Int J Parallel Prog (2016) 44:181–197
DOI 10.1007/s10766-014-0337-2

Effectiveness of Statistical Features for Early Stage Internet Traffic Identification

Lizhi Peng · Bo Yang · Yuehui Chen · Zhenxiang Chen

Received: 3 July 2014 / Accepted: 9 October 2014 / Published online: 18 January 2015

Abstract Identifying network traffic at their early stages accurately is very important for the application of traffic identification. In recent years, more and more studies have tried to build effective machine learning models to identify traffic with the few packets at the early stage. Packet sizes and statistical features have been proved to be effective features which are widely used in early stage traffic identification. However, an important issue is still unconcerned, that is whether there exists essential effectiveness differences between the two kinds of features. In this paper, we set out to evaluate the effectiveness of statistical features in comparing with packet sizes. We firstly extract the packet sizes and their statistical features of the first six packets on three traffic data sets. Then the mutual information between each feature and the corresponding traffic type label is computed to show the effectiveness of the feature. And then we execute crossover identification experiments with different feature sets using ten well-known machine learning classifiers. Our experimental results show that most classifiers get almost the same performances using packet sizes and statistical features for early stage traffic identification. And most classifiers can achieve high identification accuracies using only two statistical features.

Keywords Feature selection · Early stage traffic classification · Machine learning

1 Introduction

In the past decade, accurate traffic classification has become more and more important for network managements including deploying QoS-aware mechanisms, bandwidth

L. Peng · B. Yang (✉) · Y. Chen · Z. Chen
Shandong Provincial Key Laboratory for Network Based Intelligent Computing,
University of Jinan, Jinan 250022, People's Republic of China
e-mail: yangbo@ujn.edu.cn

budget managing, intrusion detection, etc. There are two classical techniques which are effective under traditional network conditions: port-based and payload-based methods. Unfortunately, nowadays many Internet applications use dynamic port numbers instead of well-known ones for communications, which leads to difficulty of identifying traffics by port numbers. Also many applications encrypt the data to be transmitted to avoid being detected. Therefore, payload-based techniques become ineffectual for these traffics since it is no use to inspect encrypted packet data. In recent years, machine learning techniques have been introduced into traffic classification researches and have been proven to be promising techniques [20,25,38]. However, most machine learning based traffic identification techniques extract features on a whole traffic instance [10,20,24]. The most widely used feature extracting method is presented by Moore et al. in 2005 [23]. They extract 248 statistical features based on a whole flow, such as maximum, minimum and average values of packet size, RTT. And classifiers using such statistical features can get very high performances in traffic identification. However, in real circumstances, it makes no sense to recognize Internet traffic when they have ended. Thus, we must identify Internet traffic accurately in their early stage so that we can apply subsequent management and security policies. Therefore, some researchers have turned to find effective models which are able to identify Internet traffic at their early stage. And this makes early stage identification to become a hot topic in traffic identification researches [5]. Qu et al. [30] have studied the problem of accuracy of early stage traffic identification, and found that it is possible to identify traffic accurately at its early stage.

It is relatively hard to recognize a traffic by only using several early stage packets. According to Dainotti [5], limiting the number of packets used to extract features offers several benefits including lower feature extraction complexity. However, are the simple features extracted based on so few packets effective enough for identification? Thus, the key problem of early stage traffic identification is to find out effective features in early stage of traffic. Bernaille et al. [1] presented a famous early stage traffic identification technique in 2006. They use the size of the first few data packets of each TCP flow as the features, and by applying the K-means clustering technique, they got high identification rates for ten types of application traffic. Este et al. [9] have proved in 2009 that early stage packets of an Internet flow carry enough information for traffic classification. They analyzed round trip time (RTT), packet size, inter-arrival time (IAT) and packet direction of early stage packets and found that packet size is the most effective feature for early stage classifications. Huang et al. [15] have studied the early stage application characteristics and used them for classification effectively in 2008. Recently, they extracted early stage traffic features by analyzing the negotiation behaviors of different applications. They use packet size (PS) and inter packet time (IPT) of the first ten packets for some classifiers, while for other classifiers, they use average and standard deviation values of PS and IPT of the early packets. They applied these features for machine learning based classifiers with high performances [16]. Hullár et al. [17] proposed an automatic machine learning based method consuming limited computational and memory resources for P2P traffic identification at early stage. Dainotti et al. [6] construct high effective hybrid classifiers and apply a hybrid feature extraction method for early stage traffic classification. Nguyen et al. [26] use statistical features derived from sub-flows for timely identification of VoIP traffics,

they extend the concept of early stage to "timely", since a sub-flows refers to a small number of most recent packets taken at any point in a flows lifetime. Rizzi et al. [32] proposed a highly efficient neuro-fuzzy system for early stage traffic identification.

For the studies mentioned above, packet level features or statistical features were applied to identify Internet traffics, and Este et al. [9] have evaluated the effectiveness of packet level features. However, the effectiveness of statistical features for the early stage is yet unknown. The packet level features are able to show the detailed characteristics of an Internet traffic, while they can not catch its global characteristics. On the contrary, the statistical features such as the average payload size and the standard deviation of payload sizes are able to show the global distribution characteristics of a traffic. However, the number of packets in the early stage of traffic is considerably small, usually, it ranges from 4 to 10. Thus, is an early stage statistical feature able to include enough information for identification, and are early stage statistical feature sets more effective than packet level feature sets? These questions should be answered.

