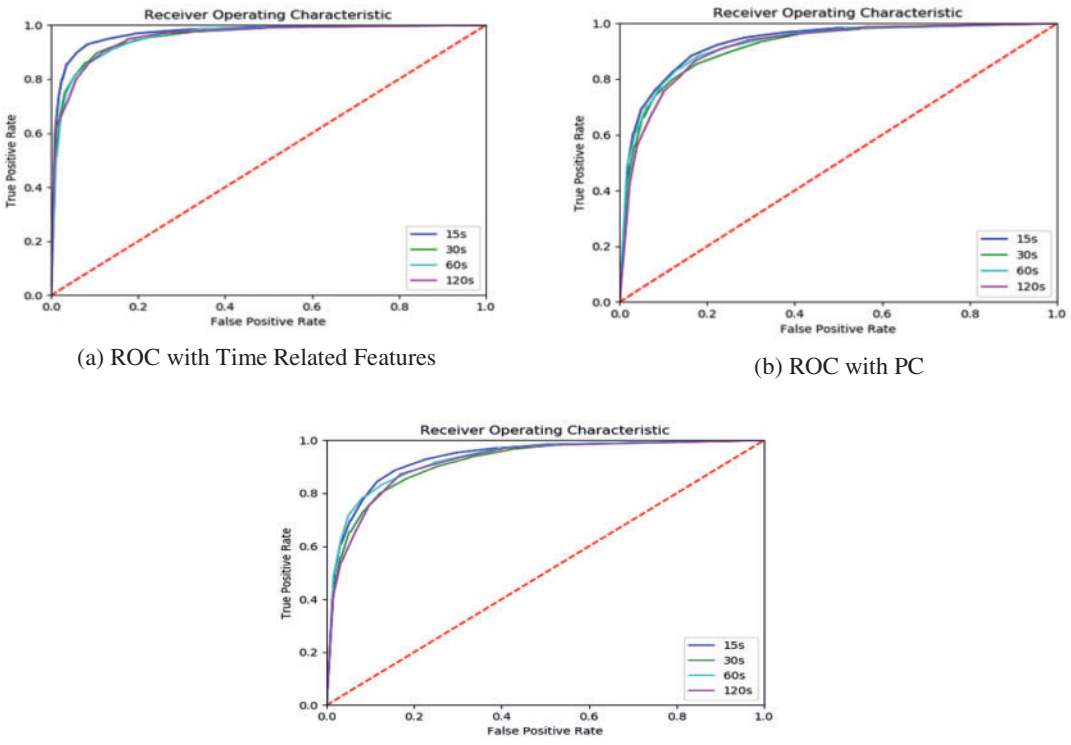
4.1 Generalizing the Approach for the Other Datasets

The same approach for the analysis also expanded to handle the rest of the datasets that were originally split in terms of the timeout factor during the construction process. Those datasets are either 15, 30, or 120 s. Tab. 5 summarizes the results of the classification experiments conducted, whereas Fig. 7 summarizes the results achieved in terms of a receiver operating characteristic curve (ROC).

Table 5: Experimental results

Timeout	Class label	Feature selection	No. of features	Precession	Recall	F1	ROC area	Accuracy
15	Non-VPN	N	24	0.954	0.97	0.962	0.986	95.02%
	VPN		24	0.942	0.914	0.928	0.986	
	Non-VPN	Y	9	0.907	0.949	0.927	0.96	90.33%
	VPN		9	0.896	0.819	0.856	0.96	
	Non-VPN	GA	8	0.927	0.907	0.917	0.976	92.06%
	VPN		8	0.915	0.933	0.924	0.976	
30	Non-VPN	N	24	0.946	0.966	0.956	0.98	94.12%
	VPN		24	0.932	0.893	0.912	0.98	
	Non-VPN	Y	9	0.905	0.946	0.925	0.95	89.92%
	VPN		9	0.886	0.809	0.846	0.95	
	Non-VPN	GA	8	0.939	0.956	0.947	0.976	93.01%
	VPN		8	0.912	0.88	0.896	0.976	
60	Non-VPN	N	24	0.927	0.891	0.908	0.978	93.06%
	VPN		24	0.933	0.956	0.944	0.978	
	Non-VPN	Y	9	0.881	0.832	0.856	0.952	89.14%
	VPN		9	0.898	0.929	0.913	0.952	
	Non-VPN	GA	8	0.908	0.876	0.892	0.974	91.76%
	VPN		8	0.923	0.944	0.934	0.974	
120	Non-VPN	N	24	0.92	0.891	0.906	0.978	93.50%
	VPN		24	0.943	0.958	0.95	0.978	
	Non-VPN	Y	9	0.854	0.817	0.835	0.943	88.71%
	VPN		9	0.904	0.925	0.914	0.943	
	Non-VPN	GA	8	0.91	0.87	0.89	0.974	92.47%
	VPN		8	0.932	0.954	0.943	0.974	

