Robust Network Traffic Classification

Jun Zhang, *Member, IEEE*, Xiao Chen, *Student Member, IEEE*, Yang Xiang, *Senior Member, IEEE*, Wanlei Zhou, *Senior Member, IEEE*, and Jie Wu, *Fellow, IEEE*

***Abstract*—As a fundamental tool for network management and security, traffic classification has attracted increasing attention in recent years. A significant challenge to the robustness of classification performance comes from zero-day applications previously unknown in traffic classification systems. In this paper, we propose a new scheme of Robust statistical Traffic Classification (RTC) by combining supervised and unsupervised machine learning techniques to meet this challenge. The proposed RTC scheme has the capability of identifying the traffic of zero-day applications as well as accurately discriminating predefined application classes. In addition, we develop a new method for automating the RTC scheme parameters optimization process. The empirical study on real-world traffic data confirms the effectiveness of the proposed scheme. When zero-day applications are present, the classification performance of the new scheme is significantly better than four state-of-the-art methods: random forest, correlation-based classification, semi-supervised clustering, and one-class SVM.**

***Index Terms*—Semi-supervised learning, traffic classification, zero-day applications.**

I. Introduction

TRAFFIC classification is fundamental to network management and security [1], which can identify different applications and protocols that exist in a network. For example, most QoS control mechanisms have a traffic classification module in order to properly prioritize different applications across the limited bandwidth. To implement appropriate security policies, it is essential for any network manager to obtain a proper understanding of applications and protocols in the network traffic. Over the last decade, traffic classification has been given a lot of attention from both industry and academia.

There are three categories of traffic classification methods: port-based, payload-based, and flow statistics-based [2]. The traditional port-based method relies on checking standard ports used by well-known applications. However, it is not always reliable because not all current applications use standard ports. Some applications even obfuscate themselves by using the well-defined ports of other applications. The payload-based method searches for the application's signature in the payload of IP packets that can help avoid the problem of dynamic ports. Hence, it is most prevalent in current industry products. However, more often than not, the payload-based method fails with encrypted traffic. In recent academic research, significant attention has been given to applying machine learning techniques to the flow statistics-based method. The statistical method only uses flow statistical features, such as interpacket time, without requiring deep packet inspection (DPI).

In the traditional framework of multiclass classification, most flow statistics-based methods employ supervised or unsupervised machine learning algorithms to classify network traffic into predefined classes based on known applications. The supervised methods can learn a traffic classifier from a set of labeled training samples. By contrast, the methods using unsupervised algorithms automatically categorize a set of unlabeled training samples and apply the clustering results to construct a traffic classifier with the assistance of other tools, such as DPI. Under the assumption that any traffic comes from a known class, a number of promising results have been reported in the literature.

However, existing flow statistics-based methods suffer from zero-day applications previously unknown in traffic classification systems. Generally speaking, the traffic of zero-day applications (zero-day traffic) is the major portion of unrecognized data making up to 60% of flows and 30% of bytes in a network traffic dataset [3]. More specifically, the problem of zero-day applications is conventional methods misclassify zero-day traffic into the known classes, which results in poor accuracies of known classes.

In this paper, a novel traffic classification scheme is proposed to tackle the problem of zero-day applications. Our scheme can effectively improve the accuracies of known classes when zero-day applications are present. The major contributions of our work are summarized as follows.

- We propose a Robust Traffic Classification (RTC) scheme, combining supervised and unsupervised learning to address the problem of zero-day applications.
- We present a new method to effectively extract the samples of zero-day traffic from unlabeled network traffic.
- We develop a new method for automating the RTC scheme parameters optimization process.

To evaluate the new scheme, a large number of experiments were carried out on multiple real-world network traffic datasets. The results show the proposed scheme significantly outperforms the state-of-the-art traffic classification methods when zero-day applications are present. Following our previous work [4], flow correlation was used in the new scheme to improve classification performance. In this paper, we provide a new quantitative study based on probability theory to show how flow correlation can benefit traffic classification.

The rest of this paper is organized as follows. Section II presents a critical review on flow statistics-based traffic

Manuscript received April 11, 2013; revised October 05, 2013; February 06, 2014; and April 16, 2014; accepted April 21, 2014; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Meo.