Contributions In this paper, we set out to study the effectiveness of the early stage statistical features of Internet traffics. We try to answer the above mentioned question using the mutual information analysis and experimental methods. 3 traffic data sets and ten machine learning classifiers are applied for our experiments. We use the application layer payload sizes as the original packet level features, and 5 statistics as the statistical features. Firstly, the mutual information of each feature and the traffic type label is computed to evaluate its effectiveness preliminary. Then we build 6 feature sets covering the pure original feature set, the pure statistical feature set and the hybrid feature set, and then all selected classifiers are applied on these feature sets to validate the effectiveness of selected features.

The rest of the paper is organized as follows: Sect. 2 illustrates the methods applied in our study, include the mutual information theory and the details of the experimental methods. We introduce the characteristics of the selected data sets and classifiers in Sects. 3 and 4 respectively. And the details of experimental results and analysis are given in Sect. 5, and we also do some discussions in this section. Finally, we make some conclusions in Sect. 6.

2 Methodology

2.1 Features

- *Payload size* The payload size has been proved to be the most effective early stage packet level feature [9]. We use the payload sizes of the first six packets as the original early stage traffic features in this study. All statistical features are computed based on the payload sizes. And we use the abbreviation of *ps* for the payload size in this paper.
- *Average* The average is also known as the arithmetical mean, which is an extensively used statistical indicator. This feature is calculated as follows:

$$avg = \sum_{i=1}^{n} ps_i \tag{1}$$

– *Standard deviation* The standard deviation shows how much variation or dispersion from the average exists. And the feature is defined as:

$$stdev = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(ps_i - avg)^2} \tag{2}$$

where n is the number of packets, i. e. six in this study.
– *Maximum and minimum* The maximum and minimum payload size are also applied in the study, and we use the abbreviations of max and min respectively.
– *Geometric mean* The geometric mean is another mean which is defined as:

$$gm = \sqrt[n]{ps_1 ps_2 ... ps_n} \tag{3}$$

– *Variance* The variance measures how far the payload sizes is spread out, which is defined as:

$$var = \frac{1}{n-1}\sum_{i=1}^{n}(ps_i - avg)^2 \tag{4}$$

2.2 Mutual Information

Mutual information is a useful measure in information theory which is widely used for feature selection [28], image processing [21], speech recognition [2] and so on. The mutual information of two random variables X, Y is a measure of the variables' mutual dependence. In information theory, mutual information is defined as

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \\ &= H(X,Y) - H(X|Y) - H(Y|X) \end{aligned} \tag{5}$$

where $H(X)$ and $H(Y)$ are the marginal entropies of X and Y respectively, $H(X|Y)$ and $H(Y|X)$ are the conditional entropies, and $H(X, Y)$ is the joint entropy of X and Y. From the point view of set theory, the relationships among $H(X)$, $H(Y)$, $H(X|Y)$, $H(Y|X)$, $H(X, Y)$ and $I(X; Y)$ can be shown Fig. 1 depicts. According to Shannon's definition of entropy, we have

$$H(X) = -\sum_{x\in X} p(x)log(p(x)) \tag{6}$$

$$H(Y) = -\sum_{y\in Y} p(y)log(p(y)) \tag{7}$$

$$H(X,Y) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)log(p(x,y)) \tag{8}$$

where $p(.)$ is the probability distribution function of a random variable. We use the tree equations in Eq. (5) and can obtain the computational formula of mutual information

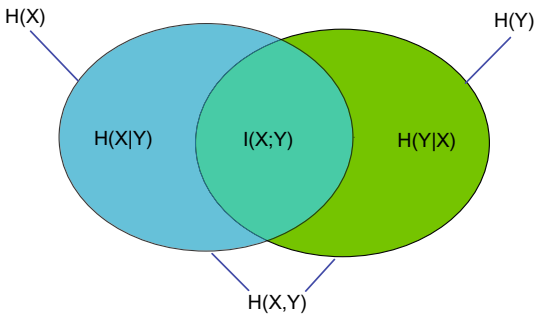

Fig. 1 The relationships among the entropies and the mutual information

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) log(\frac{p(x,y)}{p(x)p(y)}) \tag{9}$$

In the case of continuous random variables, the summation is replaced by a definite double integral:

$$I(X;Y) = \int_Y \int_X p(x,y) log(\frac{p(x,y)}{p(x)p(y)}) dxdy \tag{10}$$

There are many open source software for mutual information computation. And in our study, we apply Peng's mutual information Matlab toolbox [27].

2.3 Experimental Framework

We carry out our study as Fig. 2 depicts.

