Robust Network Traffic Identification with Unknown Applications

Jun Zhang
School of Information Technology
Deakin University, Australia
jun.zhang@deakin.edu.au

Chao Chen
School of Information Technology
Deakin University, Australia
zvm@deakin.edu.au

Yang Xiang
School of Information Technology
Deakin University, Australia
yang.xiang@deakin.edu.au

Wanlei Zhou
School of Information Technology
Deakin University, Australia
wanlei@deakin.edu.au

ABSTRACT

Traffic classification is a fundamental component in advanced network management and security. Recent research has achieved certain success in the application of machine learning techniques into flow statistical feature based approach. However, most of flow statistical feature based methods classify traffic based on the assumption that all traffic flows are generated by the known applications. Considering the pervasive unknown applications in the real world environment, this assumption does not hold. In this paper, we cast unknown applications as a specific classification problem with insufficient negative training data and address it by proposing a binary classifier based framework. An iterative method is proposed to extract unknown information from a set of unlabelled traffic flows, which combines asymmetric bagging and flow correlation to guarantee the purity of extracted negatives. A binary classifier is used as an application signature which can operate on a bag of correlated flows instead of individual flows to further improve its effectiveness. We carry out a series of experiments in a real-world network traffic dataset to evaluate the proposed methods. The results show that the proposed method significantly outperforms the-state-of-art traffic classification methods under the situation of unknown applications present.

Categories and Subject Descriptors

C.2.3 [**COMPUTER-COMMUNICATION NETWORKS**]: Network Operations

Keywords

Network Security; Traffic Classification; Unknown Traffic

ASIA CCS'13, May 8–10, 2013, Hangzhou, China.

1. INTRODUCTION

As a basic technique for securing modern information infrastructure, traffic classification aims to identify the generating application/protocol of any network traffic in the complex environment [18, 4, 11]. For instance, identifying network traffic is an essential requirement in full deployment of QoS control and intrusion detection [21, 10]. Traditional traffic classification methods rely on checking IP port numbers since early well-known applications always use own unique ports for transmitting IP packets. In the last decade, many applications employ dynamic ports to evade port-based traffic identification. Taking this problem into account, current industry products apply deep inspection to traffic classification that checks the applications' signature strings in the payload of IP packets [14]. While the payload-based methods are accurate, they fail to handle encrypted applications and protect user privacy. Recent research in the area tends to investigate flow (i.e., successive IP packets) statistical features which can be extracted from IP headers without deep inspection. Substantial attention has been paid on the application of machine learning techniques into statistical feature based traffic classification [18].

Traffic classification using flow statistical features is conventionally considered as a multi-class classification problem and addressed by using supervised and unsupervised machine learning algorithms. In supervised traffic classification [16, 1, 2, 5, 9], a classification model is learned by using the labelled training samples from each predefined traffic class. By contrast, the unsupervised (clustering) methods [23, 7, 3] automatically group a set of unlabelled training samples and apply the clustering results to construct a traffic classifier with the assistance of other tools such as payload-based software. In the multi-class framework, it is assumed that any testing flow comes from a predefined traffic class. While these multi-class classification methods have reported good performance, they cannot effectively handle the emerging applications which are unknown to the traffic classification system.

In another point of view, statistical feature based traffic classification can break down to a series of detection problems by revisiting the concept of application signature. In payload based traffic classification, any testing flow is inspected to determine whether it contains the signature of a known application. If an application signature is found, the flow is categorized to the application-based traffic class. Otherwise, the flow does not belong to the class. When the flow does not contain the signature of any known application, it is labelled as unknown. Based on the signatures of known applications, we can see that any traffic flows of either known or unknown applications can be identified. In the same way, once any known application has a statistical feature based signature, all flows generated by known and unknown applications can be dealt with straightforward. However, few work has contributed to propose a statistical feature based signature including one-class support vector machine (SVM) [9] and normalized threshold [6]. Moreover, one-class SVM classifier usually suffers from poor decision boundary without the negative information and the normalized threshold method is heuristic.

Naturally, we can cast signature construction using statistical feature as a binary classification problem. For a target known application, the positive class consists of the flows generated by the application and the negative class is created by all other flows. To obtain an accurate classifier, we need the sufficient positive and negative training samples. It is generally accepted that the sufficient training samples for known applications can be easily obtained. Considering the specific binary classification problem, we can have a sufficient positive training set, but the negative training set is always insufficient since unknown applications present. Note that it will lead to a biased classifier by simply using the training samples of other known applications as the negatives. The problem of insufficient negative training set has not been addressed in previous work [9, 6] on statistical feature based signature.

This paper is aimed to achieve robust network traffic classification with unknown applications, which relaxes the unrealistic assumption that all classes are known to the classifier. The following lists the major contributions.

- Considering unknown applications, we formulate statistical feature based signature construction as a specific binary classification problem with insufficient negative samples.
- To solve this problem, we develop a generic approach to extract unknown application information from a set of unlabelled traffic flows and propose to incorporate asymmetric bagging and flow correlation to guarantee the purity.
- We propose to use a binary classifier as the application signature which can operate on a bag of correlated flows instead of individual flows.

For performance evaluation, a series of experiments are carried out in two real-world network traffic datasets. The results show that the binary classifier based signature is superior to the-state-of-art traffic classification methods when unknown applications present.

The rest of this paper is organized as follows. Section 2 states the research problem by analysing the existing methods using flow statistical features. In Section 3, a new method for unknown information extraction is proposed, which is followed by a binary classifier based application signature in Section 4. Section 5 reports the experiments and results. Finally, Section 6 concludes this paper.

