Research Article

Effective Packet Number for 5G IM WeChat Application at Early Stage Traffic Classification

Muhammad Shafiq and Xiangzhan Yu

School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Xiangzhan Yu; yuxiangzhan@hit.edu.cn

Received 28 October 2016; Revised 29 December 2016; Accepted 15 January 2017; Published 14 February 2017

Academic Editor: Michal Choras

Copyright © 2017 Muhammad Shafiq and Xiangzhan Yu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Accurate network traffic classification at early stage is very important for 5G network applications. During the last few years, researchers endeavored hard to propose effective machine learning model for classification of Internet traffic applications at early stage with few packets. Nevertheless, this essential problem still needs to be studied profoundly to find out effective packet number as well as effective machine learning (ML) model. In this paper, we tried to solve the above-mentioned problem. For this purpose, five Internet traffic datasets are utilized. Initially, we extract packet size of 20 packets and then mutual information analysis is carried out to find out the mutual information of each packet on n flow type. Thereafter, we execute 10 well-known machine learning algorithms using crossover classification method. Two statistical analysis tests, Friedman and Wilcoxon pairwise tests, are applied for the experimental results. Moreover, we also apply the statistical tests for classifiers to find out effective ML classifier. Our experimental results show that 13–19 packets are the effective packet numbers for 5G IM WeChat application at early stage network traffic classification. We also find out effective ML classifier, where Random Forest ML classifier is effective classifier at early stage Internet traffic classification.

1. Introduction

During the last few years, early stage Internet traffic classification received a lot of importance in the area of network traffic classification, from the perspective of features extraction technique, mostly researcher's proposed machine learning models, which were based on features extraction on a whole network flow in [1–3]. In 2005, Moore et al. in [4] presented a feature extraction method which is followed by many researchers for features extraction for their research. They used the whole flow traffic and extracted 248 statistical features, such as the packet sizes and maximum, minimum, and average statistical feature values. Machine learning classifiers can get very effective performance results using these statistical features [5]. These features extraction methods also showed high performance results in the identification of anomaly detection [6]. However, these feature extraction methods are not very effective in reality. Thus it is very important to classify Internet traffic at early stage keeping in view of the security policies and quality of service (QoS)

management. In 2012 [7], Internet traffic classification with few packets has become a very hot topic in the area of network traffic classification. Thus for the problem of accuracy at their early stage Internet traffic classification Qu et al. in 2012 [8] studied that it is possible to classify Internet traffic at their early stage with effective accuracy performance.

However, no study has been carried out on 5G WeChat application at its early stage traffic identification. It is very important problem to find out how many packets are most effective for 5G WeChat application at its early stage traffic classification? As we know that there are very few studies which are related to this problem at early stage traffic, but this is the first study which is concerned with instant messaging (IM) WeChat traffic classification. In this study, we are interested to find out the most effective number of packets as well as effective machine learning classifier for WeChat traffic classification at its early stage using empirical study and information analysis. Five traffic datasets and ten well-known and widely applied ML classifiers are applied for our study. For this study, we use packet size as a feature for our study

and then we use mutual information analysis to find out the mutual relationship and identification information of the first n packets. Thereafter, n numbers of datasets are created for identification. Then, all the selected machine learning classifiers are conducted on generated datasets using different number of packets. At the end, two different statistical analyses are executed on the experimental results to find out the most effective packet number and ML classifier.

The rest of the paper is organized as follows: Section 2 demonstrates related work. The datasets used in this study are discussed in Section 3 and Section 4 includes methodology framework and detailed steps used in this study work. Then the introductory information of mutual information analysis, selected machine learning classifiers, and statistical test theory information are discussed with details in Section 5. Results and analysis are depicted in Section 6, while we have also some discussion discussed in Section 7. In the end, conclusion is presented in Section 8.

2. Related Work

Recently, some studies have been proposed to classify Internet traffic at its early stage with few packets [7], but it is very hard to classify Internet traffic with few packets at its early stage traffic. The main problem in early stage Internet traffic classification is the extraction of effective features. Bernaille et al. in 2006 [9] proposed an early stage Internet traffic classification technique using the size of few early packets of TCP flow as features and executing K -means clustering technique utilizing 10 types of application traffic; they got very effective classification results. Huang et al. in 2008 [10] studied the characteristics of early stage Internet application traffic classification. They used these characteristics for early stage Internet traffic classification. Moreover, in 2013 the authors in [11] extracted features of early stage traffic applications. Using machine learning classifiers, they used packet size, interpacket time, average and standard deviation values, packet size, and interpacket time for early stage Internet application traffic classification. Using these features, they got very high performance results for early stage Internet traffic classification. Este et al. in 2009 [12] studied the features of few packets of early stage traffic and found that these features, packet size, packet interarrival time, and packet direction of early traffic, carry enough information. They also found that these features are most effective features for early stage Internet traffic classification. Hullar et al. present an automatic machine learning (ML) method for P2P traffic classification at early stage, which consumes limited computational and memory resources for early stage traffic identification of P2P traffic. Rizzi et al. in 2013 [13] present a very effective neuro fuzzy system to identify early stage traffic. Nguyen et al. [14] further extend the early stage to “timely” for VoIP traffic classification. They derived statistical features from the subflow, while this means that subflow is a small number of packets.

In [11], the authors used 20 packets and extract feature at early stage. In [9] the authors say for early stage Internet traffic classification five packets are enough to accurately classify early stage traffic. Dainotti et al. [15] used the first 10 packets

packet size (PS) and interpacket time (IPT) for their study work; they also use the average and standard deviation values of packets size and interpacket time of the early stage traffic for early stage traffic identification. Peng et al. in 2015 [16] used payload size of first 10 packets for early stage Internet traffic classification. They say 5–7 packets are most effective packets for early stage Internet traffic identification. They also say that selecting too many packets will increase the computational complexity while selection of few packets for early stage traffic identification will decrease accuracies performance results and cannot possess enough information.

Bernaille et al. in 2006 [17] studied the problem of effective packets numbers for early stage Internet traffic classification. In their study, they used K -means, GMM, and HMM model using the size and direction of the early 10 packets of TCP connection for early stage traffic identification. They say that first four packets of early stage traffic are the most effective packets for early stage Internet traffic classification. They conducted many experiments using different eight traces traffic for their study work and they got very high identification results using the first four packets and executed three machine learning algorithms. Lim et al. in 2010 [18] used not only packet size as features but also connection level and statistical level feature for their study using a number of different datasets while conducting Naïve Bayes, C4.5 decision tree K -nearest neighbors, and Support Vector Machine for their experimental study work. They used first 10 packets for their study to identify UDP application flow and also TCP flow, but their study related to empirical study.

During the last few years, Internet user increases day by day due to presence of reliable and free of cost instant messaging and free calling applications on Internet. WeChat application is one of the instant messaging applications available online freely. WeChat is an instant messaging and free calling application developed by Tencent Holding in China. This is a multifunctionality application and can be used both in smart phone and in desktop machine. After launching the WeChat application, its online users reached 300 million [19] which was amazing traffic and thereafter in November 2015, its active costumer users reached 650 million all over the world while from outside of China its active users reached 100 million [20]. So day to day increasing active users and traffic of this application can affect performance of network. It is also important to classify WeChat messages, audio, and video call traffic accurately to manage quality of services (QoS). Huang et al. [21] proposed measurement ChatDissect tool to measure WeChat application traffic and distinguish 150 K users and 16 GB traffic of WeChat from real-world network traces. In 2013, Church and De Oliveira [22] studied the performance of mobile instant messaging sending service with traditional short messages. In 2014 O'Hara et al. [23] studied instant messaging application WhatsApp in smart phone and took some interviews and survey to study the user activity using WhatsApp application. In 2014, Fiadino et al. [24] also studied WhatsApp application flow stream and collected data in European Network which consisted of millions of data stream flows and also studied audio and video flow data stream. In 2014 Liu and Guo [25] studied video messaging services in WeChat and WhatsApp application and they

TABLE 1: Characteristics of HIT Trace I selected traffic trace.

Task	Application	Duration time	#Packets	Date
Text messages	TCP UDP SSL	1 hour	19967	23 May 2016
Picture messages	TCP UDP SSL	1 hour	91165	23 May 2016
Audio call	TCP UDP SSL	1 hour	364000	23 May 2016
Video call	TCP UDP SSL	1 hour	531830	23 May 2016

captured the traffic using mobile devices for their study. However, no study found out the number of packets that are most effective for WeChat application at its early stage traffic classification. In our previous work in [26], we only classify WeChat text messages service flow using two different environment datasets applying well-known four ML classifiers. In this paper we use only 50 features to classify text message flow and got high accuracy results.

3. Datasets

In this research work, we use five datasets, which include HIT Trace I and HIT Trace II; the details are given below.

3.1. HIT Trace I Dataset. Developing HIT Trace I dataset, we capture WeChat four functionalities such as text messages, pictures messages, audio calls, and video calls traffic and consider these traces traffic as different types of four datasets separately. For this study work, we are interested in classification and finding out the effective number of packets for WeChat real time application at early stage traffic classification. Thus we firstly capture WeChat application traffic of text messages, pictures messages, audio call, and video call traffic using Wire Shark tool [27] for a duration of 1 hour at research lab of School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China, in 23 May 2016. But we select the traffic of nonzero payload packets. In this process of capturing, we are interested to capture WeChat TCP, UDP, and SSL traffic of text messages, pictures messages, audio call, and video call. After capturing the traffic, the trace file is saved as dot PCAP extinction. The characteristics of these datasets are given in Table 1, but note that all the captured applications include TCP, UDP, and SSL application traffic instances. However, this trace traffic includes four subdatasets.

3.2. HIT Trace II Dataset. The second dataset is collected in a dormitory of Harbin Institute of Technology using Wire Shark software. We capture WeChat TCP and UDP traffic and four more other applications' traffic such as DNS, FTP, Telnet,

TABLE 2: Characteristics of HIT Trace II selected traffic trace.