- *Filter mouse traffic* A mouse traffic is that with few packets or bytes. It is hard to identify mouse traffic in Internet because they are too "little" to obtain effective features. Furthermore, it makes little sense to identify such traffic from the viewpoint of traffic identification, since they have little effects on network management [8]. In this study, we define mouse traffic as those that have no more than 10 non-zero payload packets. We firstly filter such mouse traffic from the original traffic traces. And after this step, each traffic instance in the data sets has at least 10 non-zero payload packets.
- *Extract features* For each traffic instance in an original data set, we extract the payload sizes of the earliest 6 non-zero payload packets. Then the 6 integer values are put into the feature data set along with the application type label of the traffic instance. It should be noticed that the order of the features must be in accord with the order of the packets, i.e. the first feature is the payload size of the first packet, and the second feature is that of the second packet, and so forth. And then all derived statistical features are computed based on the 6 payload size features.
- *Mutual information analysis* We compute the mutual information between each feature and its corresponding traffic type label according to formula (9) and (10). And the average value of each feature on each data set is also computed. Then we

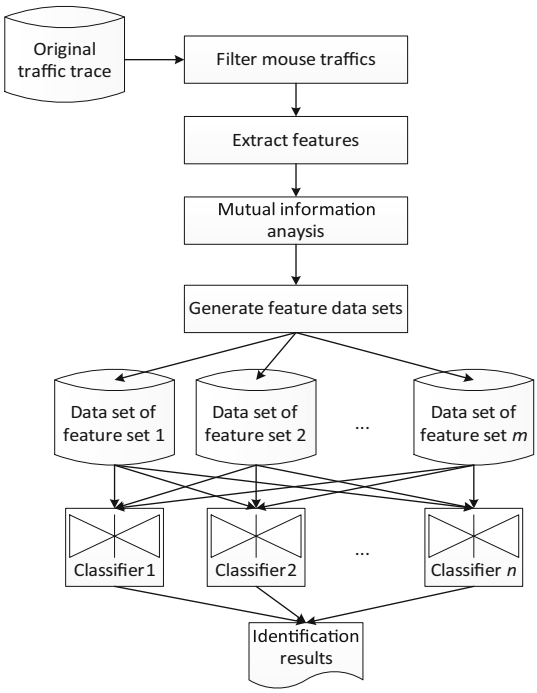

Fig. 2 Methodology framework

evaluate the effectiveness of each feature by its mutual information value. Based on the evaluating results, m feature sets used in the identification experiments are selected.

- *Generate feature data sets* For each selected feature set, we generate a corresponding data set which only contains the data of the selected features in the feature set. After this step, we get m feature data sets using for the following identification experiments.
- *Identification* We select n classifiers for this step which will be depicted in Sect. 4. For each original traffic data set, $m \times n$ crossover identification experiments will be executed using the selected classifiers on the m new-generated data sets. 5-folder crossover validation is applied for each single experiment. And we use the total identification accuracy as the performance measure. It should be noticed that we do not care the identification performances of a single classifier, because the main goal of the study is to evaluate the effectiveness of statistical features, but not to find a more effective classifier. Therefore, we will give the results according the new generated data sets of different feature sets.

3 Data Sets

We select two sets of open network traffic traces, and a set of traces collected in our campus network for our study. The characteristics of the selected traces are depicted in Table 1.

Table 1 Characteristics of the selected network traffic traces

Auckland II traces			UNIBS traces			UJN traces		
Type	#inst	Bytes	Type	#inst	Bytes	Type	#inst	Bytes
ftp	251	136,241	bittorrent	3,571	6,393,487	Web browser	11,890	58,025,350
ftp-data	463	5,260,804	edonkey	379	241,587	Chat	11,478	60,212,804
http	23,721	139,421,961	http	25,729	107,342,346	Cloud disk	1,563	109,552,924
imap	193	86,455	imap	327	860,226	Live update	2,169	28,759,962
pop3	498	98,699	pop3	2,473	4,292,419	Stream media	810	7,85,556
smtp	2,602	1,230,528	skype	801	805,453	Mail	803	2,092,862
nntp	274	22,108	msn	60	3,753	ftp	37	161,587
ssh	237	149,502	smtp	120	43,566	P2P	326	2,521,089
DNS	5,488	511,137	urd	650	132,209	Other	1,408	3,635,558
telnet	37	21,171	ssh	23	39,456	–	–	–

3.1 Auckland II Traffic Traces

Auckland II is a collection of long GPS-synchronized traces taken using a pair of DAG 2 cards at the University of Auckland which is available at [36]. There are 85 trace files which were captured from November 1999 to July 2000. Most traces were targeted at 24 h runs, but hardware failures have resulted in most traces being significantly shorter. We selected two trace files captured at Feb 14 2000 (20000214-185536-0.pcap and 20000214-185536-1.pcap) for our study. The traces include only the header bytes, with a maximum amount of 64 bytes for each frame, while the application payload is fully removed. And all IP addresses anonymised using Crypto-Pan AES encryption. The header traces were captured with a GPS synchronized mechanism using a DAG3.2E card connected to a 100 Mbps Ethernet hub interconnecting the University's firewall to their border router.

Since the application payloads were not recorded in Auckland II, DPI tools are invalid to obtain ground truths. The only way to pick out the original application type is using port numbers. In this study, we only accounted the TCP case since TCP is the predominant transport layer protocol. Each flow is thus assigned to the class identified by the server port. We selected 8 main types from Auckland II traces and filtered mouse flows with no more than 10 non-zero packets as illustrated in Sect. 2.