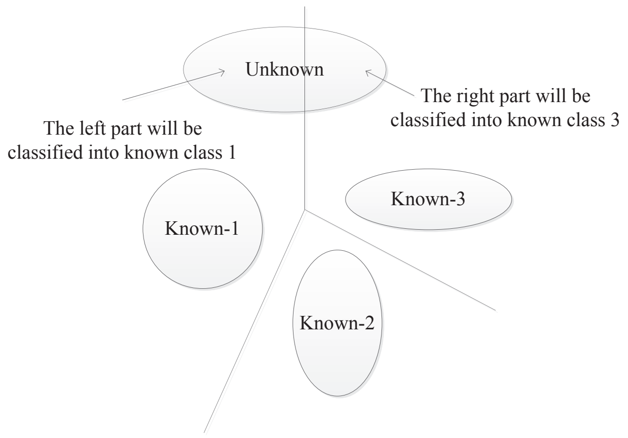

Unknown
The left part will be classified into known class 1
The right part will be classified into known class 3
Known-1
Known-3
Known-2

Figure 1: Problem of unknown application

2. PROBLEM STATEMENT AND ANALYSIS

What is the problem of traffic classification with unknown applications and why is it difficult to solve? In this section, we answer this question by providing a critical analysis on several typical methods using flow statistical features.

Considering a real-world network scenario, the traffic dataset, Ω, consists of K known classes and U unknown classes, $\Omega = \{\omega_1, ..., \omega_K, \overline{\omega}_1, ..., \overline{\omega}_U\}$. A known class ω_k is corresponding to an application which is well known by the traffic classification system. In this paper, it means that a set of labelled flow samples, ψ_k, is available for a known class, ω_k. By contrast, an unknown class is related to an unknown application of the system and no labelled flow samples are available. This scenario is common since a lot of new applications are emerging every day on Internet. Given the labelled flow samples $\{\psi_1, ..., \psi_K\}$, the problem is how to identify the class of the flows in Ω. In this paper, a flow consists of successive IP packets with the same 5-tuple: *source_ip*, *source_port*, *destination_ip*, *destination_port*, *transport_protocol*. A number of statistical features, such as number of IP packets, are used to represent a flow $\mathbf{x}$ for traffic classification.

Conventional traffic classification methods address a K-class classification problem, which is under the multi-class framework and cannot deal with unknown classes [16, 1, 2, 5, 9, 23, 7, 3, 12, 15, 17]. These methods use the labelled flow samples to form a training set straightforward, $T = \{\psi_1, ..., \psi_K\}$, and employ a learning algorithm to seek a classification model. The classifier trained by using T will classify any flows into a known class. Thus, the flows in the unknown classes, $\{\overline{\omega}_1, ..., \overline{\omega}_U\}$, will be inaccurately classified into K known classes. Traditional pattern classification algorithms assume that all testing flows come from the known (predefined) classes, so they all suffer from this issue. Fig. 1 illustrates the problem of unknown application class under the multi-class framework. We take the classification algorithm using decision boundary as an example. One can see that the optimized decision boundary in the feature space can effectively separate the known classes, but it will inaccurately classify the flows of unknown class into known class 1 and known class 3.

A semi-supervised method [8] was proposed to address a $(K+1)$-class classification problem, which extended the multi-class framework by taking unknown applications into account. Firstly, some mixture of labelled and unlabelled training samples are grouped into k clusters by using traditional clustering algorithms such as k-means. Then, k clusters are mapped to $\omega_1, ..., \omega_K$, or unknown according to the locations of the labelled (supervised) training samples. For traffic classification, a flow will be predicted to the class of its nearest cluster. Although this method demonstrates the capability of dealing with unknown applications, it is heuristic in nature, e.g., the number of clusters, k, is manually adjusted. Moreover, a large k is necessary for generating high-purity clusters, while it will lead to many false unknown clusters.

In contrast to the perspective of multi-class classification, the signature-based approach offers another view for traffic classification. By using statistical feature based signature, any testing flow can be determined whether it belongs to a known class. If the flow does not belong to any known class, it is classified as unknown. The early work on normalized threshold [6] for statistical feature based signature is heuristic, which does not investigate the information of unknown applications. One-class SVM [9] can be considered as another type of signature. For a known class ω_k, the training samples in ψ_k are used to learn a one-class SVM and other training samples in $\bigcup_{i=1, i\neq k}^{i=K} \psi_i$ are used to adjust the decision boundary. There are two issues: firstly, one-class SVM normally needs a large number of training samples and the modified method cannot outperform a traditional two-class SVM; secondly, the decision boundary is still poor due to the lake of the information about unknown classes, $\{\overline{\omega}_1, ..., \overline{\omega}_U\}$.

The above analysis motives us to create statistical feature based signatures by taking unknown applications into account. Since the signature based decision is binary, 'Yes' or 'No', we cast signature construction as a binary classification problem. However, this binary classification problem is special because only a partial negative training set is available in a real world scenario. In detail, for class ω_k, the positive training set is $T_P = \psi_k$ and the partial negative training set is $T_K = \bigcup_{i=1, i\neq k}^{i=K} \psi_i$. If we use T_P and T_K to train a classifier C_k, the decision boundary of C_k will be biased towards the negative class in the feature space. In other words, some negative samples from $\{\overline{\omega}_1, ..., \overline{\omega}_U\}$ will be inaccurately classified to the positive class. The reason is that a sufficient negative training set should include the information of unknown classes, while T_K does not. Therefore, the key problem in our work is how to obtain the information of unknown classes, $\{\overline{\omega}_1, ..., \overline{\omega}_U\}$.