Application	Duration time	#Packets	Date
WTCP	1 hour	20512	28 Apr 2016
WUDP	1 hour	16400	28 Apr 2016
DNS	1 hour	1501	26 Dec 2015
FTP	1 hour	7911	27 Dec 2016
TELNET	1 hour	15832	27 Dec 2015
WWW	1 hour	25251	28 Dec 2015

and WWW applications with duration of one hour. For our study we select traffic that is nonzero payload packets. We capture DNS application traffic on 26 December 2015 with duration of 1 hour, similarly FTP, Telnet on 27 December 2015, and WWW traffic on 28 December 2015. After capturing the traffic, we save the trace traffic as a dot PCAP extinction. The detailed characteristics of this dataset are given in Table 2. Note that WTCP mean WeChat TCP traffic only and WUDP mean WeChat UDP traffic only.

4. Study Framework

Our study framework includes two models; we propose first model for the problem of effective packet numbers and second model for the problem of effective machine learning (ML) classifier.

We carry out the study work as Figure 1 depicts the effective packet number. The detailed explanation of the executed method and steps are given below step by step.

(i) Trace Traffic. In this first step, we capture WeChat traffic using Wire Shark tool and save the captured traffic as dot PCAP extinction. Notice that we capture text messages, picture messages, audio call, and video call traffic duration of one hour, respectively. Thereafter, we select early 20 nonzero payload packets of every application and save them as features extraction.

Figure 2 depicts the effective ML classifier. The detailed explanation process is given below starting from generate features dataset step up to Wilcoxon test results.

FIGURE 1: Study Framework 1 for effective packets number.

FIGURE 2: Study Framework 2 for selection effective ML classifier.

(i) *Features Extraction.* As discussed in related work section that in the area of early traffic classification packet size feature is the most essential packet features for early stage Internet traffic classification [28]. In this study work, we used the packet size as a feature of early stage WeChat traffic. We used the packet sizes of 20 packets of early stage of WeChat application. It is noticed that we only used those packets which have nonzero payload packet [16]. For the features, we put the order of feature in a number vice manner such as packet 1, then packet 2, and then up to 20 packets and similarly packet size 1, packet size 2, and up to 20 packet sizes.

(ii) *Mutual Information.* After development of 20 features datasets, we execute mutual information analysis on between the packet sizes of the features sets of first n packets which we label, respectively, from 1 to 20 packets features to find out the mutual information of each feature's datasets. Through mutual information analysis execution, we are able to know how much identification information carries each packet and we are able to know the effectiveness of each feature's packet.

(iii) *Generate Feature Datasets.* In this section, we make features datasets such as first packet with first packet size in integers and then second packet with its packets size and so third and up to 20 features datasets.

(iv) *Selected Classifiers for Identification.* For this research study, we select 10 machine learning classifiers which are widely used in network traffic classification and are well-known machine learning algorithms. Using the selected machine learning classifiers, we execute crossover identification on the generated datasets. In this paper, we are only interested to find out the effective packet number not to find out the applied machine learning algorithm accuracy. Thus we are only interested in the results of packet numbers.

(v) *Friedman Test.* In this research work, we also use statistical tests to deeply know the effective packets numbers. Friedman tests are to be executed to find out the significant difference among the results of applied number of packets. The detailed study of statistical tests will be demonstrated in Section 5.

(vi) *Wilcoxon Test.* It is also statistical test. We will use this test as we will use Friedman test. In this research study we will first use the Friedman statistical test and then will use Wilcoxon test to find out the effective number of packets using different number of packets.

5. Methodology

In this section, we will explain all the applied methods in this paper study.

5.1. *Mutual Information.* In the information theory, mutual information is very widely used for features selections [29], image processing [30], speech recognition [31], and so on. Mutual information is the measure of mutual dependence between two random variables X and Y which defines the amount of information held by random variable. The mutual

FIGURE 3: The relationship between mutual information and the entropies.

information between two variables in information theory can be defined as

$$\begin{aligned} I(X; Y) &= H(X) - H(X|Y) = H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X, Y) \\ &= H(X, Y) - H(X|Y) - H(Y|X). \end{aligned} \quad (1)$$

In the above equation (1), the marginal entropies of X and Y are $H(X)$ and $H(Y)$, respectively. Where conditional entropies are $H(X|Y)$ and $H(Y|X)$ and joint entropy of X and Y is $H(X, Y)$, respectively, while the connection among $H(X)$, $H(Y)$, $H(X|Y)$, $H(Y|X)$, $H(X, Y)$, and $I(X; Y)$ is shown in Figure 3 according to Shannon's definition of entropy theory, we have

$$H(X) = - \sum_{x \in X} p(x) \log(p(x)), \quad (2)$$

$$H(Y) = - \sum_{y \in Y} p(y) \log(p(y)), \quad (3)$$

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log(p(x, y)), \quad (4)$$

where $p(\cdot)$ is the probability distribution function of a random variable. As in [32] we use the three equations in (1) and can obtain the computational formula of mutual information. We also used the same method for mutual information as in [32].

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log\left(\frac{p(x, y)}{p(x)p(y)}\right). \quad (5)$$

However, if the variables are continuous random variable then summation will be replaced by a definite double integral as given.

$$H(X, Y) = \int_x \int_y p(x, y) \log\left(\frac{p(x, y)}{p(x)p(y)}\right) dx dy. \quad (6)$$

For mutual information computation analysis, there is abundant free available software on Internet, but we choose for our study Peng's mutual information MATLAB toolbox [33].

TABLE 3: Classifiers selected in this study.

Classifiers	Type
BayeNet	Bayes
Naïve Bayes	Bayes
SMO	Functions
AdaBoost	Meta
Bagging	Meta
OneR	Rule
Part	Rule
NBTree	Trees
J48	Trees
Random Forest	Trees

5.2. Machine Learning Classifiers. We conducted our identification experiments using ten well-known and widely used machine learning classifiers. All the selected machine learning classifiers are executed using Weka data mining software [34]. Weka tool is a data mining application used in many areas in computer science and also used by many researchers for network traffic classification [35]. Firstly, we formatted the datasets as a comma separated value "CSV" which is a supported extinction of Weka application. Then using two folders' cross validation method, we apply all the selected machine learning classifiers. The introductory information about the applied machine learning classifiers is given below and the classifiers selected for this study are shown in Table 3.

- (i) *Bayes*: Bayes machine learning classifiers are actually based on Bayes Theorem; Bayes machine learning classifiers are very widely used in computer and engineering area and got very effective results. In this study, we utilized Bayesian network (Bayes Net) [36] machine learning classifier and also Naïve Bayes machine learning classifiers [37, 38].
- (ii) *Meta*: in this research work, we used Meta category classifier in Weka tool; we select Bagging [39] and AdaBoost [40] machine learning classifier to classify WeChat traffic accurately. Meta classifier was first trained to learn and then produce strong learning.
- (iii) *Rule*: this category algorithm just creates rules using specific policy and then executes classification result testing data. In this category, we select OneR [41] and PART [42] rule base classifier for our study.
- (iv) *Trees*: this is also called decision tree algorithms used by many researchers in their research study. It is also called statistical classifiers. In our study, we select J48 also called C4.5 classifiers [43], Naïve Bayesian trees [44], and Random Forest [45].
- (v) *SMO*: in function category, we select SOM [46, 47]. SOM is also called a supervised machine learning technique and known as Support Vector Machine, which is widely used in many areas for classification. SVM is useful for both classification and regression.

For more detailed literature review of the applied machine learning classifiers, the selected machine learning classifiers are cited with original literature review.

5.3. Statistical Test. In more depth to know the effectiveness of packets and to compare the results of the applied ML classifiers as well as to find out the significant difference among the results of the applied method, statistical tests are conducted. In this study, we executed two different statistical tests, Friedman and Wilcoxon test, on the results of methods [48, 49]. The detailed introductions to both Friedman and Wilcoxon statistical tests are given below.

(i) *Friedman Test.* Friedman test is a statistical test. It is also called Friedman nonparametric test. Friedman test is a kind of nonparametric test used to find out the significant differences between the results of applied methods. The first step in this test is to calculate the test statistics; it converts all the original results of methods to ranks. The process of ranking of this test is that it ranks the best performing on the rank of 1 and then second best 2 and so on. After ranking the average ranking (AR), it is then calculated. If r_i^j is the rank of the n th of K algorithms in i th of n datasets, thus the Friedman test also needs average ranking of algorithms:

$$R_j = \frac{1}{n} \sum_i r_i^j. \quad (7)$$

While in null hypothesis all the algorithms behave similarly, ranks should be equal and the distribution x^2 is calculated as follows:

$$x_F^2 = \frac{12m}{n(n+1)} \left[\sum_{j=1}^n AR_j - \frac{n(n+1)^2}{4} \right], \quad (8)$$

where m is the number of executions and n is the number of methods. If the distribution of methods x_F^2 is large enough thus there is value of significant difference among the applied method results and it will be rejected. The significant level for each method x_F^2 and the probability value (p value) shows the significant level and it is usually conducted for the analysis of test results. On the other hand, for multiple hypothesis testing we also apply post hoc method to determine hypothesis comparison that will be rejected at specified significance level, while in many cases lowest hypothesis result is also concerned about rejection. Lowest significance results also called adjusted p value (APV) and post hoc can be used to search the lowest p value for each hypothesis. In this study, we used for post hoc method Holm's test [48] which is very effective test for producing significant test results. For this research work, firstly we used Friedman test using $1 \times n$ comparison because $n \times n$ comparison is too long to show in this paper.

(ii) *Wilcoxon Test.* We also used Wilcoxon signed rank statistical test in this research study. Wilcoxon test is also a nonparametric test used for pairwise comparison between two methods [50]. If d_i is the difference between two methods performance scores on i th out of n problem and if the score

Actual	Positive	TP	FN
	Negative	FP	TN

TP: # of positive instances correctly classified
 TN: # of negative instances correctly classified
 FP: # of negative instances incorrectly classified
 FN: # of positive instances incorrectly classified

FIGURE 4: Confusion matrix for classification results evaluation.

is known in different ranges, then it can be normalized on intervals 0 and 1 in [51]. Thereafter, the difference is ranks by their absolute values while in case of ties practitioners will apply one method in [52]. In this case, the positive values considered that the method performed well and the second one vice versa.

$$R^+ = \sum_{d_i > 0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{rank}(d_i), \quad (9)$$

$$R^- = \sum_{d_i < 0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{rank}(d_i).$$

R^+ is the sum of positive values and R^- is the sum of negative difference values. It means that if the difference between these R^- and R^+ is high then the hypothesis will be rejected. This statistical test is also used like Friedman test to determine that the hypothesis will be rejected or not on the specific significance values α .