3.2 UNIBS Traffic Traces

UNIBS is another opening traffic traces developed by Prof. F. Gringoli and his research team, available at [35]. They developed a useful system namely GT [18] to application ground truths of captured Internet traffic. The traces were collected on the edge router of the campus network of the University of Brescia on three consecutive working days (Sept 30, Oct 1 and 2 2009). They are composed of traffic generated by a set of twenty workstations running the GT client daemon. Traffic were collected by running

Tcpdump [34] on the Faculty's router, which is a dual Xeon Linux box that connects the network to the Internet through a dedicated 100 Mb/s uplink. 99 % flows in UNIBS are TCP flows. Therefore, we again use TCP flows in this data set for our study. By using GT, UNIBS traces recorded the application information of each captured flow. We can get the application ground truths by both TCP port numbers and GT records. We also chose 8 main types in UNIBS for our study which are shown in Table 2. Different from Auckland II traces, there are two popular P2P applications in this data set, bittorrent and edonkey, recorded by GT. Skype is also selected as an import Internet application. Flows with no more than ten non-zero payload packets are also filtered.

3.3 UJN Traffic Traces

The third data set is collected in a laboratory network of the University of Jinan using Traffic Labeler (TL) [29]. The TL system captures all user socket calls and their corresponding application process information in the user mode on a Windows host, and sends the information to an intermediate NDIS driver in the kernel mode. The intermediate driver writes the application type information on the TOS field of the IP packets whose network 5-tuples (src_ip, src_port, dst_ip, dst_port, protocol) match with the network 5-tuple of the socket call. By this mean, each IP packet sent from the Windows host carries their application information. Therefore, traffic samples collected on the network have been labeled with the accurate application information and can be used for training effective traffic classification models. We deployed 10 TL instances on Windows user hosts in the laboratory network of Provincial Key Laboratory for Network Based Intelligent Computing. A mirror port of the uplink port of the switch was set, and a data collector was deployed at the mirror port. The deployed TL instances ran at work hours every day. The data collecting process lasted 2 days in May 2013. Again, flows with no more than 10 non-zero payload packets are also filtered.

4 Classifiers

We execute our identification experiments using 10 well-known machine learning classifiers. We use Weka data mining software [37] as our experiment tool. All classifiers are run in Weka and all generated data sets are formatted into the Weka data file with the extension name of "arff". The classifiers we selected fall into five categories according to Weka:

- *Bayes* Bayes classifiers are based on Bayes theorem, which is widely applied in many engineering areas. In this study, we choose Naive Bayes classifier [7,22] and Bayesian network (BayesNet) [12] as Bayes classifiers.
- *Meta* Strictly speaking, meta classifier is a kind of classification framework based on a specific classifier. This technique firstly trains a group of "weak learn", and then generate a "strong learn" based on the weak learns. We choose adaptive Boost M1 (AdaBoost) [13] and Bagging [3] for our study.

Table 2 Classifiers selected in the study

Classifiers	Type
NaiveBayes [7,22]	Bayes
BayesNet [12]	Bayes
AdaBoost [13]	Meta
Bagging [3]	Meta
OneR [14]	Rule
PART [11]	Rule
KNN [4]	Lazy learning
J48 [31]	Trees
NBTree [19]	Trees
RandomForest [33]	Trees

- *Rule* As the name suggests, a rule based classifier extracts rules using a specific policy, e. g. probability and decision trees, and uses the rules to classify testing data. OneR [14] and PART [11] are selected for this category in this study.
- *Trees* This refers to decision trees. A decision tree divides the target feature space hierarchically. Each division produces a node on the decision trees. A classification procedure is a procedure that goes from the root node to a specific leaf node on the tree. In this study, C4.5 decision trees (J48) [31], Naive Bayesian trees (NBTree) [19] and random forest (RandomForest) [33] are selected for this category.
- *Lazy learning* Strictly speaking, there is no general training procedure for a lazy learning classifier. It just loads the training data in the training phase, and executes real classification decisions in the testing phase. We choose the k-nearest neighbor (KNN) [4] classifier for this category.

Table 2 lists all classifiers applied in this study. We cite the original literature of each classifier in the table. Readers can find technical details of each classifier in its corresponding literature.

5 Experimental Results and Analysis

5.1 Mutual Information Analysis

We show the mutual information between each feature and the corresponding traffic type label for each data set in Fig. 3. In the figure, we use the abbreviation of each feature: ps_i is the payload size of the ith packet, and the abbreviations of the statistical features are in accord with that described in Sect. 2. And the exact data are listed in Table 3 in the Appendix. The variance is the best performed feature which achieves the highest mutual information value for each of the three data sets. The payload sizes of the first two packets and the minimum payload size get low level mutual information. The results mean that the variance is the most effective feature in all of the compared features from the point of view of mutual information. On the contrary, the minimum payload size and the payload sizes of the first two packets contain few identification