Fig. 7 depicts a ROC curve in three cases: RF with time-related features, RF with PC, and RF with GA, where the experiments were conducted with different time intervals 15, 30, 60 and 120 s). As shown in Fig. 7 the highest ROC in all cases is RF with time-related features.



(a) ROC with Time Related Features

(b) ROC with PC

(c) ROC with GA

Figure 7: ROC result clusters in terms of feature reduction impact

4.2 Evaluation Metrics

The discrimination evaluation of the best (optimal) solution during classification training can be defined for binary classification analysis using the confusion matrix, as illustrated in Tab. 6. The predicted class is represented by the table's columns whereas the actual class is represented by the table's row. TP and TN denote the number of correctly recognized positive and negative instances in this confusion matrix. FP and FN, on the other hand, represent the number of misclassified negative and positive cases, respectively [40].

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{1}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{TN}} \tag{2}$$

$$\mathrm{F} - \mathrm{Measure} = \frac{2 * P * R}{\mathrm{P} + \mathrm{R}} \tag{3}$$

The receiver operating characteristic curve (ROC curve) is a graphical representation of sensitivity. The ROC curve plots sensitivity (TPR) against specificity (FPR) for various test cut-off points. For various test cut-off points, the sensitivity (TPR) on the y-axis and (1–specificity) (FPR) on the x-axis are plotted. For the sake of simplicity, it is typically depicted as a square box, with both axes ranging from 0 to 1.

Table 6: Confusion matrix

	Predicted positive	Predicted negative
Actually positive	True positive (TP)	False negative (FN)
Actually negative	False positive (FP)	True negative (TN)

Tab. 6 can be used to construct a variety of regularly used metrics, such as equations ranging from (1) to (3). The recall (R) is a metric that indicates how well a dataset can be searched for relevant instances, while the precision (P) expresses the proportion of the data points that the model says was relevant and where relevant. Finally, the F-Measure (FM) can be used to calculate the harmonic mean of recall and precision. There are, however, a slew of additional options. However, they can all be utilized to calculate classification and, as a result, to judge the model's quality in classification procedures [33].

4.3 Experimental Machine

All of the previously described analyses as well as the results of all experiments were carried out on machines with the hardware specifications shown in Tab. 7.

Table 7: Experimental environment HW specifications

Brand	ThinkPad E560
RAM	16GB
HD	250GB SSD
System processor	Intel(R) Core(TM) i7-6500U CPU © 2.50 GHz 2.60
OS	Windows 10 enterprise

5 Discussion of Results

The experimental results show that the effectiveness of this approach is very high. However, the results show that data with the timeout length of 15 s always yielded the best results in terms of precision and recall, as well as the derived F1 measure and accuracy, for both reduced and non-reduced feature sets, except for GA. However, the experiments showed another important result for the classification process with other timeout values (30, 60, and 120 s), where the reduced features and the non-reduced features experiments yielded similar results, especially for the 15 s timeout dataset, where the accuracy reached 95.02%, 94.12%, 93.06% and 93.50% for 15, 30, 60 and 120 s, respectively when all-time features were used. However, using the GA with 15 and 30 s timeouts achieved the same ROC. It is also worth noting that class imbalance issues have been addressed since the classes began employing the SMOTE algorithm, and the classes are now balanced.