5.4. Evaluation Criteria for Performance Measurement. For the performance measurements confusion matrix is the base of traffic classification measurements. Figure 4 shows the confusion matrix for traffic classification performance evaluation, wherein rows refer to the actual class of the instances and column refers to the predicted class of instances.

The metrics that are used in this Internet traffic classification using confusion metrics are described below step by step:

- (i) *True Positive (TP):* it means that Class A is truly identified as belonging to Class A.
- (ii) *True Negative (TN):* it means that Class A is truly identified as not belonging to Class A.
- (iii) *False Positive (FP):* it means that Class A is not truly identified as belonging to Class A.
- (iv) *False Negative (FN):* it means that Class A is not truly identified as not belonging to Class A.

Using the above given metrics, different metrics can be made for the evaluation of classification performance [53, 54], but

note that effective classifiers will minimize the FP and FN values. However, in this regard we used accuracy and AUC measurement metrics defined as follows.

(i) *Accuracy*. Classification accuracy can be defined as the truly classified samples in overall classified samples and its formula is given below. Mathematically, accuracy can be defined as the sum of TP and TN divided by the sum of TN, TN, FP, and FN.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (10)$$

A classifier performance is measured by accuracy result. It shows the overall effectiveness of classification model.

(ii) *Sensitivity*. Remember that sensitivity and recall are the same metrics in traffic classification technique. So (4) can be used for sensitivity.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (11)$$

(iii) *Specificity*. It can be defined as the performance ability of machine learning classifier to classify negative results. Equation (5) shows its formula while mathematically it can be defined as TN divided by sum of FP and TN.

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} = 1 - \text{FPR}. \quad (12)$$

(iv) *Area under Curve*. It is also called receiver operating characteristics (ROC) curve [55], which defines the performance of machine learning classifiers. It also shows the trade-off among FPR and TPR, while FPR is also known as specificity and TPR is called sensitivity. The AUC values can be computed by using confusion matrix values by TPR and FPR.

$$\text{AUC} = \frac{1 + \text{TPR} - \text{FPR}}{2} \quad (13)$$

since specificity = 1 – FPR and sensitivity = TPR.

Replacing 1 – FPR by specificity and TPR by sensitivity, we will get

$$\text{AUC} = \frac{\text{Sensitivity} + \text{Specificity}}{2}. \quad (14)$$

Equation (14) shows that AUC is the average of sensitivity and specificity.

6. Experimental Results and Analysis

In this section, we will explain the detailed experimental results and analysis. Firstly, we will explain the mutual information analysis results of HIT Trace I dataset including four subdatasets and then HIT Trace II dataset results, then give the result analysis of applied methods to validate the effectiveness of packets, and lastly give the results of statistical test for effective ML classifier.

FIGURE 5: Mutual information results of HIT Trace I dataset.

6.1. Mutual Information Analysis Results

6.1.1. Mutual Information Results of the HIT Trace I Dataset.

Figure 5 shows the mutual information method analysis results. In Figure 3 the mutual information of the first two packets of text messages and picture messages is higher compared to the mutual information analysis results of audio call and video call packets. The audio call and video call traffic results are no more than 0.1 values. It means that the first two packets are not contributing information, while in text and picture message packet contributes highly compared with 2 to 4 packets. However, in text messages traffic packet numbers 8-9 give high information identification values and in picture messages packets numbers 7-8 give high information identification values while in audio call traffic packets numbers 6-7 give high information identification values and in video call traffic type is very different as compared to other traffic data; in this traffic packet numbers 19-20 give very high identification information. More details of mutual information results are shown in Table 16.

6.1.2. Mutual Information Results of the HIT Trace II Dataset.

Figure 6 shows the mutual information method analysis results with details. In Figure 6 the mutual information of the first two packets of FTP, DNS, and WWW application is higher as compared to the other WTCP, WUDP, and Telnet application. Similarly packets 2-3 are also not contributing very effectively. Its means that packets 1-3 do not give much identification information. However, packets 6 and 17 give very effective identification information and remaining packets are not contributing very well as compared to the other packets. Moreover, with the perspective application FTP and DNS give very effective identification information compared with other applications.

6.2. Analysis Results of HIT Trace I Dataset

6.2.1. *Results of the Text Messages Traffic Dataset*. Figure 7 shows the accuracy results of the WeChat Text message

FIGURE 6: Mutual information results of HIT Trace II dataset.

FIGURE 7: Accuracy results of WeChat text message dataset.

dataset while the details results are shown in Table 19. All the applied machine learning classifiers get very low accuracy using first two and three packets of early traffic, because it is very difficult to identify Internet traffic with only few packets. Due to this reason, all the applied machine learning classifiers get very low accuracy results using early few packets. Naïve Bayes, Hoeffding, and Random Forest get very low accuracy result using early two packets. However, using text messages dataset, we could not conclude that the first packets are more effective for early stage Internet traffic classification. It is worse to say that the first three packets for early stage Internet traffic classification are effective. However, after three packets using first four packets of WeChat text messages dataset all the applied machine learning classifiers get very effective accuracy results except Random Forest and Part ML classifiers. These three ML classifiers get low accuracy results using first four packets. Support Vector Machine (SVM) gets

FIGURE 8: AUC result of WeChat text message dataset.

low accuracy result compared to other machine learning classifiers. However, all the classifiers give continuously incensements in accuracy results using all the packets numbers. Except Random Forest classifier, this classifier shows poor accuracy results which are not stable. Thus we can infer that the first four and five packets carry enough identification information for early stage classification WeChat text messages dataset. Note that we use two folders' cross validation method in this study work.

Figure 8 shows the AUC result of WeChat text messages dataset and the details are shown in Table 23. From the figure, it is clear that the first two three packets cannot contribute to the AUC identification using the selected machine learning classifiers and the AUC result continuously increases using the selected machine learning classifiers conducting the first four packets to twenty packets. Thus the entire conducted machine learning classifiers give very effective AUC results but some of them such as SMO and OneR ML classifiers give noneffective results for the WeChat text messages dataset. As discussed in accuracy analysis, the Naïve Bayes, Hoeffding, and Random Forest give low accuracy results, while in AUC result, Naïve Bayes, Hoeffding, and Random Forest give effective AUC results compared to accuracy of WeChat Text dataset. It means that there exists imbalance data in WeChat Text message dataset.

Table 4 shows Friedman's statistical test results for accuracy result. In Friedman' test result the packet number nineteen is the best performed one in the accuracy results with the lowest ranking values being 1.7555. While comparing the p values and adjusted p values, the packet numbers 10–12 of adjusted p values are less than p values and numbers 15–17 are also the same as adjusted p value which is less than p values. These are the best performance packets.

For better understanding the results of Friedman's test, we also execute Wilcoxon sign rank test. Table 5 shows Wilcoxon pairwise test results for the WeChat text messages dataset. From the table the p value of 20 packets is greater than 0.05 for the accuracy results. Thus we conclude that there is no

TABLE 4: Friedman's test results for WeChat text messages dataset.

Number of packets	Accuracy		
	Ranking	p value	APV
2	15.7555	2.651213	4.241941
3	18.6555	1.875779	3.376403
4	16.5555	4.599337	7.818873
5	14.2222	7.531631	1.129744
6	11.6666	8.339059	1.167468
7	11.5555	9.078222	0.001089
8	11.5555	9.855377	0.001089
9	13.9999	1.379661	1.793560
10	09.0000	0.003965	0.031727
11	09.0000	0.001812	0.016316
12	07.1499	0.031893	0.159467
13	10.6666	4.370569	0.004370
14	05.8888	0.107549	0.322647
15	08.7999	0.005088	0.035618
16	07.6999	0.018064	0.108387
17	06.6000	0.053955	0.215821
18	03.3999	0.512053	1.024107
19	01.7555	—	—
20	03.3333	0.537955	1.024107
Friedman's test	—	0.0 00000	

TABLE 5: Wilcoxon pairwise test results for WeChat text messages dataset.

19 pkts versus n pkts	Accuracy		
	R^+	R^-	p value
3	1.000	27.00	0.027
4	22.00	14.00	0.570
5	47.00	7.50	0.041
6	36.00	0.00	0.011
7	50.50	4.50	0.018
8	44.00	1.00	0.011
9	47.50	7.50	0.041
10	44.00	1.00	0.011
11	54.00	1.00	0.007
12	54.00	1.00	0.007
13	50.00	5.00	0.022
14	54.00	1.00	0.007
15	54.00	1.00	0.007
16	54.00	2.50	0.007
17	52.00	1.00	0.110
18	54.00	1.00	0.007
19	54.00	1.00	0.007
20	54.00	1.00	0.007

significant difference existing between the results of 19 packets and other packets for the WeChat text messages dataset.

6.2.2. Results of the Picture Messages Traffic Dataset. Figure 9 shows the accuracy results of the WeChat Pictures Messages

FIGURE 9: Accuracy results of WeChat pictures message dataset.

dataset and the details results are shown in Table 18. The results of the text messages dataset are different from the pictures messages dataset. The packet number three gives very low significant increase of accuracy results compared with the first two packets.

From the results, it is concluded that the packet number three does not give identification results for the accuracy. It is also observed in the result that all the applied classifiers got continuously increment results using all the number of packets except Support Vector Machine (SMO) classifiers, continuously giving random results using all numbers of packets, while OneR classifiers give very poor identification result in the first 12 packets and then their results are continuously increasing. It means that there exist imbalance data in the dataset. The AUC results for the WeChat pictures messages dataset are a little bit similar to accuracy results. In Figure 10 and Table 22 the AUC results are shown, in which all the machine learning classifiers get the same AUC results but only SMO and OneR ML classifiers results are different from the other ML classifiers which hit high AUC results values.