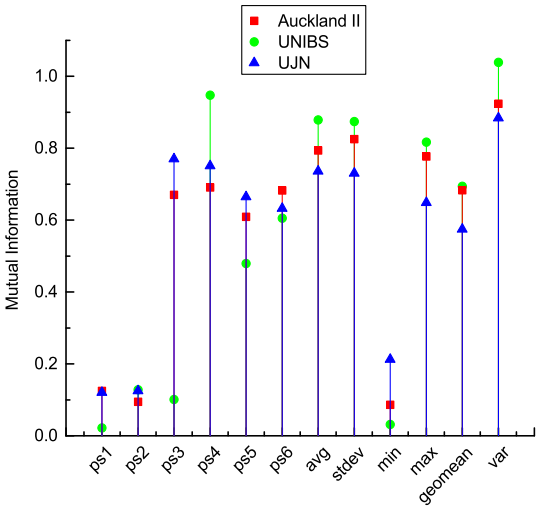

Fig. 3 Mutual information of packet sizes and statistical features

information. When observing the mutual information of the subsequent four packets, it can be seen that all values are far higher than that of the first two packets, except the value of ps_3 for UNIBS data set. Thus, we say that the 3rd-6th packets contain the vast majority of identification information. Most statistical features show their effectiveness in Fig. 2. All statistical features except the min feature gain high mutual information values for all of the three data sets. It is somehow surprising that the min feature gains low values while the max feature hits considerable high values. In many studies using statistics, the minimum is always applied together with the maximum because they are a couple of contrary measures. Our results show that the maximum payload size is far more effective than the minimum one. Thus, we discard the min feature in the following identification experiments and reserve the max feature. The avg feature gets the third highest average mutual information values, and the value is a little lower than that of the $stdev$ feature. It means that the average payload size contains plenty of identification information, which makes the avg feature to be an effective statistical feature. As another mean value feature, the gm feature does not show such effectiveness as avg does. Its average mutual information value is the last but one, while far higher than that of the min feature. The $stdev$ feature is another effective statistical feature which gets the second highest average value.

Based on above analysis, we select six feature sets for the following identification experiments. The first one is the original payload size feature sets with 6 features, we call it $6ps$. Three pure statistical feature sets are selected as the second type, each of these feature sets only includes two statistical features, and these are $avg + stdev$, $gm + var$ and $max + var$. Finally, two hybrid feature sets which contain both original payload size and statistical features are selected, $4ps + avg + var$ includes the payload sizes of the 3rd-6th packets and $avg + var$, $6ps + avg + var$ includes all payload sizes and $avg + var$. We select the features with high mutual information values, and assemble them into the feature sets empirically. Therefore, there may exist more effective feature set, but it does not affect our experimental evaluations.

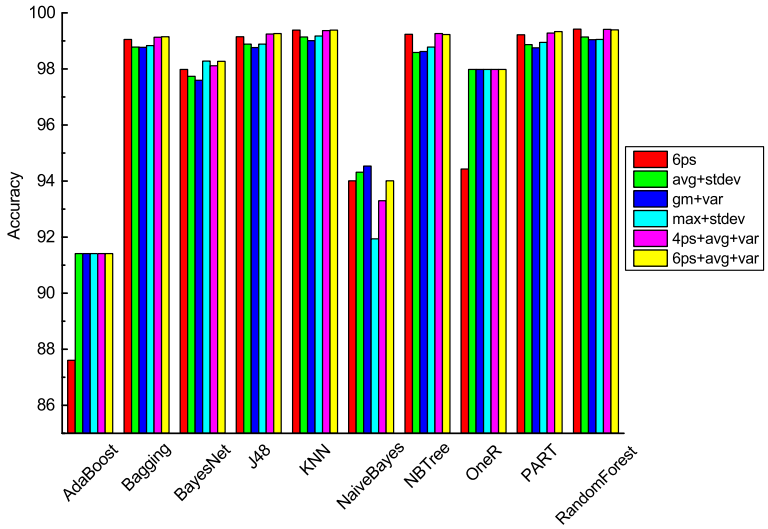

Fig. 4 Results of Auckland II data set

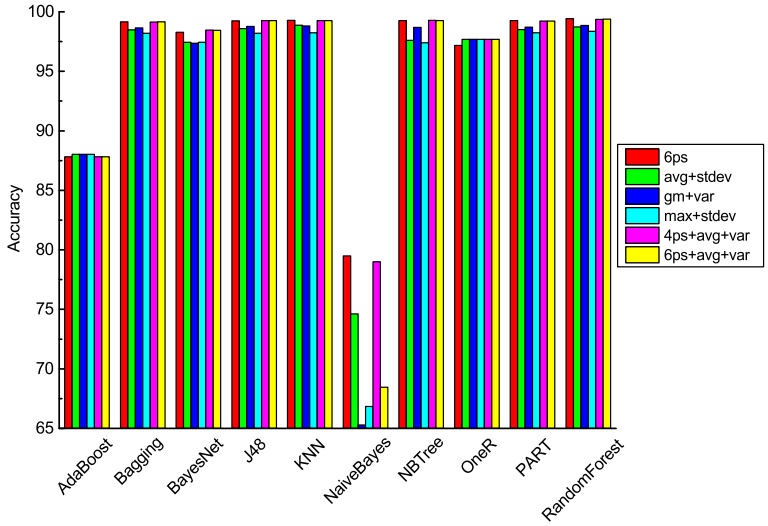

Fig. 5 Results of UNIBS data set

5.2 Identification Results

We show the identification results of each data set in a column chart (Figs. 3, 4, 5), and the detailed results can be found in Tables 4, 5 and 6 in Appendix.