Inspired by the semi-supervised methods [8], we realize that the samples randomly collected from the target network are unlabelled, but they must contain traffic flows generated by unknown applications. Formally, we can obtain a set of unlabelled samples, $\Omega_r \subset \Omega$. Then, the key problem becomes how to extract the examples of unknown classes from Ω_r. Once some flows from unknown classes are identified, they can be combined with T_K to form a better negative training set, T_N. Then, T_P and T_N can be used to learn an accurate binary classifier which is considered as the signature of ω_k

3. UNKNOWN INFORMATION EXTRACTION

This section presents our generic method of unknown information extraction. The proposed method is able to extract the negative samples from a large number of unlabelled network traffic during an iterative process. Moreover, two new techniques, asymmetric bagging and flow correlation, are applied to improve the effectiveness of the iterative extraction.

```
input : positive trainng set $T_P$; partial negative
        training set $T_K$; unlabelled data set $\Omega_r$; a
        binary classification algorithm $\Phi$
output: sufficient negative training set $T_N$
$T_N \leftarrow T_K$;
// create output flow set
Use $\Phi$ to create a classifier $C_0$ from $T_P$ and $T_K$;
// the biased classifier $C_0$ will produce
   many false positives
Classify $\Omega_r$ by $C_0$;
Put positive samples classified by $C_0$ into $P_1$;
Put negative samples classified by $C_0$ into $N_1$;
$i = 1$
while $N_i \neq \emptyset$ do
    $T_N = T_N \cup N_i$;
    Use $\Phi$ to create a classifier $C_i$ from $T_P$ and $T_N$;
    Classify $P_i$ by $C_i$;
    Put positive samples classified by $C_i$ into $P_{i+1}$;
    Put negative samples classified by $C_i$ into $N_{i+1}$;
    $i = i + 1$;
end
Return $T_N$;
```

Algorithm 1: Unknown sample extraction

3.1 An Iterative Extraction Method

This method aims to identify the negative samples as much as possible during an iterative process from the unlabelled data which contains universal traffic flows of known and unknown classes. The detailed process is listed in Algorithm 1. We take the class ω_k as an example to illustrate the proposed method. For the binary classification, the positive class is ω_k and the negative class includes other known classes $\Omega'_K = \{\omega_2, ..., \omega_K\}$ and all unknown classes $\overline{\Omega}_U = \{\overline{\omega}_1, ..., \overline{\omega}_U\}$.

To start this process, we need an initial binary classifier given $T_P = \psi_k$ and $T_N = T_K = \bigcup_{i=1, i\neq k}^{i=K} \psi_i$. As mentioned before, if we use the classification algorithm Φ to create a classifier from T_P and T_K, the biased classifier will inaccurately classify some negative samples as positive. However, in the other hand, it definitely can produce some new negative samples which may be from some known or unknown classes. Based on this observation, we can construct an initial binary classifier C_0 from T_P and T_K.

$$C_0 = \Phi(T_P, T_K), \tag{1}$$

Then, C_0 is used to classify the unlabelled data set Ω_r, which produces a positive sample set P_1 and a negative sample set N_1.

$$C_0 : \ \Omega_r = P_1 \cup N_1. \tag{2}$$

In the training data, T_P is sufficient, while T_K is insufficient without the samples for $\{\overline{\omega}_1, ..., \overline{\omega}_U\}$. Therefore, the decision boundary of C_0 will be biased towards the negative class. Ideally, P_1 will consist of the flows of ω_k and many flows of unknown classes $\overline{\Omega}_U$. N_1 will contain the flows of Ω'_K and some flows of unknown classes $\overline{\Omega}_U$. We can see that the negative samples N_1 classified by C_0 could be high-purity and contains the information of unknown classes. Thus, N_1 can be complementary to the negative training set T_N. In addition, P_1 includes many flows of unknown classes, which motivates us to further extract the unknown information from P_1.

Iteratively, we can update the negative training set T_N by adding a high-purity negative set N_i and train a more accurate classifier C_i from T_P and T_N. In fact, a series of binary classifiers push the decision boundary back to the real borderline of the positive class step by step. The stop criteria is $N_i = \emptyset$, then T_N becomes steady. Finally, we have

$$T_N = T_K \cup N_1 ... \cup N_i,\ (where\ N_i = \emptyset). \tag{3}$$

This iterative process has the capability of effectively extract unknown information from a large number of unlabelled network traffic and significantly improve the negative training set.

The proposed method can employ most traditional classification algorithms such as SVM and random forest for implementation. A similar method [22] has been proposed for web page classification without negative examples, which combines two algorithms: 1-DNF for constructing the initial classifier and SVM for the iterative process. By considering the traffic classification situation, 1-DNF is difficult to incorporate partial negative training samples. We need a new classifier to ensure the high-purity of classified negative flows and thus to extract unknown information accurately. Therefore, we propose to apply two techniques, asymmetric bagging and flow correlation, to construct a new classifier for unknown information extraction.

3.2 Asymmetric Bagging

The purity of extracted negative samples is critical to the performance of the final classifier used for testing. However, we observed that simply applying a traditional classifier in the iterative method cannot guarantee the high purity of obtained new negative samples. On the one hand, the insufficient negative training set will let the decision boundary be biased towards the negative class. On the other hand, the size of the negative training set is much larger than that of the positive training set. This imbalance will push the decision boundary towards the positive class. In short, the imperfect decision boundary cannot avoid false negative, i.e., classifying real positives as negatives, so the extracted negative set is not pure.

We propose to apply asymmetric bagging [19, 25] to help address the problem of low-purity negative samples. The idea is to train multiple classifiers using different negative samples and combine the results of these classifiers. Each time we randomly select a subset of negative samples with the same size to the positive training set. Then, we combine the selected negative training set with the positive training set to train a classifier.

$$C_i = \Phi(T_P, T_{Ni}),\ T_{Ni} \subset T_N\ \&\ \|T_{Ni}\| = \|T_P\|. \tag{4}$$

Once the classifier is ready, we use it to classify all testing flows. Therefore, we can train multiple classifiers to conduct classification and obtain multiple class labels for each flow. Finally, we combine the multiple predicted labels of each flow to make a final decision. For example, random forest can be used to train a classifier and the majority vote rule can be used to combine classification results.

$$\begin{array}{l} assign\ \mathbf{x} \longrightarrow \omega_k\ if \\ \sum_{i=1}^{i=A} C_i(\mathbf{x}) > \frac{A}{2}, \end{array} \tag{5}$$

where $C_i(\mathbf{x}) = 1$ means flow $\mathbf{x}$ is classified to ω_k. $C_i(\mathbf{x}) = 0$ denotes that flow $\mathbf{x}$ is rejected by ω_k.