Table 6 shows Friedman's test results for the accuracy result of WeChat pictures messages datasets. In the table packet number 18 gives the best performance result for the accuracy. The average ranking result of 18 is 04.3333 values for the accuracy, which are the lowest average ranking results. However observing the p value and adjusted PV for the accuracy, the packets number six to seventeen p values are less than when compared to adjusted p values. Thus these are the best behaving packets number for accuracy result, while the packet number five and packet number nine p values are greater than adjusted p values. Thus there is no significant difference among the results with accuracy.

Table 7 shows the Wilcoxon test results for the accuracy results. In the table, it is clear that the packet numbers 13–15 and 20 packets p values are greater than the standard level of 0.05 for the accuracy results. Thus the packet numbers 13–15

FIGURE 10: AUC results of WeChat pictures message dataset.

TABLE 6: Friedman's test results for WeChat pictures messages dataset.

Number of packets	Accuracy		
	Ranking	p value	APV
2	17.7777	1.01164	1.71979
3	18.5555	1.67590	3.01663
4	14.2222	8.35921	0.00133
5	12.6666	9.73464	0.01362
6	10.4999	0.01375	0.15129
7	09.3555	0.04478	0.38802
8	8.35555	0.10754	0.57081
9	12.7555	7.85968	0.01178
10	11.3555	0.00508	0.06614
11	10.7000	0.01098	0.13184
12	09.5555	0.03880	0.38802
13	07.0000	0.28332	1.13331
14	05.0500	0.76568	1.53137
15	07.0000	0.28332	1.13331
16	08.5555	0.09513	0.57081
17	09.4444	0.04271	0.38802
18	04.3333	—	—
19	08.8555	0.07060	0.49426
20	4.44444	0.96830	1.53137
Friedman's test	—	0.00000	

and 20 are not significantly different for the WeChat pictures messages dataset.

6.2.3. Results of the Audio Call Traffic Dataset. Figure 11 shows the accuracy results of all the selected machine learning classifiers for the WeChat Audio Call dataset. Comparing the results of previous datasets, the results of WeChat audio call dataset are very complex as shown in Figure 11. It is also clear from the figure that the first three packets do not gain

TABLE 7: Wilcoxon pairwise test results for WeChat pictures messages dataset.

18 pkts versus n pkts	Accuracy		
	R^+	R^-	p value
2	10.50	17.00	0.527
3	28.00	0.00	0.017
4	43.50	1.50	0.013
5	36.00	0.00	0.012
6	54.00	1.00	0.007
7	45.00	1.00	0.008
8	54.00	0.00	0.007
9	45.00	1.00	0.008
10	45.00	0.00	0.008
11	55.00	0.00	0.005
12	55.00	0.00	0.005
13	55.00	0.00	0.005
14	55.00	0.00	0.005
15	55.00	0.00	0.005
16	55.00	0.00	0.005
17	55.00	0.00	0.005
19	55.00	0.00	0.005
20	55.00	0.00	0.005

FIGURE 11: Accuracy results of WeChat audio call dataset.

identification performance effectively, while packet number four gets effective identification performances for accuracy results. Again the SMO machine learning classifiers give random performance results, which are not stable results, while OneR machine learning classifiers give stable performance results after 11 packets and Bayes Net classifiers give effective result using packet number nine while its performance is continuous after 12 packets. The detailed accuracy results are shown in Table 17.

The AUC results for the WeChat Audio Call dataset accuracy are shown in Figure 12 and Table 21. The AUC results

TABLE 8: Friedman's test results for WeChat audio call dataset.

Number of packets	Accuracy		
	Ranking	p value	APV
2	17.0499	2.0498	3.4847
3	18.3555	1.4017	2.5231
4	11.9999	0.0068	0.0964
5	13.8888	5.4616	0.0081
6	10.5555	0.0318	0.3827
7	9.85555	0.0590	0.5909
8	6.99999	0.4744	1.6160
9	7.77777	0.3015	1.5077
10	5.64999	0.8270	1.6160
11	13.9499	4.3705	0.0069
12	11.7000	0.0087	0.1134
13	7.99999	0.2491	1.4950
14	6.20000	0.6620	1.6160
15	8.35555	0.1965	1.3759
16	5.10000	—	—
17	10.0000	0.0515	0.5667
18	9.11111	0.1119	1.0076
19	7.22222	0.4040	1.6160
20	8.77777	0.1525	1.2206
Friedman's test	—	0.0000	

are very simple as compared to the accuracy results. The first three packets gain low AUC result while the first four packets gain effective AUC result for accuracy results. However, SMO and Bayes Net machine learning classifiers give low level AUC result for the first four packets and other machine learning classifiers give accurate AUC results, while Random Forest machine learning classifier gives very accurate AUC results for the accuracy.

Table 8 shows Friedman's test results for the WeChat audio call dataset accuracy. The packet number sixteen gets the lowest average ranking values for accuracy and all the p values of packets are less than from the adjusted p values except packet numbers 5 and 11. Thus we can say that there is no significant difference among the results, while the Wilcoxon test results are shown in Table 9, in which only packet numbers 2–5 and 11–15 get the p value less than 0.05, which mean that these packets are significantly different from the other packets results.

6.2.4. Results of the Video Call Traffic Dataset. Figure 13 shows the accuracy results of WeChat video call datasets. The accuracy results of video call dataset are different from the previous datasets accuracy results. The result of this dataset is a little bit complex; however, all the machine learning classifiers get effective accuracy results using all the packets datasets for accuracy. But the result C4.5 decision classifier is completely straight line conducting all the packets datasets. The results of the first two and three packets are lowest using SMO and Naïve Bayes classifiers but after four packets its accuracy result increases continuously. Similarly, using Bayes Net, Random Forest, Bagging, and OneR classifiers

TABLE 9: Wilcoxon pairwise test results for WeChat audio call dataset.

18 pkts versus n pkts	Accuracy		
	R^+	R^-	p value
3	10.00	18.00	0.495
4	28.00	0.000	0.017
5	48.00	7.000	0.036
6	36.00	0.000	0.012
7	52.00	3.000	0.012
8	45.00	0.000	0.008
9	52.00	3.000	0.012
10	45.00	0.000	0.008
11	45.00	0.000	0.007
12	55.00	0.000	0.005
13	54.00	1.000	0.007
14	55.00	0.000	0.005
15	55.00	0.000	0.005
16	55.00	0.000	0.005
17	53.00	2.000	0.009
18	55.00	0.000	0.005
19	54.00	1.000	0.007
20	55.00	0.000	0.005

FIGURE 12: AUC results of WeChat audio call dataset.

with packet number 17, all the classifiers give lowest results, but the remaining classifiers get high accuracy results using packet number 17. In Figure 14 and Table 24 we have shown the AUC results for the WeChat video call dataset. The AUC result pattern is simple as compared to accuracy result. Most of the applied machine learning classifiers get the effective AUC result for the video call dataset except OneR and SMO machine learning classifiers, because the results of the OneR and SMO are the lowest compared to other machine learning classifiers, while Table 10 shows Friedman's statistical test results for the accuracy of WeChat video call datasets. The packet number two gets the highest average rank values

FIGURE 13: Accuracy results of WeChat video call dataset.

FIGURE 14: AUC results of WeChat video call dataset.

compared to other packets average rank results and its value is 05.20. Similarly, all the p values are less than adjusted p values except packet numbers 4-5. It means that there does not exist significant difference among these results with respect to accuracy.

In Table 11, we have shown the Wilcoxon test results for the WeChat video call dataset accuracy. In Table 11, all the p values are less than the standard level 0.05 except packet number 4. It means that the results of the entire packet except 4 packets are significantly different from all other results.

6.3. Analysis Results of HIT Trace II Dataset. Figure 15 shows the accuracy results of HIT Trace II dataset. All the applied machine learning classifiers get low accuracy result for early stage Internet traffic, because it is very difficult to classify Internet traffic using first few packets. However, we are not interested in classification accuracy. We are interested to

TABLE 10: Friedman's test results for WeChat video call dataset.

Number of packets	Accuracy		
	Ranking	p value	APV
2	05.20	—	—
3	08.25	1.67590	2.5231
4	10.00	8.35921	0.0964
5	11.80	9.73464	0.0081
6	11.55	0.01375	0.3827
7	13.00	0.04478	0.5909
8	12.50	0.10754	1.6160
9	10.10	7.85968	1.5077
10	10.70	0.00508	1.6160
11	11.05	0.01098	0.0069
12	11.45	0.03880	0.1134
13	12.85	0.28332	1.4950
14	15.12	0.76568	1.6160
15	09.10	0.28332	1.3759
16	09.35	0.09513	0.5245
17	05.25	0.04271	0.5667
18	05.70	0.00543	1.0076
19	06.65	0.07060	1.6160
20	06.50	0.96830	1.2206
Friedman's test	—	0.0000	

TABLE 11: Wilcoxon pairwise test results for WeChat video call dataset.

18 pkts versus n pkts	Accuracy		
	R^+	R^-	p value
3	21.00	0.000	0.027
4	10.00	0.000	0.063
5	28.00	0.000	0.018
6	28.00	0.000	0.018
7	36.00	0.000	0.012
8	28.00	0.000	0.018
9	35.00	1.000	0.017
10	42.00	2.500	0.018
11	43.00	2.000	0.015
12	43.00	2.000	0.015
13	43.00	2.000	0.015
15	39.00	6.000	0.035
16	39.00	15.00	0.050
17	29.50	14.00	0.405
18	31.00	14.00	0.314
19	31.00	14.00	0.314
20	31.00	14.00	0.314

find out the most effective packet numbers and effective ML classifier. Moreover, packet numbers 13-14 give the same identification results, but its identification information is low as compared to other packets' accuracy results. However, the accuracy results of packet numbers 12 and 18 are continuously increasing. It means that their accuracy results are very good as compared to other packets' accuracy results. Moreover, all

FIGURE 15: Accuracy results of HIT Trace II dataset.