Figure 4 shows the identification accuracies for the Auckland II data set. Most classifiers perform very well using all selected feature sets. The AdaBoost and NaiveBayes

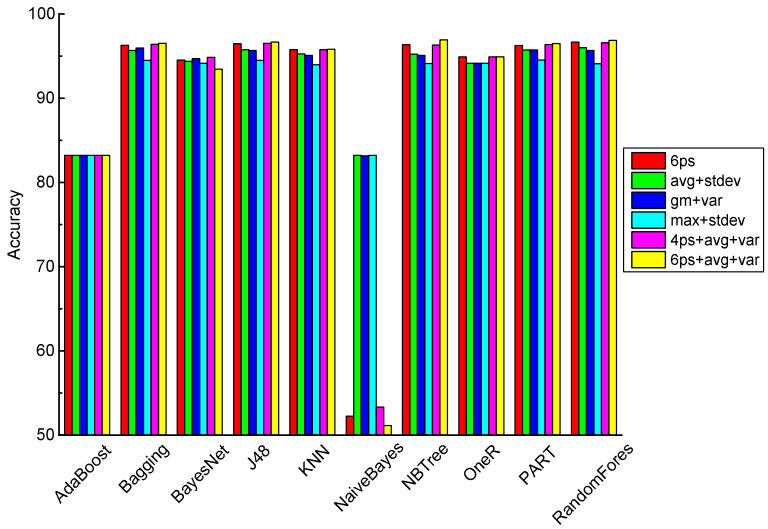

Fig. 6 Results of UJN data set

classifiers seem to be relatively weak. The hybrid feature set of $6ps + avg + var$ is the best performed feature set for this data set. It achieves the highest accuracy for 6 classifiers, and its average accuracy value is also the highest one among the results of all selected feature sets. It is easy to comprehend that the hybrid feature set contains the most of identification information. And the results on Auckland II data set testified this. However, for most classifiers, the differences between the accuracies of any two feature sets are not significant. All the three pure statistical feature sets ($avg + stdev$, $gm + var$, $max + var$) perform almost as well as the best performed feature set does. AdaBoost and NaiveBayes show unstable performances for different feature sets. Yet, it does not affect the evaluation of the effectiveness of the selected feature sets.

The results for UNIBS data set are shown in Fig. 5. Again, most classifiers get high identification accuracies greater than 97 %, AdaBoost and NaiveBayes do not perform so well as other classifiers do. All feature sets get identification accuracies with small differences for each classifiers except NaiveBayes, which is in accord with the circumstance of the Auckland II data set. The hybrid feature set of $4ps+avg+var$ achieves the highest average accuracy, which is very close to the value of the $6ps$ feature set. And the numbers of highest accuracies that the two feature sets get are also very close, which are 4 and 5 respectively.

The most significant characteristic of the results for UJN data set shown in Fig. 6 is that NaiveBayes gets far higher accuracies using the pure statistical feature sets than using the feature sets including the original payload size features. This makes that the average results of the three statistical feature sets (which range from 92.05 to 92.87 %) are obviously higher than that of the other three feature sets (which range from 90.20 to 90.43 %). While for other classifiers, the differences among the results are quite small.

5.3 Analysis and Discussions

Although the results of the three applied traffic data sets are different in detail, some lessons can be learned from the mutual information analysis and the identification results:

- Statistical features show high performances for early stage traffic identification as we expected. Most of the selected statistical features get high level of mutual information, and the statistical feature sets also get high accuracies in the identification experiments. From the point of view of mathematics, a statistical indicator usually shows a global view of a data sequence. Therefore it is able to represent a global feature of the sequence.
- An outstanding merit of statistical features is that it is possible to achieve high identification performances only using two statistical features. Each of the statistical feature sets we used only contains two selected features, and performs almost as well as the other feature sets do. It means that it is possible to express a traffic object exactly just using a few of its early stage global features.
- We have found accidentally in the study that the minimum is far less effective than the maximum. The reasons are not very hard to be discovered: the lower packet payload size limits of different applications are relatively fixed in a same range, while the upper limits usually vary with the applications. For example, the minimum payload sizes of a chat traffic and ftp traffic may both be in a range of 1–10 bytes, and the differences are not significant. However their maximum payload sizes are quite different: the chat traffic usually generates the maximum packet with several hundred bytes, while the maximum packet of the ftp traffic usually reaches the MTU size.
- The NaiveBayes classifier does not perform stably using different feature sets. The reason behind lies in the classification mechanism of the NaiveBayes classifier, and further discussions exceed the topic of this study. Although the performances of classifiers is not the main point of our study, the characteristic of the NaiveBayes classifier should be noticed when applying this classifier.

6 Conclusions

We have tried to evaluate the effectiveness of the statistical features for early stage traffic identification in this paper. We use both mutual information analysis and experimental methods for our study. Three traffic data sets include two opening data sets and 10 well-known classifiers are applied. According to the experimental results, we conclude that: as global features, most statistical features are as effective as the payload sizes do for early stage traffic identification. And high identification performances can be achieved by using few statistical features, our experimental results have shown this. However, not all statistical features are effective for early stage traffic identification. Our study shows that the minimum feature is not suited for identification. Furthermore, some original features such as the payload size of the first two packets in our study, are also not effective for identification. Thus, Features using for identification application should be carefully selected. How to select high effective feature sets for

early stage traffic identification is an important problem to be resolved in our future work.