In the asymmetric bagging-based training process, the negative and positive training sets have the same size, which can avoid the unbalance problem. The decision boundary of a single classifier will be biased towards the negative class. Classifier combination can be helpful to improve the final classification accuracy. In this way, we can guarantee that the purity of the extracted negative samples is high.

3.3 Flow Correlation

Due to unknown applications, the decision boundary of a classifier trained by using insufficient negative samples could not be optimal, which will affect the quality of extracted unknown information. The previous work [24] showed that flow correlation can improve the traffic classification performance with an insufficient training set. Since flow correlation focuses on investigating the relationship among real world data, it can be applied to either known or unknown applications.

We propose to incorporate flow correlation into the asymmetric bagging-based classification process in order to further improve the effectiveness of unknown information extraction. We use the 3-tuple heuristic to determine correlated flows which are modelled by "bag of flows" (BoF).

- 3-tuple heuristic: in a certain period of time, the flows sharing the same 3-tuple {source_ip, destination_ip, transport_protocol}form a BoF.

For example, several flows initiated by different hosts are all connecting to a same host at TCP port 80 in a short period. These flows are very likely generated by the same application such as a web browser. In general, we can apply 3-tuple heuristic to determine BoFs, but it needs to answer which classes the BoFs belong to. With flow correlation, an aggregation classifier aims to classify BoFs instead of individual flows. Given a BoF $X = \{\mathbf{x}_1, ..., \mathbf{x}_B\}$, we aggregate the results of B flows produced by a binary classifier C to predict the class label of X. In this paper, we also employ the majority vote rule to perform flow prediction aggregation due to its good performance. A BoF X can be classified to class ω_k only if

$$\sum_{\mathbf{x} \in X} C(\mathbf{x}) > \|X\|/2. \tag{6}$$

Algorithm 2 shows the ensemble classification with asymmetric bagging and flow correlation.

4. APPLICATION SIGNATURE WITH FLOW CORRELATION

This section presents the new approach to construct application signatures with flow correlation (ASFC). Moreover, the technical justification is also provided to confirm the effectiveness of the proposed approach.

4.1 Proposed Approach: ASFC

Taking unknown applications into account, this work focus on developing a new application signature based traffic

```
input  : positive trainng set $T_P$; negative training
         set $T_N$; testing set $\Omega_t$; 3-tuple information
         set $\Omega_3$; a binary classification algorithm $\Phi$;
         number of asymmetric bagging classifier $A$
output: testing class label set $L_t$
Construct BoFs $\mathbf{X} = \{X_j\}$ by running 3-tuple
heuristic on $\Omega_3$;
for $i \leftarrow 1$ to $A$ do
    Obtain $T_{Ni}$ by random sampling in $T_N$;
    // $T_{Ni}$ has the same size to $T_P$
    Use $Phi$ to create $C_i$ from $T_P$ and $T_{Ni}$
    for $j \leftarrow 1$ to $\|\boldsymbol{X}\|$ do
        for $k \leftarrow 1$ to $\|X_j\|$ do
            Classify $\mathbf{x}_{jk}$ by $C_i$;
        end
        Use majority vote rule to aggregate the
        predictions of flows in BoF $X_j$.
    end
end
Obtain $L_t$ by combining the predictions of BoFs
produced by $A$ asymmetric bagging classifier
$\{C_1, ..., C_A\}$;
Return $L_t$
```

Algorithm 2: Ensemble classification

classification. Section 3 has addressed the key problem of unknown information extraction. Another problem is how to construct the signature of each application using the available training data.

In this paper, we create a binary classifier and use it as the application signature in a fast way. Given the positive and negative training sets, we can train a binary classifier for an application. If a testing flow is classified to positive, it means this flow comes from the application. Otherwise, the testing flow does not belong to the application. With the consideration of flow correlation, we perform an ensemble classification on BoFs instead of individual flows. Then, the accuracy of the application signature can be improved by inspecting a bag of correlated flows simultaneously .

We summarize the proposed approach for signature based traffic classification as follows. For convenience, we take an application as example.

1. Collect a set of unlabelled traffic flows Ω_r from the target network traffic;

2. Extract a complete negative training set T_N from Ω_r using Algorithm 1;

3. Use T_P and T_N to train a binary classifier as the application signature;

4. Perform BoF-based classification according to Section 3.3.

Any traditional binary classification algorithm can be used in this approach. In our experiments, we select random forest as the basic classifier for our approach, but other algorithms are also tested.

4.2 Technique Justification

The proposed approach relies on the techniques of unknown information extraction. The critical problem is that the system does not have any training samples for unknown applications. The simple classifier by using the training samples of known applications cannot deal with the flows generated by unknown applications. The method of unknown information extraction was proposed to extract samples of unknown applications from a set of unlabelled network traffic. In unknown information extraction, asymmetric bagging balanced the positive and negative training sets which avoids decision boundary moving towards the positive class, so as to guarantee the purity of extracted negative samples. We further applied flow correlation in asymmetric bagging based classification, which can effectively increase the amount of extracted negative samples.

Moreover, the final ensemble classification process can improve the classification accuracy by investigating the relationship in real world data. Suppose $\varphi(\mathbf{x}, L)$ is a simple classifier (predictor) and L is the training set. The aggregation can be described as $\varphi_A(X, L) = E_{\mathbf{x}\in X}\varphi(\mathbf{x}, L)$. Let y be the class label of a flow $\mathbf{x}$ which belongs to a BoF X. Both y and $\mathbf{x}$ are random variables which are drawn from the distribution independent of the training set L. The average classification error on BoFs, estimated by the simple predictor $\varphi(\mathbf{x}, L)$, is $E_{y,\mathbf{x}\in X}(y - \varphi(\mathbf{x}, L))^2$. The corresponding classification error estimated by the aggregated predictor is $E_{y,\mathbf{x}\in X}(y - \varphi_A(X, L))^2$. Since

$$E_{y,\mathbf{x}\in X}\varphi^2(\mathbf{x}, L) \geq (E_{y,\mathbf{x}\in X}\varphi(\mathbf{x}, L))^2, \tag{7}$$

after some modification, we obtain

$$E_{y,\mathbf{x}\in X}(y - \varphi(\mathbf{x}, L))^2 \geq E_{y,\mathbf{x}\in X}(y - \varphi_A(X, L))^2. \tag{8}$$

This shows flow correlation can further improve the final classification performance.