FIGURE 16: AUC results of HIT Trace II dataset.

classifiers show stable accuracy results, but Random Forest algorithm gives effective results compared to other machine learning classifiers. Thus the first six packets carry enough identification information as well as 15–19.

The AUC results for the HIT Trace II dataset is shown in Figure 16 and Table 25. The AUC result for the HIT Trace II is very simple as compared to other traces AUC result. For example, only packet number 5 and packet number 17 get low AUC values and all the remaining packets gain good AUC results as compared to packet AUC results. Moreover, Bagging classifier gives low AUC for packet number 14 and all the remaining classifiers give high AUC result for packet number 14. Similarly, for packet number 5 all classifiers give good AUC except SMO and AdaBoost machine learning classifiers. However, all the machine learning classifiers give high AUC values results for early stage packets. However the detailed AUC results are shown in Table 26.

TABLE 12: Friedman's test results for HIT Trace II dataset.

No of Packets	Accuracy		
	Ranking	p value	APV
2	11.55	7.31326	0.00877
3	13.99	1.35463	2.03194
4	10.05	0.00541	0.05410
5	09.50	0.01037	0.08302
6	09.50	0.01806	0.12645
7	15.60	6.13671	9.81874
8	11.90	4.37056	0.00611
9	11.65	6.32470	0.00822
10	10.95	0.00169	0.01863
11	09.80	0.00731	0.06583
12	07.35	0.08751	0.35006
13	15.80	4.05545	7.29981
14	15.80	4.05545	7.29981
15	07.99	0.04919	0.29514
16	05.55	0.32051	0.64103
17	08.00	0.049191	0.29514
18	03.05	—	—
19	07.30	0.091261	0.35006
20	05.15	0.40402	0.64103
Friedman's test	—	0.0000	

Table 12 shows Friedman's test results for the HIT Trace II dataset accuracy. The packet number 18 gets the lowest average ranking values for accuracy and all the p values of packets are less than from the adjusted p values except packet numbers 8 and 9. Thus we can say that there is no significant difference among the results, while the Wilcoxon test results are shown in Table 13 in which only packet numbers 16 and 18 get the p values less than 0.05, which mean that these packets are significantly different from the other packets results.

6.4. Analysis Results of Algorithms. Table 14 shows Friedman's test results for the applied machine learning classifiers. The Random Forest machine learning classifier gets the lowest average ranking values as compared to other machine learning classifiers and all the p values are less than from the adjusted p values except Hoeffding, Bayes Net, SMO, and OneR ML classifiers, while the Wilcoxon test results are shown in Table 15 in which only classifiers OneR, Part, C4.5, and Random Forest get the p value less than 0.05, which mean that these classifiers are significantly different from the other packets results.

7. Analysis and Discussion

Although the results of the five applied IM and WeChat traffic datasets are different, with respective accuracy results and AUC results, some information can be learned from the applied five datasets at early stage WeChat traffic classification.

- (i) From this study, it is clear that analyzing the results of information analysis and classification experiments results analysis of the first three packets for early stage IM do not carry enough identification information.
- (ii) From the experimental results, the early traffic packets carry enough identification information for the

TABLE 13: Wilcoxon pairwise test results for HIT Trace II dataset.

18 pkts versus n pkts	Accuracy		
	R^+	R^-	p value
3	24.50	30.50	0.459
4	04.00	16.00	0.776
5	32.00	23.00	0.646
6	29.00	07.00	0.123
7	17.00	38.00	0.284
8	25.00	20.00	0.766
9	26.50	28.50	0.919
10	25.00	20.00	0.766
11	34.50	20.50	0.475
12	38.50	06.50	0.058
13	19.50	35.50	0.414
14	19.50	35.50	0.414
15	39.50	15.50	0.221
16	39.50	05.50	0.043
17	40.00	15.00	0.203
18	40.50	04.50	0.033
19	35.00	10.00	0.139
20	36.50	08.50	0.970

TABLE 14: Friedman's test results for algorithm.

Algorithm	Accuracy		
	Ranking	p value	APV
RForest	1.942	—	—
Part	3.236	0.18280	0.311291
C4.5	3.342	0.15564	0.311291
Naive Bayes	4.947	0.00225	0.008256
Bagging	4.973	0.00206	0.008256
Hoeffding	5.315	6.05545	0.003027
Bayes Net	7.000	2.69412	1.616472
Adaboost	7.815	2.31268	1.618882
SMO	7.973	8.52091	6.816731
OneR	8.447	3.66247	3.296231
Friedman's test	—	0.0000	

WeChat early traffic classification. However, all the applied machine learning classifiers get very high effective identification performance results using the early stage traffic except Support Vector Machine and OneR machine learning classifiers results are very poor compared to other applied ML classifiers.

- (iii) Through accuracy results, the classification performance can be easily evaluated for the early stage Internet traffic classification. But in some cases, some classifiers get high identification performance results and in some cases not very effective, it is due to imbalanced datasets.
- (iv) OneR and SVM classifiers performance is always poor with increase of nonzero payload packet numbers. However, the performance of OneR and SVM classifiers is very different as compared to other machine learning classifiers.

TABLE 15: Wilcoxon pairwise test results for algorithms.

Algorithms	Accuracy		
	R^+	R^-	p value
Naive Bayes	145.00	8.000	0.001
SMO	50.500	102.5	0.218
Adaboost	30.000	48.00	0.480
Bagging	127.00	26.00	0.170
OneR	23.000	148.0	0.006
Part	171.00	0.000	0.000
Hoeffding	121.00	32.00	0.350
C4.5	171.00	0.000	0.000
Rforest	189.00	1.000	0.000

- (v) However, it is clear from the experimental results that Random Forest gives very accurate results for all the applied datasets.

8. Conclusion

In this paper, we have tried to find out the most effective packet numbers for the IM WeChat early stage traffic classification as well as effective machine learning classifier. Using mutual information analysis five datasets (text messages, picture messages, audio call, and video call traffic), HIT Trace II, and ten well-known machine learning classifiers are applied. According to experimental results, we conclude that the nonzero payload size packets carry enough identification information for WeChat instant messages applications traffic classification. However, the packet numbers 13–19 are effective packets for 5G WeChat application traffic identification. Moreover, the experimental results of the five datasets are different due to different functionality of 5G WeChat application. However, in the results all the utilized datasets are not the same and the first three packets do not carry enough identification information and give very poor results, while for WeChat early stage traffic classification, according to our experimental analysis, the packet numbers 13–19 are most effective packet numbers. While for effective ML classifiers, we conclude that Random Forest machine learning classifier is effective ML classifier for IM early stage traffic classification.

There is still gap for further research in the early Internet traffic classification. A new method should be developed to select effective packet numbers for 5G WeChat application early stage traffic identification, while selecting more packets for Internet traffic classification increases computational complexity while minimum features will decrease classification accuracy of machine learning classifier so that more models should be developed that show how many packets should be used for accurate IM application traffic classification.

Appendix

Detailed Results of the Experimental Work

See Tables 16–26.

TABLE 16: Mutual information analysis.

Number of packets	Mutual information			
	Text messages	Picture messages	Audio call	Video call
1	1.267128	1.3519	1.0892737	1.1317
2	1.094605	1.2914	1.4401684	1.4416
3	1.258356	1.1384	1.2647211	1.3305
4	1.267128	1.3431	1.4335891	1.4416
5	1.267128	1.3607	1.4401684	1.3305
6	0.933778	1.3607	1.4518649	1.4416
7	1.103377	1.4318	1.4401684	1.3305
8	1.364587	1.3431	1.4401684	1.4533
9	1.241780	1.1677	1.4401684	1.3305
10	1.103377	1.3607	1.2150123	1.4416
11	1.268531	1.1677	1.4461477	1.3480
12	1.079986	1.3519	1.4401684	1.4689
13	1.065364	1.3607	1.2150123	1.4416
14	1.097529	1.3607	1.2413281	1.4416
15	1.267128	1.3431	1.4138526	1.4416
16	1.267128	1.3607	1.0249456	1.3480
17	1.103377	1.3607	1.2233193	1.4416
18	1.097529	1.3607	1.1882304	1.1375
19	1.077061	1.3607	1.2203953	1.1492
20	1.267128	1.3431	1.2501018	1.4777

TABLE 17: Accuracy results of WeChat audio traffic dataset.

Pkts #	Bayes Net	N/Bayes	SMO	Adaboost	Bagging	OneR	Part	Hoeffbing	C4.5	R/forest	Average
2	33.33	33.33	50.00	50.00	33.33	33.33	33.33	33.33	33.33	50.00	38.331
3	22.22	55.55	33.33	55.55	22.22	22.22	33.33	33.33	33.33	55.55	36.663
4	33.33	66.66	66.66	66.66	33.33	33.33	58.33	66.66	58.33	100.0	58.329
5	26.66	66.66	33.33	66.66	66.66	26.66	53.33	60.00	53.33	100.0	55.329
6	33.33	77.77	72.22	66.66	66.66	33.33	50.00	55.55	50.00	100.0	60.552
7	57.14	76.19	47.61	66.66	66.66	28.57	66.66	80.95	66.66	100.0	65.710
8	79.16	70.83	70.83	66.66	75.00	33.33	91.66	79.16	91.66	100.0	75.829
9	66.66	70.37	37.03	66.66	92.66	29.62	92.59	92.96	92.59	100.0	74.114
10	80.00	63.33	70.00	66.66	76.66	33.33	100.0	80.00	100.0	100.0	76.998
11	51.51	51.51	60.60	60.60	60.60	33.33	84.84	51.51	84.84	93.93	63.327
12	50.00	72.22	61.11	61.11	66.66	61.11	80.55	75.00	80.55	94.44	70.275
13	51.28	84.61	41.02	61.53	94.87	61.53	92.30	76.92	97.43	94.87	75.636
14	73.80	83.33	61.90	61.90	88.09	61.90	92.85	78.57	97.61	95.23	79.518
15	73.33	71.11	51.11	62.22	88.88	64.44	88.88	75.55	95.55	97.77	76.884
16	83.33	77.08	66.66	64.58	83.33	64.58	97.91	79.16	95.83	97.91	81.037
17	80.39	70.58	33.33	58.82	74.50	76.47	92.15	78.43	92.15	92.15	74.897
18	79.62	72.22	68.51	59.25	74.07	77.77	88.88	70.37	88.88	90.74	77.031
19	73.68	71.92	38.59	64.91	89.47	70.17	96.49	78.94	96.49	94.73	77.539
20	65.00	70.00	61.66	68.33	81.66	76.66	86.66	70.00	86.66	95.00	76.163

TABLE 18: Accuracy results of WeChat picture traffic dataset.