Acknowledgments This research was partially supported by the National Natural Science Foundation of China under Grant Nos. 61472164, 61173078, 61203105, 61173079, and 61373054, the Provincial Natural Science Foundation of Shandong under Grant Nos. ZR2012FM010, ZR2011FZ001, ZR2013FL002 and ZR2012FQ016.

Appendix: Detailed Results of the Experimental Study

See Tables 3, 4, 5 and 6.

Table 3 Mutual information of all features (the best performed one of each column is shown in bold)

Features	Auckland II	UNIBS	UJN	Avg.
ps1	0.1248	0.0217	0.1204	0.0890
ps2	0.0945	0.1281	0.1249	0.1158
ps3	0.6705	0.1007	0.7701	0.5138
ps4	0.6907	0.9474	0.7507	0.7963
ps5	0.6090	0.4793	0.6646	0.5843
ps6	0.6825	0.6056	0.6326	0.6402
avg	0.7935	0.8786	0.7362	0.8028
stdev	0.8253	0.8739	0.7302	0.8098
min	0.0861	0.0316	0.2123	0.1100
max	0.7772	0.8171	0.6483	0.7475
gm	0.6833	0.6941	0.5742	0.6505
var	**0.9238**	**1.0387**	**0.8838**	**0.9488**

Table 4 Accuracy results for the Auckland II data set (the best performed one of each row is shown in bold)

Algorithms	6ps	avg + stdev	gm + var	max + stdev	4ps + avg + var	6ps + avg + var
NaiveBayes	94.01	94.32	**94.54**	91.94	93.30	94.01
BayesNet	97.98	97.73	97.60	**98.29**	98.11	98.27
AdaBoost	87.61	**91.41**	**91.41**	**91.41**	**91.41**	**91.41**
Bagging	99.05	98.78	98.77	98.83	99.13	**99.15**
OneR	94.43	**97.98**	**97.98**	**97.98**	**97.98**	**97.98**
PART	99.22	98.87	98.75	98.94	99.28	**99.34**
KNN	99.38	99.14	99.01	99.17	99.37	**99.39**
J48	99.15	98.88	98.76	98.89	99.24	**99.26**
RandomForest	**99.42**	99.14	99.04	99.05	**99.42**	99.40
NBTree	99.24	98.59	98.62	98.78	**99.26**	99.23
Avg.	96.95	97.48	97.45	97.33	97.65	**97.74**

Table 5 Accuracy results for the UNIBS data set (the best performed one of each row is shown in bold)

Algorithms	6ps	avg + stdev	gm + var	max + stdev	4ps + avg + var	6ps + avg + var
NaiveBayes	**79.49**	74.61	65.29	66.84	79.00	68.46
BayesNet	98.29	97.45	97.37	97.44	**98.47**	98.44
AdaBoost	87.82	**88.02**	**88.02**	**88.02**	87.82	87.82
Bagging	**99.16**	98.48	98.64	98.19	99.13	**99.16**
OneR	97.17	**97.68**	**97.68**	**97.68**	**97.68**	**97.68**
PART	**99.27**	98.50	98.72	98.25	99.23	99.22
KNN	**99.28**	98.87	98.82	98.25	99.26	99.27
J48	99.24	98.60	98.77	98.20	**99.27**	99.25
RandomForest	**99.42**	98.73	98.85	98.37	99.37	99.39
NBTree	99.26	97.61	98.69	97.39	**99.29**	99.27
Avg.	95.84	94.85	94.08	93.86	**95.85**	94.80

Table 6 Accuracy results for the UJN data set (the best performed one of each row is shown in bold)

Algorithms	6ps	avg + stdev	gm + var	max + stdev	4ps + avg + var	6ps + avg + var
NaiveBayes	52.26	**83.23**	83.15	**83.23**	53.35	51.15
BayesNet	94.54	94.40	94.70	94.15	**94.86**	93.45
AdaBoost	**83.23**	**83.23**	**83.23**	**83.23**	**83.23**	**83.23**
Bagging	96.29	95.69	95.97	94.52	96.41	**96.51**
OneR	**94.90**	94.15	94.15	94.15	**94.90**	**94.90**
PART	96.25	95.73	95.73	94.54	96.37	**96.49**
KNN	95.77	95.26	95.10	93.97	95.77	**95.83**
J48	96.47	95.77	95.69	94.50	96.51	**96.67**
RandomForest	96.65	96.01	95.69	94.09	96.59	**96.88**
NBTree	96.33	95.24	95.08	94.11	96.31	**96.92**
Avg.	90.27	**92.87**	92.85	92.05	90.43	90.20