Let us take the majority vote rule as an example for flow prediction aggregation. Suppose a basic classifier with an error rate of $p < 0.5$, it means the classifier is better than random guessing. Since the flows in a BoF are diverse, the classifier will make different errors in predicting the class of different flows. In the binary classification problem, for the majority vote to be incorrect, it requires that $\|X\|/2$ or more flows in X are classified incorrectly. The probability that r flows are classified incorrectly is

$$C_r^g \cdot p^r(1-p)^{g-r} = \frac{g!}{r!(g-r)!}p^r(1-p)^{g-r} \tag{9}$$

Therefore, the probability that the majority vote is incorrect is

$$P(error) = \sum_{r=\lfloor(\|X\|/2+1)/2\rfloor}^{\|X\|/2} \frac{g!}{r!(g-r)!}p^r(1-p)^{g-r}. \tag{10}$$

For example, given a BoF with $g = 30$ and the basic classifier with an error rate $p = 0.3$, the probability of the majority vote being incorrect is 0.006, which is much less than the individual classification error rate. In general, we have $P(error) < p$, so the classification accuracy can be improved by classifying BoFs with the majority vote rule.

5. PERFORMANCE EVALUATION

We carried out a series of network traffic identification experiments to evaluate the proposed ASFC approach. It was aimed to answer an important question, why the proposed ASFC approach is superior.

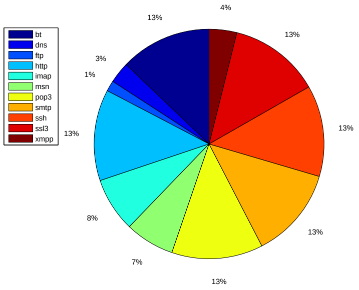


Figure 2: Dataset Distribution

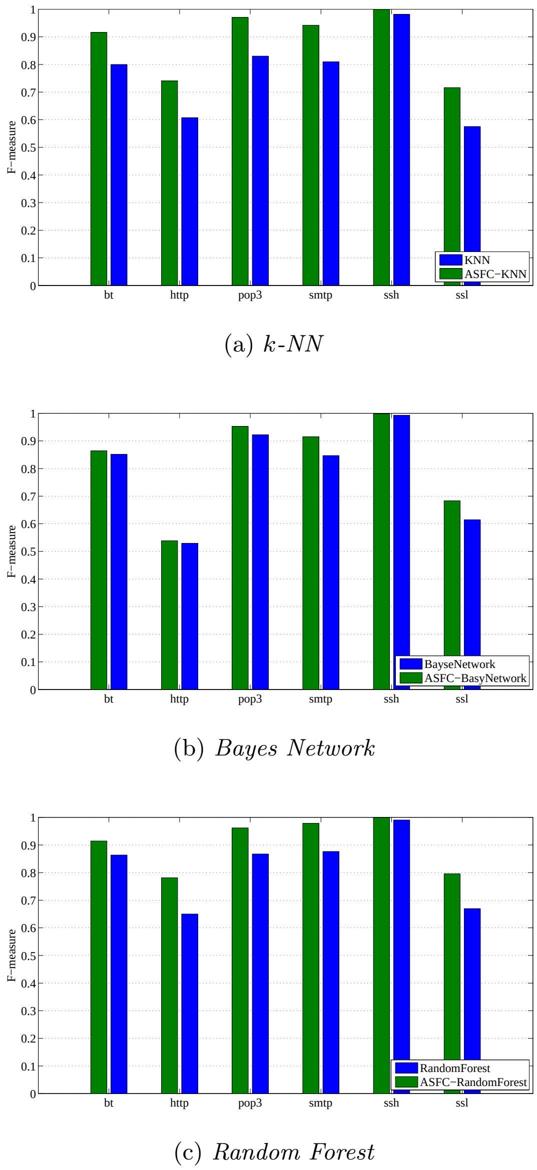
Table 1: Unidirectional statistical features

Type	Unidirection features	No.
Packets	Number of packets transferred	2
Bytes	Volume of bytes transferred	2
Packet Size	Min., Max., Mean and Std Dev. of packet size	8
Inter-Packet Time	Min., Max., Mean and Std Dev. of Inter Packet Time	8
	Total	20

5.1 Datasets and Experiments

A real-world dataset was used to evaluate our new techniques in this work. The *isp* dataset is a full payload traffic dataset collected at a medium sized ISP in Australia [20]. In order to build ground truth, we have developed a deep payload inspection tool (DPI) which uses string signature of packets' payload to accurately identify the class of traffic flows. A number of application signatures were developed based on the previous experience and some open source tools like l7-filter (http://l7-filter.sourceforge.net) and Tstat (http://tstat.tlc.polito.it). For some encrypted or new applications, we inspected the flows manually to determine the class of flows. Finally, the *isp* dataset is constituted by over 80,000 traffic flows from 11 application-oriented classes. Figure 2 shows the distribution of traffic classes in the dataset. In the experiments, 20 unidirectional flow statistical features were extracted to represent traffic flows, which are listed in Table 1. We applied feature selection to further remove irrelevant and redundant features from the feature set [13]. The process of feature selection yielded 6 features for the isp dataset.