J. Zhang, X. Chen, Y. Xiang, and W. Zhou are with the School of Information Technology, Deakin University, Melbourne, Vic. 3125, Australia (e-mail: jun.zhang@deakin.edu.au; chxiao@deakin.edu.au; yang.xiang@deakin.edu.au; wanlei@deakin.edu.au).

J. Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: jiewu@temple.edu).

Digital Object Identifier 10.1109/TNET.2014.2320577

classification. In Section III, a novel traffic classification scheme is proposed to deal with zero-day applications. Section IV presents a new method of parameter optimization for the proposed scheme. For performance evaluation, a large number of experiments and results are reported in Section V. Section VI provides further discussion on the proposed scheme. Finally, Section VII concludes this paper.

II. Related Work

Current research on network traffic classification focuses on the application of machine learning techniques to flow statistics-based methods [2]. This can avoid problems suffered by port-based and payload-based methods such as dynamic ports, encrypted applications, and user privacy. However, the flow statistic-based method will not be practical until it meets several challenges. Previously, the biggest challenge was real-time traffic classification at increasing wire speeds. Now, operators face another challenge—zero-day applications—due to the tremendous development rate of new applications [5]. We provide a review of state-of-the-art flow statistics-based methods with consideration given to zero-day applications.

Let us start with a typical real-world network scenario. Suppose the traffic dataset, Ω, consists of N known classes and U unknown classes, $\Omega = \{\omega_1, \ldots, \omega_N, \overline{\omega}_1, \ldots, \overline{\omega}_U\}$. In this paper, a set of labeled flow samples, ψ_n, is available for a known class, ω_n. By contrast, no labeled flow samples are available for an unknown class associated with a previously unknown application in the system. Given a flow in the dataset, the traffic classification problem is to identify if it belongs to a specific known class. A flow consists of successive IP packets with the same 5-tuple: source IP, source port, destination IP, destination port, transport protocol.

A. N-Class Classification

Conventional flow statistics-based methods address an N-class classification problem without consideration of zero-day traffic. A typical supervised classification method uses the labeled flow samples, $T = \bigcup_{i=1}^{N} \psi_i$, straightforward, and employs a machine learning algorithm to construct a classifier. The classifier trained by using T will classify any testing flow into one of the predefined classes. Thus, zero-day traffic flows in unknown classes, $\{\overline{\omega}_1, \ldots, \overline{\omega}_U\}$, will be misclassified into N known classes. The classification performance will be severely affected by zero-day traffic. In early work, Moore and Zuev [6] applied the naive Bayes techniques to classify network traffic based on the flow statistical features. Later, several well-known algorithms were also applied to traffic classification, such as Bayesian neural networks [7], and support vector machines [8]. Erman *et al.* [9] proposed using unidirectional statistical features to facilitate traffic classification in the network core. For real-time traffic classification, several supervised classification methods [10], [11] using only the first few packets were proposed. Considering the first few packets of flows could be missed or disguised, some researchers proposed classifying a subflow captured at any given time [12], [13]. Bermolen *et al.* [14] studied certain popular P2P-TV applications, and found P2P-TV traffic can simply be identified by the count of packets and bytes exchanged among peers during small time windows. Our previous work [4] incorporated flow correlation into supervised classification, which displayed superior classification performance, even when the training set was insufficient. Glatz *et al.* [15] proposed a new scheme to classify one-way traffic into classes such as "Malicious Scanning," "Service Unreachable," etc., based on prefixed rules. Thus, no training stage was needed. Jin *et al.* [16] developed a lightweight traffic classification architecture combining a series of simple linear binary classifiers and embracing three key innovative mechanisms to achieve scalability and high accuracy. A similar idea of a classifier combination was also applied in Callado *et al.*'s work [17]. Carela-Espanol *et al.* [18] analyzed the impact of sampling when classifying NetFlow data and proposed an improvement to the training process in order to reduce the impact of sampling. Other existing work includes the Pearson's chi-Square test-based technique [19], probability density function (PDF)-based protocol fingerprints [20], and small time-windows-based packet count [21].

Previous work has also applied unsupervised clustering algorithms to categorize unlabeled training samples and used the clusters produced to construct a traffic classifier. McGregor *et al.* [22] proposed grouping traffic flows into a small number of clusters using the expectation maximization (EM) algorithm and manually labeling each cluster to an application. Some other well-known clustering algorithms, such as AutoClass [23], k-means [24], DBSCAN [25], and Fuzzy C-means [26], were also applied to traffic classification. Bernaille *et al.* [27] applied the k-means algorithm to traffic clustering and labeled the clusters to applications by using a payload analysis tool. Wang *et al.* [28] proposed integrating statistical feature-based flow clustering with a payload signature matching method to eliminate the requirement of supervised training data. Finamore *et al.* [29] combined flow statistical feature-based clustering and payload statistical feature-based clustering for mining unidentified traffic.