Pkts #	Bayes Net	N/Bayes	SMO	Adaboost	Bagging	OneR	Part	Hoeffbing	C4.5	R/forest	Average
2	33.33	33.33	33.33	33.33	33.33	33.33	33.33	66.66	33.33	33.33	36.663
3	22.22	44.44	33.33	44.44	22.22	22.22	33.33	44.44	33.33	44.44	34.441
4	33.33	66.66	58.33	58.33	33.33	33.33	66.66	83.33	66.66	75.00	57.496
5	33.33	73.33	40.00	73.33	53.33	26.66	46.66	86.66	46.66	80.00	55.996
6	33.33	72.22	66.66	61.11	72.22	33.33	77.00	88.88	77.77	88.88	67.144
7	66.66	76.19	42.85	71.42	57.14	28.57	80.95	90.47	80.95	80.95	67.615
8	62.50	79.16	62.50	66.66	66.66	33.33	83.33	91.66	83.33	75.00	70.413
9	62.96	70.37	40.74	62.96	59.25	29.62	77.77	81.48	77.77	81.48	64.444
10	76.66	80.00	50.00	60.00	63.33	33.33	73.33	80.00	80.00	76.66	67.331
11	57.57	78.78	33.33	63.63	69.69	36.36	81.81	81.81	81.81	78.78	66.357
12	80.55	69.44	61.11	66.66	69.44	66.66	80.55	75.00	80.55	80.55	73.051
13	82.05	76.92	41.02	66.66	87.17	66.66	79.48	82.05	79.48	89.74	75.123
14	78.57	76.19	64.28	66.66	78.57	66.66	85.71	83.33	90.47	90.47	78.091
15	77.77	68.88	92.22	66.66	82.22	66.66	86.66	77.77	80.00	91.11	78.995
16	87.50	64.58	52.08	66.66	77.08	62.50	83.33	77.08	87.50	87.50	74.581
17	74.50	66.66	39.21	64.70	76.47	66.66	82.35	78.43	88.23	92.15	72.936
18	77.77	75.92	92.96	66.66	79.62	68.51	94.44	79.62	94.44	98.14	82.808
19	75.43	68.42	40.35	63.15	75.43	66.66	87.71	78.94	94.73	87.71	73.853
20	78.33	73.33	65.00	66.66	73.33	70.00	88.33	83.33	95.00	95.00	78.831

TABLE 19: Accuracy results of WeChat text traffic dataset.

Pkts #	Bayes Net	N/Bayes	SMO	Adaboost	Bagging	OneR	Part	Hoeffbing	C4.5	R/forest	Average
2	66.66	33.33	33.33	33.33	66.66	50.00	33.33	33.33	66.66	66.66	48.32
3	33.33	22.22	33.33	33.33	44.44	55.55	22.22	33.33	55.55	22.22	35.55
4	58.33	41.66	41.66	66.66	58.66	41.66	33.33	66.66	66.66	58.33	53.36
5	73.33	60.00	53.33	86.66	80.00	73.33	26.66	86.66	73.33	46.66	65.99
6	66.66	44.44	50.00	77.77	72.22	61.11	33.33	77.77	72.22	72.22	62.77
7	85.71	85.71	61.90	80.95	85.71	71.42	28.57	80.95	85.71	47.61	71.42
8	83.33	70.83	62.50	79.16	91.66	70.83	33.33	79.16	91.66	58.33	72.07
9	70.37	62.96	66.66	88.88	74.07	59.25	29.62	88.88	77.77	48.14	66.66
10	76.66	70.00	70.00	90.00	93.33	83.33	33.33	90.00	83.33	60.00	74.99
11	84.84	78.78	75.75	90.90	81.81	66.66	48.48	90.90	90.90	54.54	76.35
12	86.11	77.77	88.88	91.66	91.66	77.77	66.66	91.66	91.66	61.11	82.49
13	79.48	82.05	69.23	89.74	76.92	74.35	74.35	89.74	84.61	38.46	75.89
14	85.71	85.71	69.04	90.47	95.23	78.57	76.19	90.47	92.85	61.90	82.61
15	71.11	84.44	68.88	88.88	91.11	80.00	68.88	88.88	93.33	62.22	79.77
16	75.00	70.83	93.75	95.83	91.66	77.08	72.91	95.83	91.66	60.41	82.49
17	82.35	82.35	92.15	96.07	92.15	80.39	70.58	96.07	96.07	41.17	82.93
18	88.88	85.15	94.44	96.29	92.59	81.48	85.18	96.29	94.44	62.96	87.77
19	87.71	94.73	96.49	100.0	92.98	84.21	87.71	100.0	100.0	63.15	90.69
20	85.00	93.33	95.00	100.0	91.66	81.66	95.00	100.0	100.0	61.66	90.33

TABLE 20: Accuracy results of WeChat video traffic dataset.

Pkts #	Bayes Net	N/Bayes	SMO	Adaboost	Bagging	OneR	Part	Hoeffbing	C4.5	R/forest	Average
2	66.66	33.33	33.33	66.66	83.33	66.66	33.33	66.66	100	66.66	61.66
3	66.66	55.55	33.33	77.77	88.88	88.88	33.33	77.77	100	77.00	69.91
4	66.66	83.33	83.33	100.0	83.33	66.66	33.33	100.0	100	66.66	78.33
5	66.66	86.66	80.00	100.0	86.66	86.66	33.33	100.0	100	73.33	81.33
6	66.66	88.88	77.77	100.0	83.33	83.33	66.66	100.0	100	72.22	83.88
7	66.66	90.47	76.19	100.0	85.71	85.71	66.66	100.0	100	76.19	84.75
8	66.66	91.66	75.00	100.0	83.33	83.33	75.00	100.0	100	75.00	84.99
9	66.66	88.88	74.07	96.29	81.48	81.48	74.07	96.29	100	74.07	83.32
10	70.00	90.00	73.33	93.33	76.66	83.33	76.66	96.66	100	73.33	83.33
11	69.69	87.87	72.72	93.93	78.78	81.81	78.78	96.96	100	75.75	83.62
12	69.44	86.11	72.22	94.44	77.77	83.33	88.88	97.22	100	75.00	84.44
13	69.23	87.17	94.87	94.87	79.48	79.48	87.17	97.43	100	74.35	86.40
14	66.66	95.23	76.19	95.23	76.19	78.57	83.33	95.23	100	71.42	83.80
15	66.66	84.44	86.66	93.33	73.33	77.77	84.44	93.33	100	62.22	82.21
16	68.75	85.41	85.41	89.58	72.91	75.00	83.33	95.83	100	62.50	81.87
17	35.70	82.35	82.35	88.23	47.05	50.98	82.35	92.15	100	41.17	70.23
18	62.96	81.48	83.33	88.88	51.85	51.85	81.48	85.18	100	40.74	72.77
19	61.40	78.94	82.45	89.47	54.38	54.38	85.96	89.47	100	42.10	73.85
20	61.66	86.66	81.66	88.33	53.33	51.66	85.00	88.33	100	41.66	73.82

TABLE 21: AUC results of WeChat audio traffic dataset.

Pkts #	Bayes Net	N/Bayes	SMO	Adaboost	Bagging	OneR	Part	Hoeffbing	C4.5	R/forest	Average
2	0.660	0.660	0.750	0.750	0.660	0.660	0.660	0.660	0.660	0.750	0.687
3	0.611	0.777	0.666	0.777	0.611	0.611	0.666	0.666	0.666	0.777	0.682
4	0.666	0.833	0.733	0.833	0.666	0.666	0.791	0.833	0.791	1.000	0.781
5	0.343	0.833	0.416	0.733	0.833	0.642	0.766	0.675	0.766	1.000	0.700
6	0.666	0.888	0.780	0.833	0.833	1.00	0.75	0.777	0.750	1.000	0.827
7	0.785	0.880	0.738	0.833	0.833	0.642	0.833	0.904	0.833	1.000	0.828
8	0.895	0.854	0.854	0.833	0.875	0.666	0.958	0.895	0.958	1.000	0.878
9	0.833	0.851	0.685	0.833	0.814	0.648	0.962	0.814	0.962	1.000	0.840
10	0.90	0.816	0.850	0.833	0.883	0.666	1.00	0.900	1.00	1.000	0.884
11	0.757	0.757	0.803	0.803	0.803	0.666	0.924	0.757	0.924	0.969	0.816
12	0.750	0.882	0.805	0.805	0.833	0.805	0.902	0.875	0.902	0.972	0.853
13	0.756	0.923	0.705	0.807	0.974	0.807	0.961	0.884	0.987	0.974	0.877
14	0.869	0.916	0.809	0.809	0.908	0.809	0.964	0.892	0.988	0.976	0.894
15	0.866	0.855	0.755	0.811	0.944	0.822	0.944	0.877	0.977	0.988	0.883
16	0.916	0.885	0.833	0.822	0.916	0.822	0.984	0.895	0.979	0.989	0.904
17	0.847	0.852	0.416	0.794	0.872	0.882	0.960	0.892	0.940	0.960	0.841
18	0.898	0.861	0.842	0.796	0.870	0.888	0.944	0.851	0.944	0.953	0.884
19	0.868	0.859	0.692	0.824	0.947	0.850	0.982	0.959	0.982	0.973	0.893
20	0.825	0.850	0.808	0.841	0.908	0.883	0.933	0.850	0.933	0.975	0.880

TABLE 22: AUC results of WeChat picture traffic dataset.