In the experiments, each dataset was separated into three parts, one for training, one for unlabelled, and the other for testing. The training, unlabelled, and testing parts have 25%, 25%, and 50% of traffic flows in the dataset, respectively. To simulate the problem unknown applications, the idea is to set several small classes to "unknown". In detail, the classes of DNS, FTP, IMAP, MSN, XMPP in the *isp* dataset are set as unknown. Therefore, the modified isp dataset consists of 6 known classes and 5 unknown classes. All unknown classes includes about 22% of traffic flows in the dataset. For each known class, 1,000 flow samples are randomly selected from the training part to form a supervised training set. It is important to note that no any samples of unknown classes are available for the classification system.



(a) *k-NN*

(b) *Bayes Network*

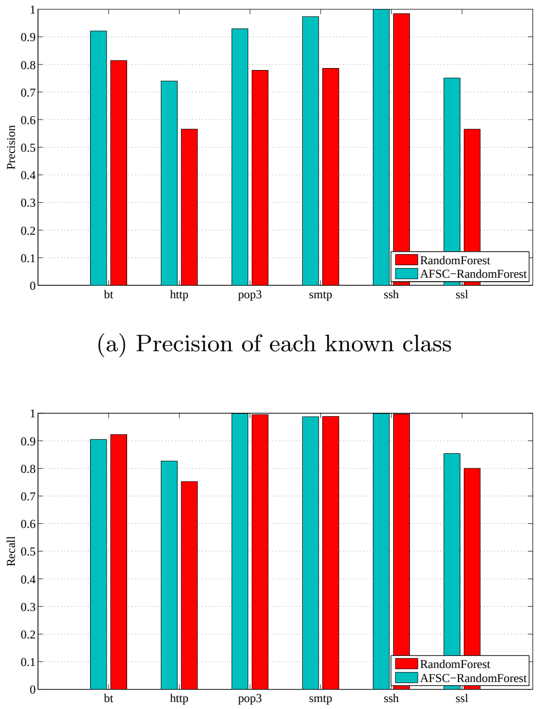
(c) *Random Forest*

Figure 3: F-measure on *isp*

For the empirical study, we use different metrics to evaluate ASFC from multiple perspectives. All used metrics are derived from the four different possible outcomes of a single prediction for a two-class case, true positive (TP), true negative (TN), false positive (FP), and false negative (FN). A false positive is when the real "negative" is incorrectly classified as "positive". A false negative is when the real "positive" is incorrectly classified as "negative". True positives and true negatives are obviously correct classifications. The average performance over 100 runs are reported in this paper.

5.2 Identification Performance of ASFC

We performed a set of experiments to test the identification performance of the proposed ASFC approach, in which three different learning algorithms, *k*-Nearest Neighbour (*k*-NN), Bayes Network (BayesNet) and RandomForest, are used to construct a basic binary classifier. Following the



(a) Precision of each known class

(b) Recall of each known class

Figure 4: Precision and recall of using random forest

existing works in the area, we use *Precision*, *Recall* and *F-measure* to measure the final performance.

- Precision is defined as the ratio of correctly classified flows over all predicted flows in a class. It can be calculated by

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

- Recall is defined as the ratio of correctly classified flows over all ground truth flows in a class. It can be calculated by

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

- F-measure is a combination of precision and recall, it is a widely adopted metric to evaluate per-class performance. It can be calculated by

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{13}$$

Fig. 3 shows the overall performance in terms of the F-measure. To demonstrate the advantage of ASFC, the results of solely using these learning algorithms for flow identification are also reported. One can see that ASFC can effectively improve the F-measure in each class which is independent to the basic learning algorithm. ASFC with random forest can achieve the best performance as shown in Fig. 3(c). For the classes, http, pop3, smtp, and ssl, the improvements can achieve 10%. Although ssh class is easy to identify, ASFC can further improve its F-measure. The F-measure of ASFC with random forest is higher than the method without ASFC about 5%.

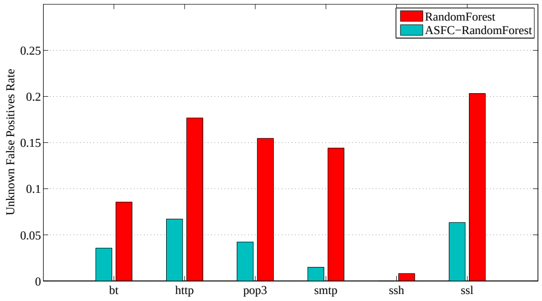


Figure 5: Unknown False Positives Rate

We take ASFC with random forest as an example to study how ASFC can affect the precision and recall for each class. Fig. 4 reports the precision and recall of each class produced by ASFC with random forest, with comparison to the results of random forest. An important observation for this figure is that ASFC can significantly increase the precision without decreasing the recall of each class. It's why the F-measure got much improved. For example, the precision of smtp can be improved about 20% by using ASFC and its recall doesn't drop. In class ssl, ASFC can improve the precision over 15% and the recall by 5% as well.

5.3 Impact of Unknown Information Extraction

This section aims to investigate why unknown information extraction is helpful to deal with unknown applications. In this section, we introduce two new metrics, unknown flow extraction rate and unknown false positive rate. For convenience, the flows generated by unknown applications are called "unknown flows" in the rest of the paper.

- Unknown flow extraction rate is defined as the ratio of the number of extracted unknown flows to the amount of all unknown flows in the unlabelled dataset. A high rate means the proposed unknown extraction method can automatically and effectively collect the information of unknown applications from universal real world network traffic.

- Unknown false positive rate is defined as the ratio of the number of unknown flows that are inaccurately identified as known to the amount of flows that are classified to the known class. We use this rate to show the impact of unknown applications to flow identification. ASFC is developed to significantly reduce the unknown false positive rate.