The achieved accuracy rate for the proposed models was 95%. In comparison to previous studies (as stated in Tab. 8), the proposed approach can distinguish between encrypted and non-encrypted queries more efficiently, with acceptable FP and FN rates. In conclusion, the proposed model is adequate to build a network traffic classifier that organizations can rely on to protect their environment.

Table 8: Comparison of our model and other studies (where A: accuracy, P: precision)

Study	Feature extraction algorithm	Classification algorithms	Highest achieved metric
[41]	NO	ML/DL	95–98 P
[32]	NO	DL	98.77%, 92.92% A
[27]	NO	CNN + metric learning	98.53% P
[34]	NO	DL	96.76%; 92.56%. P
[42]	NO	ML	94.6% A
[35]	NO	DL	98% A
RF-ST	NO	ML	95.02% A
RF-PC	YES	ML	90.33% A
PF-GA	YES	ML	93.0141% A

6 Conclusion

This paper's goal was to classify encrypted network traffic data into the appropriate traffic category. This could help ISPs provide a better quality of service to their clients, as well as monitoring and controlling the network, safeguarding the network from viruses and attacks, and performing appropriate network planning and troubleshooting. To achieve this, time-related features have been used to characterize VPN and non-VPN traffic. In addition, we proposed an ML analysis approach with the aid of a random forest algorithm to classify the traffic after considering the features reduction technique based on the variance and the correlated features with normalized data; i.e., standardized features. The proposed approach achieved a high ROC rate of between 95% and 98.6% for the different traffic timeout datasets, which proves its efficiency. Accordingly, we applied the same approach to the rest of the datasets (15, 30, and 120 s) to consider the different timeout recordings, and the same results of accuracies were achieved. It should be noted that results obtained while not carrying out the required feature reduction would yield lower accuracies along with model overfitting problems. It seems that this approach will provide valuable benefits to the literature due to its ability to classify data while maintaining privacy for the VPN traffic networks. In the future, we will increase the number of traffic classes that the behavior analyzer can detect. Furthermore, advanced methods will be applied for hyperparameter optimization and feature selection, such as earthworm optimization algorithm and PCA.

Acknowledgement: We are very grateful to the reviewers for their valuable comments that helped improve the paper. We wish to express our gratitude to all members of our colleges, Princess Sumaya University for Technology (PSUT), Jouf University, and Yarmouk University, for their support.

Author Contributions: Conceptualization, methodology and writing—original draft preparation, all authors; writing—review and editing, M. Al-Fayoumi. All authors have read and agreed to the published version of the manuscript.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Lee, A. Abdullah, N. Jhanjhi and S. Kok, "Classification of botnet attacks in IoT smart factory using honeypot combined with machine learning," *PeerJ Computer Science*, vol. 7, pp. 1–23, 2021.

[2] T. Kobbaey, R. Hamzaoui, S. Ahmad, M. Al-Fayoumi and N. Thomos, "Enhanced collision resolution and throughput analysis for the 802.11 distributed coordination function," *International Journal of Communication Systems*, vol. 34, no. 16, pp. e4953, 2021.

[3] M. Al-Fawa'reh, M. Al-Fayoumi, S. Nashwan and S. Fraihat, "Cyber threat intelligence using PCA-DNN model to detect abnormal network behavior," *Egyptian Informatics Journal*, 2021. https://doi.org/10.1016/j.eij.2021.12.001.

[4] M. Fawa'reh, M. Qasaimeh, I. AbuArja and M. Al-Fayoumi, "Mitigating deep learning attacks against text image CAPTCHA using arabic scheme," *International Journal on Communications Antenna and Propagation (IRECAP)*, vol. 11, no. 4, pp. 288–296, 2021.

[5] R. Faek, M. Al-Fawa'reh and M. Al-Fayoumi, "Exposing bot attacks using machine learning and flow level analysis," in *Proc. DATA'21*, Ma'an, Jordan, pp. 99–106, 2021.

[6] A. V. Elijah, A. Abdullah, N. Z. JhanJhi, M. Supramaniam and O. Balogun Abdullateef, "Ensemble and deep-learning methods for two-class and multi-attack anomaly intrusion detection: An empirical study," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 9, pp. 520–528, 2019.