Pkts #	Bayes Net	N/Bayes	SMO	Adaboost	Bagging	OneR	Part	Hoeffbing	C4.5	R/forest	Average
2	0.666	0.666	0.666	0.666	0.666	0.666	0.666	0.833	0.666	0.666	0.682
3	0.611	0.722	0.666	0.722	0.611	0.611	0.666	0.722	0.666	0.722	0.671
4	0.666	0.833	0.791	0.791	0.666	0.666	0.833	0.916	0.833	0.875	0.787
5	0.666	0.866	0.700	0.866	0.766	0.633	0.733	0.933	0.733	0.900	0.779
6	0.666	0.861	0.833	0.805	0.861	0.666	0.888	0.944	0.888	0.944	0.835
7	0.833	0.880	0.714	0.857	0.785	0.642	0.904	0.865	0.904	0.904	0.828
8	0.812	0.895	0.812	0.833	0.833	0.666	0.916	0.958	0.916	0.875	0.851
9	0.814	0.851	0.703	0.814	0.796	0.648	0.888	0.907	0.888	0.907	0.821
10	0.883	0.900	0.750	0.800	0.816	0.666	0.866	0.900	0.900	0.883	0.836
11	0.787	0.893	0.666	0.818	0.848	0.681	0.909	0.909	0.909	0.893	0.831
12	0.902	0.847	0.805	0.833	0.847	0.833	0.902	0.875	0.902	0.902	0.864
13	0.910	0.884	0.705	0.833	0.935	0.833	0.897	0.910	0.910	0.948	0.876
14	0.892	0.880	0.821	0.833	0.892	0.833	0.928	0.916	0.952	0.952	0.889
15	0.888	0.844	0.811	0.833	0.911	0.833	0.933	0.888	0.900	0.955	0.879
16	0.937	0.822	0.760	0.833	0.885	0.812	0.916	0.885	0.896	0.920	0.866
17	0.872	0.833	0.696	0.823	0.882	0.833	0.911	0.892	0.941	0.960	0.864
18	0.888	0.879	0.814	0.833	0.898	0.842	0.972	0.898	0.972	0.990	0.898
19	0.877	0.842	0.701	0.815	0.877	0.833	0.938	0.894	0.973	0.938	0.868
20	0.891	0.866	0.825	0.833	0.866	0.850	0.941	0.916	0.975	0.967	0.893

TABLE 23: AUC results of WeChat text traffic dataset.

Pkts #	Bayes Net	N/Bayes	SMO	Adaboost	Bagging	OneR	Part	Hoeffbing	C4.5	R/forest	Average
2	0.666	0.833	0.833	0.833	0.666	0.666	0.833	0.916	0.833	1.000	0.807
3	0.666	0.944	0.888	0.833	0.777	0.666	0.888	0.944	0.888	1.000	0.849
4	0.916	0.833	0.833	0.833	0.916	0.666	1.000	0.916	1.000	1.000	0.891
5	0.900	0.933	0.866	0.833	0.933	0.666	1.000	0.933	1.000	1.000	0.906
6	0.888	0.916	0.861	0.833	0.944	0.833	1.000	0.916	0.909	1.000	0.910
7	0.880	0.928	0.880	0.833	0.952	0.833	1.000	0.928	1.000	1.000	0.923
8	0.875	0.916	0.875	0.833	0.958	0.875	1.000	0.916	1.000	1.000	0.924
9	0.870	0.907	0.870	0.833	0.944	0.870	0.981	0.907	0.981	1.000	0.916
10	0.866	0.916	0.866	0.850	0.950	0.883	0.983	0.883	0.966	1.000	0.916
11	0.863	0.906	0.878	0.848	0.939	0.893	0.984	0.893	0.969	0.985	0.915
12	0.861	0.916	0.875	0.847	0.930	0.944	0.986	0.888	0.972	1.000	0.921
13	0.974	0.897	0.871	0.846	0.935	0.935	0.987	0.900	0.974	1.000	0.931
14	0.880	0.892	0.857	0.833	0.976	0.916	0.976	0.880	0.976	1.000	0.918
15	0.933	0.888	0.655	0.833	0.922	0.922	0.966	0.866	0.966	1.000	0.895
16	0.927	0.875	0.812	0.843	0.927	0.916	0.979	0.864	0.947	1.000	0.909
17	0.911	0.754	0.705	0.823	0.911	0.911	0.960	0.735	0.941	1.000	0.865
18	0.916	0.759	0.703	0.814	0.907	0.907	0.925	0.759	0.944	1.000	0.863
19	0.912	0.771	0.710	0.807	0.894	0.929	0.947	0.771	0.947	1.000	0.868
20	0.908	0.758	0.708	0.808	0.933	0.925	0.941	0.766	0.941	1.000	0.868

TABLE 24: AUC results of WeChat video traffic dataset.

Pkts #	Bayes Net	N/Bayes	SMO	Adaboost	Bagging	OneR	Part	Hoeffbing	C4.5	R/forest	Average
2	0.666	0.800	0.833	0.833	0.666	0.666	0.666	0.833	0.666	0.833	0.746
3	0.666	0.777	0.611	0.666	0.611	0.611	0.666	0.722	0.666	0.777	0.677
4	0.708	0.708	0.791	0.791	0.708	0.666	0.833	0.791	0.833	0.833	0.766
5	0.766	0.866	0.733	0.866	0.800	0.633	0.933	0.900	0.933	0.866	0.829
6	0.750	0.805	0.861	0.833	0.722	0.666	0.888	0.861	0.888	0.861	0.813
7	0.809	0.857	0.738	0.928	0.928	0.642	0.904	0.928	0.904	0.928	0.856
8	0.812	0.854	0.791	0.916	0.854	0.666	0.895	0.958	0.895	0.958	0.859
9	0.833	0.796	0.740	0.851	0.814	0.648	0.944	0.870	0.944	0.888	0.832
10	0.850	0.916	0.800	0.883	0.85	0.666	0.950	0.966	0.950	0.916	0.874
11	0.878	0.833	0.772	0.924	0.893	0.742	0.954	0.909	0.954	0.954	0.881
12	0.944	0.888	0.805	0.930	0.888	0.833	0.958	0.958	0.958	0.958	0.912
13	0.846	0.871	0.692	0.897	0.910	0.871	0.948	0.884	0.948	0.923	0.879
14	0.845	0.892	0.809	0.928	0.928	0.880	0.952	0.976	0.952	0.964	0.912
15	0.844	0.900	0.811	0.855	0.922	0.844	0.944	0.955	0.944	0.966	0.898
16	0.968	0.885	0.802	0.875	0.854	0.864	0.979	0.958	0.979	0.958	0.912
17	0.960	0.901	0.705	0.911	0.911	0.852	0.980	0.960	0.980	0.980	0.914
18	0.972	0.907	0.814	0.944	0.925	0.925	0.981	0.962	0.981	0.972	0.938
19	0.982	0.921	0.815	0.938	0.973	0.938	1.000	0.964	1.000	1.000	0.953
20	0.975	0.975	0.808	0.925	0.966	0.975	1.000	0.958	1.000	1.000	0.958

TABLE 25: Accuracy results of HIT Trace traffic dataset.

Pkts #	Bayes Net	N/Bayes	SMO	Adaboost	Bagging	OneR	Part	Hoeffbing	C4.5	R/forest	Average
2	33.33	41.66	41.66	33.33	16.66	16.66	33.33	16.66	33.33	41.66	30.82
3	27.77	38.88	11.11	22.22	22.22	11.11	38.88	27.77	38.88	44.44	28.32
4	16.66	33.33	45.83	25.00	16.66	16.66	41.66	45.83	41.66	45.83	32.91
5	23.33	40.00	23.33	26.66	36.66	13.33	40.00	43.33	40.00	46.66	33.3
6	33.33	50.00	47.22	30.55	30.55	16.66	38.88	47.22	38.88	30.55	36.38
7	23.80	30.95	19.04	23.80	30.95	14.28	26.19	38.09	28.57	38.09	27.37
8	29.16	31.25	31.25	29.16	35.41	16.66	37.50	35.41	39.58	37.50	32.28
9	27.77	35.18	24.07	27.77	37.03	14.81	37.03	37.03	35.18	42.59	31.84
10	30.00	36.66	33.33	30.00	35.00	16.66	35.00	35.00	35.00	45.00	33.16
11	22.72	31.81	25.75	30.33	34.84	27.27	42.42	36.36	42.42	45.45	33.93
12	34.72	36.11	34.72	33.33	34.72	37.50	38.88	33.33	41.66	50.00	37.49
13	24.35	23.07	21.79	24.35	29.48	23.07	34.61	26.92	32.05	32.05	27.17
14	24.35	23.07	21.79	24.35	29.48	23.07	34.61	26.92	32.05	32.05	27.17
15	34.44	34.44	22.22	32.22	45.55	33.33	40.00	36.66	40.00	45.55	36.44
16	38.54	38.54	33.33	33.33	36.45	36.45	41.66	36.45	43.75	44.79	38.32
17	35.29	36.27	19.60	32.35	44.11	31.37	46.07	30.39	44.11	42.15	36.17
18	38.88	37.96	30.55	33.33	50.92	37.96	50.92	37.03	49.07	52.77	41.93
19	35.08	37.71	19.29	33.33	42.98	33.33	41.22	35.08	42.10	45.61	36.57
20	35.83	38.33	30.83	33.33	42.50	37.50	42.50	36.66	44.16	44.16	38.58

TABLE 26: AUC results of HIT Trace II traffic dataset.