Table 2 lists the unknown flow extraction rate of ASFC with random forest for each class. For example, the rate is 89.93% for bt that means our iterative method can automatically extract 89.93% from all unknown flows in the unlabelled dataset. In other words, we can successfully obtain the representative samples of unknown applications. Then, these extracted unknown flows are combined into the training set so as to effectively enhance the robustness of the final classifier. From Table 2, we can see that the proposed iter-

Table 2: Unknown flow extraction rate

Class	bt	http	pop3	smtp	ssh	ssl3
Extraction Rate	89.93%	58.79%	76.91%	82.44%	99.74%	63.91%

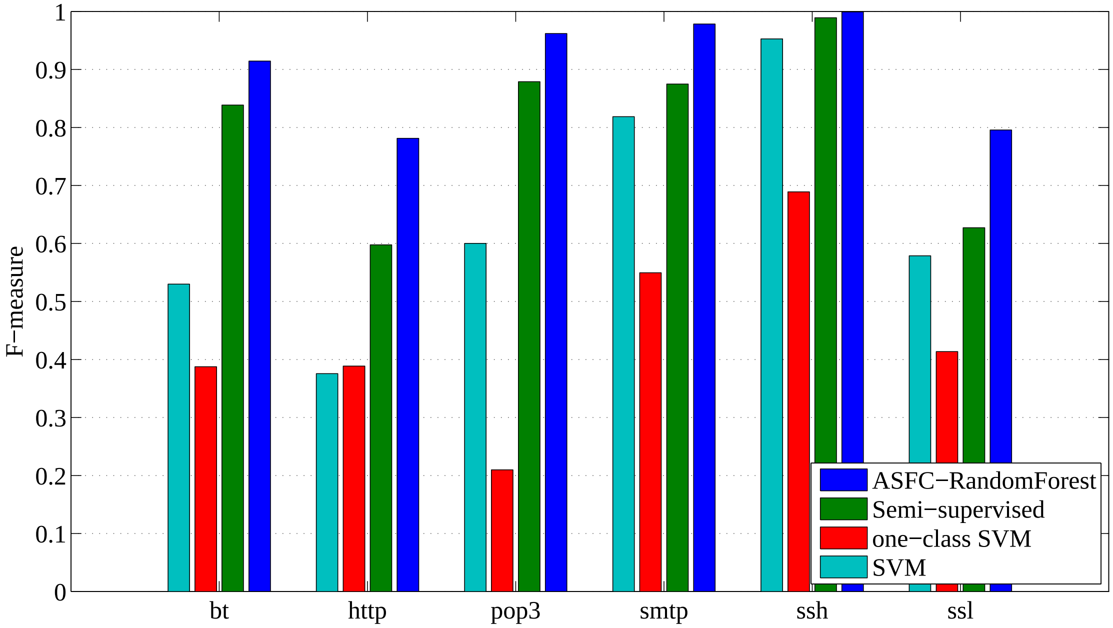


Figure 6: Methods comparison

ative method can extract most of unknown flows from the unlabelled dataset for robust classifier training of each class.

Moreover, we studied how the extracted unknown flows were helpful to improve the identification performance based on the unknown false positive rates as shown in Fig. 5. ASFC with random forest is compared to the method of solely using random forest. The results show that our ASFC with unknown information extraction can effectively reduce the amount of unknown flows that are inaccurately classified to known classes. For example, over 15% of unknown flows in the testing data are inaccurately identified as smtp, but this rate reduces to about 4% when using ASFC. In other words, the proposed approach can effectively detect the flows of unknown applications.

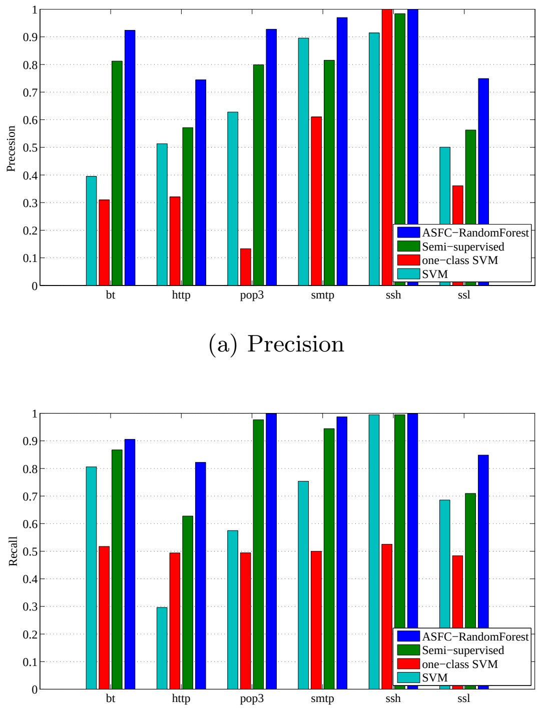
Based on the above results and analysis, we can draw an initial conclusion that ASFC can automatically extract sufficient information of unknown applications from unlabelled network traffic and utilize it to significantly reduce the unknown false positive rate. Therefore, ASFC demonstrates the strong capability of improving the robustness of flow identification.

5.4 Comparison with Other Methods

This section compares our proposed ASFC and two existing works, Erman's semi-supervised method [8] and Este's One-class SVM method [9]. In Erman's semi-supervised methods, we run k-means on the mixture data consisting of the training samples of known classes and the unlabelled data, where k was set to 400 based on the experiments. We found that the performance of flow identification using one-class SVM is very poor. Considering Este's idea, we used SVM to construct a binary classifier, thus the positive training samples and negative training samples can jointly optimize the decision boundary.

Fig. 6 reports the F-measures of these methods. Based on the results, the ranking list is ASFC-RandomForest, semi-supervised, SVM, and one-class SVM. In particular, our ASFC-RandomForest is much better than other competing methods. For example, ASFC-RandomForest outperforms semi-supervised over 15% in http class. The improvement is about 10% in other classes except ssh. For ssh, both methods achieve nearly 100% F-measure.

Fig. 7 shows the precision and recall of all competing methods. We can see that ASFC always significantly outperforms other methods in either precision or recall for any classes. For example, the precision of ASFC is higher than the second best, semi-supervised, over 15% in http class. The improvement on recall can achieve 20% in this class. In class ssl, the improvements on both precision and recall are over 15%. In general, ASFC has the superior capability to improve precision and recall when unknown applications present. The basic reason is ASFC can effectively extract unknown information and combine correlated flows to make more accurate decisions.