[7] A. Homsi, J. Al Nemri, N. Naimat, H. A. Kareem, M. Al-Fayoumi *et al.,* "Detecting twitter fake accounts using machine learning and data reduction techniques," in *Proc. of the 10th Int. Conf. on Data Science, Technology and Applications*, DATA 2021, SciTePress, Lieusaint - Paris, France, pp. 88–95, 2021.

[8] V. Paxson, "Empirically derived analytic models of wide-area TCP connections," *IEEE/ACM Transactions on Networking*, vol. 2, no. 4, pp. 316–336, 1994.

[9] G. Gómez Sena and P. Belzarena, "Early traffic classification using support vector machines," in *Proc. of the 5th Int. Latin American Networking Conf.*, Pelotas, Brazil, pp. 60–66. 2009.

[10] G. Aceto, A. Dainotti, W. De Donato and A. Pescapé, "PortLoad: Taking the best of two worlds in traffic classification," in *Proc. INFOCOM IEEE Conf. on Computer Communications Workshops*, San Diego, CA, USA, pp. 1–5, 2010.

[11] S. H. Yeganeh, M. Eftekhar, Y. Ganjali, R. Keralapura and A. Nucci, "Cute: Traffic classification using TErms," in *Proc. ICCCN'21*, IEEE, Munich, Germany, pp. 1–9, 2012.

[12] A. Rao, A. Legout, Y. Lim, D. Towsley, C. Barakat *et al.,* "Network characteristics of video streaming traffic," in *Proc. of the Seventh Conf. on Emerging Networking Experiments and Technologies*, Tokyo, Japan, pp. 1–12, 2011.

[13] D. Wang, L. Zhang, Z. Yuan, Y. Xue and Y. Dong, "Characterizing application behaviors for classifying p2p traffic," in *Proc. 2014 Int. Conf. on Computing, Networking and Communications (ICNC)*, Honolulu, HI, USA, pp. 21–25, 2014.

[14] S. E. Coull and K. P. Dyer, "Traffic analysis of encrypted messaging services: Apple iMessage and beyond," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 5–11, 2014.

[15] J. Sherry, C. Lan, R. A. Popa and S. Ratnasamy, "Blindbox: Deep packet inspection over encrypted traffic," in *Proc. of the 2015 ACM Conf. on Special Interest Group on Data Communication*, New York, NY, USA, pp. 213–226, 2015.

[16] M. Di Mauro and M. Longo, "Revealing encrypted WebRTC traffic via machine learning tools," in *Proc. 12th Int. Joint Conf. on e-Business and Telecommunications (ICETE)*, Colmar, France, vol. 4, pp. 259–266, 2015.

[17] M. S. I. Mamun, A. A. Ghorbani and N. Stakhanova, "An entropy-based encrypted traffic classifier," in *Proc. Int. Conf. on Information and Communications Security*, Beijing, China, pp. 282–294, 2015.

[18] L. Bernaille and R. Teixeira, "Early recognition of encrypted applications," in *Proc. Int. Conf. on Passive and Active Network Measurement*, Louvain-la-neuve, Belgium, pp. 165–175, 2007.

[19] A. W. Moore and D. Zuev, "Internet traffic classification using Bayesian analysis techniques," in *Proc. ACM SIGMETRICS*, Banff, Alberta, Canada, vol. 33, pp. 50–60, 2005.