Pkts #	Bayes Net	N/Bayes	SMO	Adaboost	Bagging	OneR	Part	Hoeffding	C4.5	R/forest	Average
2	0.666	0.708	0.708	0.666	0.583	0.583	0.666	0.583	0.666	0.708	0.653
3	0.638	0.694	0.555	0.611	0.611	0.555	0.694	0.638	0.694	0.722	0.641
4	0.666	0.666	0.543	0.625	0.583	0.583	0.708	0.729	0.708	0.729	0.654
5	0.305	0.700	0.305	0.343	0.683	0.566	0.700	0.518	0.700	0.733	0.555
6	0.666	0.750	0.556	0.652	0.652	0.583	0.694	0.736	0.694	0.652	0.663
7	0.619	0.654	0.595	0.619	0.654	0.571	0.630	0.690	0.642	0.690	0.636
8	0.645	0.656	0.656	0.645	0.677	0.583	0.687	0.677	0.697	0.687	0.661
9	0.638	0.675	0.620	0.638	0.685	0.574	0.685	0.685	0.675	0.712	0.658
10	0.650	0.683	0.666	0.650	0.675	0.583	0.675	0.675	0.675	0.725	0.665
11	0.613	0.659	0.628	0.651	0.674	0.636	0.712	0.681	0.712	0.727	0.669
12	0.673	0.680	0.673	0.666	0.673	0.687	0.694	0.666	0.708	0.750	0.687
13	0.621	0.615	0.608	0.621	0.647	0.615	0.673	0.634	0.660	0.660	0.635
14	0.621	0.615	0.608	0.621	0.375	0.615	0.673	0.634	0.660	0.660	0.608
15	0.672	0.672	0.611	0.800	0.727	0.666	0.700	0.683	0.700	0.727	0.695
16	0.692	0.692	0.666	0.666	0.682	0.682	0.502	0.682	0.718	0.723	0.670
17	0.437	0.681	0.261	0.661	0.720	0.656	0.730	0.651	0.526	0.710	0.603
18	0.694	0.689	0.652	0.666	0.754	0.689	0.754	0.685	0.745	0.763	0.709
19	0.675	0.688	0.596	0.666	0.714	0.666	0.706	0.675	0.710	0.728	0.682
20	0.679	0.691	0.654	0.666	0.712	0.687	0.712	0.683	0.720	0.720	0.692

Competing Interests

The authors declare that they have no competing interests.

References

- [1] A. Este, F. Gringoli, and L. Salgarelli, "Support vector machines for TCP traffic classification," *Computer Networks*, vol. 53, no. 14, pp. 2476–2490, 2009.
- [2] W. Li and A. W. Moore, "A machine learning approach for efficient traffic classification," in *Proceedings of the 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS '07)*, pp. 310–317, IEEE, Istanbul, Turkey, October 2007.
- [3] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *Proceedings of the ACM SIGMETRICS International Conference On Measurement and Modeling of Computer Systems (SIGMETRICS '05)*, pp. 50–60, Alberta, Canada, June 2005.
- [4] A. Moore, D. Zuev, and M. Crogan, *Discriminators for Use in Flow-Based Classification*, Department of Computer Science, Queen Mary and Westfield College, 2005.
- [5] F. Palmieri and U. Fiore, "A nonlinear, recurrence-based approach to traffic classification," *Computer Networks*, vol. 53, no. 6, pp. 761–773, 2009.
- [6] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, "Network anomaly detection with the restricted Boltzmann machine," *Neurocomputing*, vol. 122, pp. 13–23, 2013.
- [7] A. Dainotti, A. Pescapé, and K. C. Claffy, "Issues and future directions in traffic classification," *IEEE Network*, vol. 26, no. 1, pp. 35–40, 2012.
- [8] B. Qu, Z. Zhang, L. Guo, and D. Meng, "On accuracy of early traffic classification," in *Proceedings of the IEEE 7th International Conference on Networking, Architecture and Storage (NAS '12)*, pp. 348–354, Xiamen, China, June 2012.
- [9] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 2, pp. 23–26, 2006.
- [10] N.-F. Huang, G.-Y. Jai, and H.-C. Chao, "Early identifying application traffic with application characteristics," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 5788–5792, Beijing, China, May 2008.
- [11] N.-F. Huang, G.-Y. Jai, H.-C. Chao, Y.-J. Tzang, and H.-Y. Chang, "Application traffic classification at the early stage by characterizing application rounds," *Information Sciences*, vol. 232, pp. 130–142, 2013.
- [12] A. Este, F. Gringoli, and L. Salgarelli, "On the stability of the information carried by traffic flow features at the packet level," *SIGCOMM Computer Communication Review*, vol. 39, no. 3, pp. 13–18, 2009.
- [13] A. Rizzi, S. Colabrese, and A. Baiocchi, "Low complexity, high performance neuro-fuzzy system for Internet traffic flows early classification," in *Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC '13)*, pp. 77–82, Sardinia, Italy, July 2013.
- [14] T. T. T. Nguyen, G. Armitage, P. Branch, and S. Zander, "Timely and continuous machine-learning-based classification for interactive IP traffic," *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, pp. 1880–1894, 2012.
- [15] A. Dainotti, A. Pescapé, and C. Sansone, "Early classification of network traffic through multi-classification," in *Proceedings of the International Workshop on Traffic Monitoring and Analysis*, Springer, Vienna, Austria, April 2011.
- [16] L. Peng, B. Yang, and Y. Chen, "Effective packet number for early stage internet traffic identification," *Neurocomputing*, vol. 156, pp. 252–267, 2015.
- [17] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 2, pp. 23–26, 2006.

- [18] Y. Lim, H. Kim, J. Jeong, C. Kim, T. Kwon, and Y. Choi, "Internet traffic classification demystified: on the sources of the discriminative power," in *Proceedings of the 6th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT '10)*, ACM, Philadelphia, Pa, USA, 2010.
- [19] More than 300 million users engage, http://www.chinadaily.com.cn/cndy/2013-01/17/content_16128915.htm.
- [20] By the numbers: 50+ Amazing WeChat Statistics, <http://expandedramblings.com/index.php/wechat-statistics/>.
- [21] Q. Huang, P. P. C. Lee, C. He, J. Qian, and C. He, "Fine-grained dissection of WeChat in cellular networks," in *Proceedings of the 23rd IEEE International Symposium on Quality of Service (IWQoS '15)*, pp. 309–318, Portland, Ore, USA, June 2015.
- [22] K. Church and R. De Oliveira, "What's up with whatsapp?: comparing mobile instant messaging behaviors with traditional SMS," in *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '13)*, pp. 352–361, ACM, Munich, Germany, August 2013.
- [23] K. P. O'Hara, M. Massimi, R. Harper, S. Rubens, and J. Morris, "Everyday dwelling with whatsapp," in *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*, 1142, 1131 pages, ACM, Baltimore, Md, USA, February 2014.
- [24] P. Fiadino, M. Schiavone, and P. Casas, "Vivisecting whatsapp through large-scale measurements in mobile networks," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 133–134, 2014.
- [25] Y. Liu and L. Guo, "An empirical study of video messaging services on smartphones," in *Proceedings of the 24th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '14)*, pp. 79–84, March 2014.
- [26] M. Shafiq, X. Yu, and A. A. Laghari, "WeChat text messages service flow traffic classification using machine learning technique," in *Proceedings of the 6th International Conference on IT Convergence and Security (ICITCS '16)*, pp. 1–5, Prague, Czech Republic, September 2016.
- [27] Wireshark, <http://www.wireshark.org/>.
- [28] A. Este, F. Gringoli, and L. Salgarelli, "On the stability of the information carried by traffic flow features at the packet level," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 3, pp. 13–18, 2009.
- [29] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.
- [30] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *IEEE Transactions on Medical Imaging*, vol. 16, no. 2, pp. 187–198, 1997.
- [31] L. R. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '86)*, Tokyo, Japan, April 1986.
- [32] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proceedings of the IEEE-IECEJ-ASJ International Conference on Acoustics, Speech, and Signal Processing (ICASSP '86)*, vol. 86, pp. 49–52, Tokyo, Japan, April 1986.
- [33] H. Peng, Mutual information Matlab toolbox, <http://www.mathworks.com/matlabcentral/fileexchange/14888-mutual-information-computation>.
- [34] "Weka 3: Data Mining Software in Java," <http://www.cs.waikato.ac.nz/ml/weka/>.
- [35] L. Peng, B. Yang, Y. Chen, and Z. Chen, "Effectiveness of statistical features for early stage internet traffic identification," *International Journal of Parallel Programming*, vol. 44, no. 1, pp. 181–197, 2016.
- [36] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, no. 2-3, pp. 131–163, 1997.
- [37] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Machine Learning*, vol. 29, no. 2-3, pp. 103–130, 1997.
- [38] M. E. Maron, "Automatic indexing: an experimental inquiry," *Journal of the ACM (JACM)*, vol. 8, no. 3, pp. 404–417, 1961.
- [39] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [40] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the 2nd European Conference on Computational Learning Theory (EuroCOLT '95)*, Springer, Barcelona, Spain, 1995.
- [41] R. C. Holte, "Very simple classification rules perform well on most commonly used datasets," *Machine Learning*, vol. 11, no. 1, pp. 63–90, 1993.
- [42] E. Frank and I. H. Witten, "Generating accurate rule sets without global optimization," in *Proceedings of the 15th International Conference on Machine Learning*, July 1998.
- [43] R. C. Quinlan, *4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc, San Francisco, Calif, USA, 1993.
- [44] R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD '96)*, pp. 202–207, Portland, Ore, USA, August 1996.
- [45] V. Svetnik, A. Liaw, C. Tong, J. Christopher Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: a classification and regression tool for compound classification and QSAR modeling," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 6, pp. 1947–1958, 2003.
- [46] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [47] J. Kim, J. Hwang, and K. Kim, "High-performance internet traffic classification using a Markov model and Kullback-Leibler divergence," *Mobile Information Systems*, vol. 2016, Article ID 6180527, 13 pages, 2016.
- [48] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.
- [49] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano, F. Risso, and K. C. Claffy, "Gt: picking up the truth from the ground for internet traffic," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 5, pp. 12–18, 2009.
- [50] D. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC Press, 2003.
- [51] D. Quade, "Using weighted rankings in the analysis of complete blocks with additive block effects," *Journal of the American Statistical Association*, vol. 74, no. 367, pp. 680–683, 1979.

- [52] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference*, Springer, Berlin, Germany, 2011.
- [53] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, "Performance measures for information extraction," in *Proceedings of the DARPA Broadcast News Workshop*, pp. 249–252, Herndon, Va, USA, February 1999.
- [54] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*, Springer, 1st edition, 2008.
- [55] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.