(a) Precision

(b) Recall

Figure 7: Precision and recall of competing methods

6. CONCLUSION AND DISCUSSION

This paper solves a critical problem of traffic classification with unknown applications presented, which normally exists in real world networks. To address this problem, we proposed an iterative method to extract unknown information from a set of unlabelled traffic flows, which combines asymmetric bagging and flow correlation to guarantee the purity of extracted negatives. For traffic classification, a binary classifier was created as an application signature which can operate on a bag of correlated flows instead of individual flows to further improve classification accuracy. We carry out a series of experiments in a real-world network traffic dataset to evaluate the proposed methods. The results show that the proposed method outperforms the-state-of-art traffic classification methods under the extreme situation of unknown applications present. It is due to the high unknown extraction rate which leads to less unknown flows inaccurately classified to known classes.

Our empirical study suggests that the iterative extraction method only works when the asymmetric bagging and flow correlation techniques are used. Asymmetric bagging and flow correlation both have significant impact to the final performance. For example, with flow correlation, the F-measure can improve 5% for a class. By combining asymmetric bagging and flow correlation, the improvement can achieve 10%. In the future, we will develop practical software to facilitate unknown information extraction and apply it to real-time traffic classification.

7. REFERENCES

[1] T. Auld, A. W. Moore, and S. F. Gull. Bayesian neural networks for internet traffic classification. *IEEE Trans. Neural Netw.*, 18(1):223–239, January 2007.

[2] L. Bernaille and R. Teixeira. Early recognition of encrypted applications. In *Proceedings of the 8th international conference on Passive and active network measurement*, pages 165–175, Berlin, Heidelberg, 2007.

[3] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian. Traffic classification on the fly. *SIGCOMM Comput. Commun. Rev.*, 36:23–26, April 2006.

[4] D. Bonfiglio, M. Mellia, M. Meo, and D. Rossi. Detailed analysis of skype traffic. *IEEE Trans. Multimedia*, 11(1):117 –127, Jan. 2009.

[5] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli. Revealing skype traffic: when randomness plays with you. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications*, pages 37–48, New York, NY, USA, 2007.

[6] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli. Traffic classification through simple statistical fingerprinting. *SIGCOMM Comput. Commun. Rev.*, 37:5–16, January 2007.

[7] J. Erman, M. Arlitt, and A. Mahanti. Traffic classification using clustering algorithms. In *Proceedings of the SIGCOMM workshop on Mining network data*, pages 281–286, New York, NY, USA, 2006.

[8] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson. Offline/realtime traffic classification using semi-supervised learning. *Performance Evaluation*, 64(9-12):1194–1213, October 2007.

[9] A. Este, F. Gringoli, and L. Salgarelli. Support vector machines for tcp traffic classification. *Computer Networks*, 53(14):2476–2490, September 2009.

[10] Z. M. Fadlullah, T. Taleb, A. V. Vasilakos, M. Guizani, and N. Kato. DTRAB: combating against attacks on encrypted protocols through traffic-feature analysis. *IEEE/ACM Trans. Netw* 18(4):1234–1247, August 2010.

[11] A. Finamore, M. Mellia, M. Meo, and D. Rossi. KISS: Stochastic packet inspection classifier for UDP traffic. *IEEE/ACM Trans. Netw* 18(5):1505–1515, October 2010.

[12] E. Glatz and X. Dimitropoulos. Classifying internet one-way traffic. *SIGMETRICS Perform. Eval. Rev.*, 40(1):417–418, Jun. 2012.

[13] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003.

[14] P. Haffner, S. Sen, O. Spatscheck, and D. Wang. ACAS: automated construction of application signatures. In *Proceedings of the ACM SIGCOMM workshop on Mining network data*, pages 197–202, New York, NY, USA, 2005. ACM.

[15] Y. Jin, N. Duffield, J. Erman, P. Haffner, S. Sen, and Z.-L. Zhang. A modular machine learning system for flow-level traffic classification in large networks. *ACM Trans. Knowl. Discov. Data*, 6(1):4:1–4:34, Mar. 2012.

[16] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee. Internet traffic classification demystified: myths, caveats, and the best practices. In

Proceedings of the ACM CoNEXT Conference, pages 1–12, New York, NY, USA, 2008.

[17] T. Nguyen, G. Armitage, P. Branch, and S. Zander. Timely and continuous machine-learning-based classification for interactive ip traffic. *IEEE/ACM Trans. Netw* 20(6):1880-1894, December 2012.

[18] T. T. Nguyen and G. Armitage. A survey of techniques for internet traffic classification using machine learning. *IEEE Commun. Surv. Tutor.* 10(4):56–76, Fourth Quarter 2008.

[19] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* 28(7):1088–1099, July 2006.

[20] Y. Wang, Y. Xiang, J. Zhang, and S.-Z. Yu. A novel semi-supervised approach for network traffic clustering. In *International Conference on Network and System Security*, Milan, Italy, September 2011.

[21] Y. Xiang, W. Zhou, and M. Guo. Flexible deterministic packet marking: An ip traceback system to find the real source of attacks. *IEEE Trans. Parallel Distrib. Syst.* 20(4):567–580, April 2009.

[22] H. Yu, J. Han, and K. C.-C. Chang. PEBL: web page classification without negative examples. *IEEE Trans. Knowl. Data Eng.* 16(1):70–81, Jan. 2004.

[23] S. Zander, T. Nguyen, and G. Armitage. Automated traffic classification and application identification using machine learning. In *Annual IEEE Conference on Local Computer Networks*, pages 250–257, Los Alamitos, CA, USA, 2005.

[24] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan. Network traffic classification using correlation information. *IEEE Trans. Parallel Distrib. Syst.* 24(1):104–117, January 2013.

[25] J. Zhang and L. Ye. Content based image retrieval using unclean positive examples. IEEE Trans. Image Process. 18(10):2370–2375, Oct. 2009.