[20] M. Al-Fawa'reh and M. Al-Fayoumi, "Detecting stealth-based attacks in large campus networks," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, pp. 4262–4277, 2020.
[21] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh *et al.,* "Network monitoring using traffic dispersion graphs (tdgs)," in *Proc. of the 7th ACM SIGCOMM Conf. on Internet Measurement*, San Diego, California, USA, pp. 315–320, 2007.
[22] T. Karagiannis, K. Papagiannaki and M. Faloutsos, "BLINC: Multilevel traffic classification in the dark," in *Proc. SIGCOMM'05*, New York, NY, USA, ACM, pp. 229–240, 2005.
[23] M. Al-Fayoumi, J. Alwidian, M. Abusaif and I. M. East, "Intelligent association classification technique for phishing website detection," *International Arab Journal of Information Technology*, vol. 17, no. 4, pp. 488–496, 2020.
[24] S. Al-Saqqa, M. Al-Fayoumi and M. Qasaimeh, "Intrusion detection system for malicious traffic using evolutionary search algorithm," *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, vol. 14, no. 5, pp. 1381–1389, 2021.
[25] P. Velan, M. Čermák, P. Čeleda and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," *International Journal of Network Management*, vol. 25, no. 5, pp. 355–374, 2015.
[26] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features," in *Proc. of the 2nd Int. Conf. on Information Systems Security and Privacy (ICISSP 2016)*, Rome, Italy, pp. 407–414, 2016.
[27] S. Bagui, X. Fang, E. Kalaimannan, S. C. Bagui and J. Sheehan, "Comparison of machine-learning algorithms for classification of VPN network traffic flow using time-related features," *Journal of Cyber Security Technology*, vol. 1, no. 2, pp. 108–126, 2017.
[28] S. Leroux, S. Bohez, P. -J. Maenhaut, N. Meheus, P. Simoens *et al.,* "Fingerprinting encrypted network traffic types using machine learning," in *Proc. NOMS 2018, The IEEE/IFIP Network Operations and Management Symp.*, IEEE, Taipei, Taiwan, pp. 1–5, 2018.
[29] Y. Zion, "Classification and enrichment of encrypted traffic using machine learning algorithms," Master dissertation, Ariel University, Israel, 2018.
[30] S. Miller, K. Curran and T. Lunney, "Detection of virtual private network traffic using machine learning," *International Journal of Wireless Networks and Broadband Technologies (IJWNBT)*, vol. 9, no. 2, pp. 60–80, 2020.
[31] F. Pacheco, E. Expósito and M. Gineste, "A framework to classify heterogeneous internet traffic with machine learning and deep learning techniques for satellite communications," *Computer Networks*, vol. 173, p. 107213, 2020.
[32] L. Guo, Q. Wu, S. Liu, M. Duan, H. Li *et al.,* "Deep learning-based real-time VPN encrypted traffic identification methods," *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 103–114, 2020.
[33] M. Camelo, P. Soto and S. Latre, "A general approach for traffic classification in wireless networks using deep learning," *IEEE Transactions on Network and Service Management*, p. 1, 2021. DOI 10.1109/TNSM.2021.3130382.
[34] S. A. Aswad and E. Sonuç, "Classification of VPN network traffic flow using time related features on apache spark," in *Proc. of the 4th Int. Symp. on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Istanbul, Turkey, pp. 1–8, 2020.
[35] B. Lu, N. Luktarhan, C. Ding and W. Zhang, "ICLSTM: Encrypted traffic service identification based on inception-LSTM neural network," *Symmetry*, vol. 13, no. 6, p. 1080, 2021.
[36] I. H. Witten and E. F. -D. M. Machine, "*Data Mining: Learning Tools and Techniques*," Burlington, Massachusetts, USA: Morgan Kaufmann Publishers, 500, 2005.
[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion *et al.,* "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[38] F. Fuentes and D. C. Kar, "Ethereal *vs.* Tcpdump: A comparative study on packet sniffing tools for educational purpose," *Journal of Computing Sciences in Colleges*, vol. 20, no. 4, pp. 169–176, 2005.
[39] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[40] E. Alawneh, M. Al-Fawa'reh, M. T. Jafar and M. Al Fayoumi, "Sentiment analysis-based sexual harassment detection using machine learning techniques," in *Proc. ISESD*, IEEE, Bandung, Indonesia, pp. 1–6, 2021.
[41] M. Chen, X. Wang, M. He, L. Jin, K. Javeed *et al.,* "A network traffic classification model based on metric learning," *Computers, Materials & Continua*, vol. 64, no. 2, pp. 941–959, 2020.
[42] T. C. Obasi, "Encrypted network traffic classification using ensemble learning techniques," Ph.D. dissertation, Carleton University, Ottawa, Ontario, Canada, 2020.