References

1. Bernaille, L., Teixeira, R., Akodkenou, I., Soule, A., Salamatian, K.: Traffic classification on the fly. In: ACM SIGCOMM'06, pp. 23–26 (2006)
2. Bahl, L.B., de Souza, P., Mercer, R.P., et al.: Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'86), pp. 49–52, IEEE Press (1986)
3. Breiman, L.: Bagging predictors. Mac. Learn. **24**, 123–140 (1996)
4. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
5. Dainotti, A., Pescapé, A., Claffy, K.C.: Issues and future directions in traffic classification. IEEE Netw. **26**(1), 35–40 (2012)
6. Dainotti, A., Pescapé, A., Sansone, C.: Early classification of network traffic through multi-classification. Lect. Notes Comput. Sci. **6613**, 122–135 (2011)
7. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Mach. Learn. **29**, 103–137 (1997)

8. Estan, C., Varghese, G.: New directions in traffic measurement and accounting: focusing on the elephants, ignoring the mice. ACM Trans. Comput. Syst. **21**(3), 270–313 (2003)
9. Este, A., Gringoli, F., Salgarelli, L.: On the stability of the information carried by traffic flow features at the packet level. In: ACM SIGCOMM'09, pp. 13–18 (2009)
10. Este, A., Gringoli, F., Salgarelli, L.: Support vector machines for TCP traffic classification. Comput. Netw. **53**, 2476–2490 (2009)
11. Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. In: The Fifteenth International Conference on Machine Learning, pp. 144–151. IEEE Press (1998)
12. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Mach. Learn. **29**(2–3), 131–163 (1997)
13. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
14. Holte, R.C.: Very simple classification rules perform well on most commonly used datasets. Mach. Learn. **11**(1), 63–90 (1993)
15. Huang, N., Jai, G., Chao, H.: Early identifying application traffic with application characteristics. In: IEEE International Conference on Communications (ICC'08). pp. 5788–5792 (2008)
16. Huang, N., Jai, G., Chao, H., et al.: Application traffic classification at the early stage by characterizing application rounds. Inf. Sci. **232**(20), 130–142 (2013)
17. Hullár, B., Laki, S., Gyorgy, A.: Early identification of peer-to-peer traffic. In: 2011 IEEE International Conference on Communications (ICC), pp. 1–6. IEEE Press (2011)
18. Gringoli, F., Salgarelli, L., Dusi, M., et al.: Gt: picking up the truth from the ground for internet traffic. ACM SIGCOMM Comput. Commun. Rev. **39**(5), 12–18 (2009)
19. Kohavi, R.: Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. In: The Second International Conference on Knowledge Discovery and Data Mining (KDD), pp. 202–207. IEEE Press (1996)
20. Li, W., Moore, A.W.: A machine learning approach for efficient traffic classification. In: Proceedings of IEEE MASCOTS'07, pp. 310–317 (2007)
21. Maes, F., Collignon, A., Vandermeulen, D., et al.: Multimodality image registration by maximization of mutual information. IEEE Trans. Med. Imaging **16**(2), 187–198 (1997)
22. Maron, M.E.: Automatic indexing: an experimental inquiry. J. ACM **8**(3), 404–417 (1961)
23. Moore, A.W., Zuev, D., Crogan, M.: Discriminators for use in flow-based classification. Intel Research Tech. Rep (2005)
24. Moore, A.W., Zuev, D.: Internet traffic classification using Bayesian analysis techniques. In: ACM SIGMETRICS'05, pp. 50–60 (2005)
25. Nguyen, T.T.T., Armitage, G.: A survey of techniques for internet traffic classification using machine learning. IEEE Commun. Surv. Tutor. **10**(4), 56–76 (2008)
26. Nguyen, T.T.T., Armitage, G., Branch, P., et al.: Timely and continuous machine-learning-based classification for interactive IP traffic. IEEE/ACM Trans. Netw. **20**(6), 1880–1894 (2012)
27. Peng, H.: Mutual infomation Matlab toolbox, http://www.mathworks.com/matlabcentral/fileexchange/14888-mutual-information-computation
28. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal. Mach. Intell. **27**(8), 1226–1238 (2005)
29. Peng, L., Zhang, H., Yang, B., et al.: Traffic labeller: collecting internet traffic samples with accurate application information. China Commun. **11**(1), 67–78 (2014)
30. Qu, B., Zhang, Z., Guo, L., et al.: On accuracy of early traffic classification. In: IEEE 7th International Conference on Networking, Architecture and Storage (NAS), pp. 348–354. IEEE Press (2012)
31. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kauffman, Los Altos (1993)
32. Rizzi, A., Colabrese, S., Baiocchi, A.: Low complexity, high performance neuro-fuzzy system for internet traffic flows early classification. In: 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 77–82. IEEE Press (2013)
33. Svetnik, V., Liaw, A., Tong, C., et al.: Random forest: a classification and regression tool for compound classification and QSAR modeling. J. Chem. Inf. Comput. Sci. **43**(6), 1947–1958 (2003)
34. Tcpdump/Libpcap. http://www.tcpdump.org
35. UNIBS: Data sharing. http://www.ing.unibs.it/ntw/tools/traces/
36. Waikato Internet Traffic Storage (WITS). http://www.wand.net.nz/wits
37. Weka 3: Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka/

38. Zander, S., Nguyen, T.T.T., Armitage, G.: Automated traffic classification and application identification using machine learning. In: IEEE Conference on Local Computer Networks 30th Anniversary, IEEE Press (2005)