In addition, Ma *et al.* [30] analyzed three mechanisms using statistical and structural content models for traffic identification. Their classification methods rely on the content of IP payload and employ unsupervised clustering techniques.

Some empirical studies evaluated the traffic classification performance of different methods. The early works were reported by Roughan *et al.* [31] and Williams *et al.* [32]. Kim *et al.* [3] extensively evaluated the ports-based CorelReef method, the host behavior-based BLINC method, and seven common statistical feature-based methods using supervised algorithms on seven different traffic traces. Lim *et al.* [33] identified the role of feature discretization for different supervised classification algorithms during the empirical study. However, these empirical studies did not investigated traffic classification with zero-day applications. In addition, Lee *et al.* [34] recently developed a benchmark tool integrating 11 state-of-the-art traffic classifiers.

B. $(N+1)$-Class Classification

A semi-supervised method [35] was proposed to take unknown applications into account. First, a mixture of labeled and unlabeled training samples are grouped into k clusters using traditional clustering algorithms such as k-means. Then, traffic

clusters are mapped to $\omega_1, \ldots, \omega_N$, or unknown, according to the locations of the labeled (supervised) training samples. For traffic classification, a flow is predicted to the class of its nearest cluster. This method demonstrates the potential of dealing with zero-day traffic generated by unknown applications. Our work is based on Erman's semi-supervised method [35] and makes contributions to zero-day traffic identification and automatic parameter optimization. Later, the ensemble clustering technique is introduced to improve the semi-supervised method [36]. Liu *et al.* [37] extended Erman's work to classify encrypted traffic by using the composite feature set and combining the first 40-B payload with statistical features of the flow level.

Some methods addressed a one-class classification problem that has potential to deal with zero-day traffic. Considering one-class classification, 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 identified as unknown traffic. This means the problem of zero-day applications can be bypassed. An early work is creating a one-class classifier using a normalized threshold on statistical features [20]. This method is heuristic and unreliable because the normalized threshold is hard to tune beforehand, especially without information concerning zero-day traffic. A modified one-class SVM method has been proposed for traffic classification [8]. For a known class ω_n, the training samples in ψ_n are used to learn a one-class SVM, and other training samples in $\bigcup_{i=1,i\neq n}^{i=N} \psi_i$ are used to adjust the decision boundary. This method has two issues. First, one-class SVM [38] normally requires a large number of training samples. Second, the decision boundary is poor due to a lack of information about the unknown classes, $\{\overline{\omega}_1, \ldots, \overline{\omega}_U\}$. Xie *et al.* [39] proposed a subflow scheme that learns to identify each application in isolation, instead of distinguishing them individually using subspace clustering. However, the binary classifier for each application is heuristic and relies on a predefined distance threshold. Moreover, the implementation of their scheme is unclear.

III. Proposed Scheme: RTC

As discussed in Section II, existing traffic classification methods suffer the problem of zero-day applications due to a lack of zero-day traffic samples in the classifier training stage. How to obtain sufficient zero-day traffic samples becomes a key question for fundamentally solving this problem. Our work is motivated by the observation that unlabeled network data contains zero-day traffic. We aim to build a robust classifier by extracting zero-day samples and incorporating them into the training stage.

This section presents a robust traffic classification scheme to deal with zero-day applications. Fig. 1 shows a new framework of RTC. There are three important modules in the proposed framework: unknown discovery, "bag of flows" (BoF)-based traffic classification, and system update. The module of unknown discovery aims to automatically find new samples of zero-day traffic in a set of unlabeled traffic randomly collected from the target network. The module of BoF-based traffic classification takes prelabeled training samples and zero-day traffic samples as input to build a classifier for robust traffic classification. To achieve fine-grained classification, the module of a system update can intelligently analyze the zero-day traffic and construct new classes to complement the system's knowledge. In this paper, we provide an implementation of RTC in which the algorithms of random forest and k-means are employed to perform supervised classification and unsupervised learning (clustering).

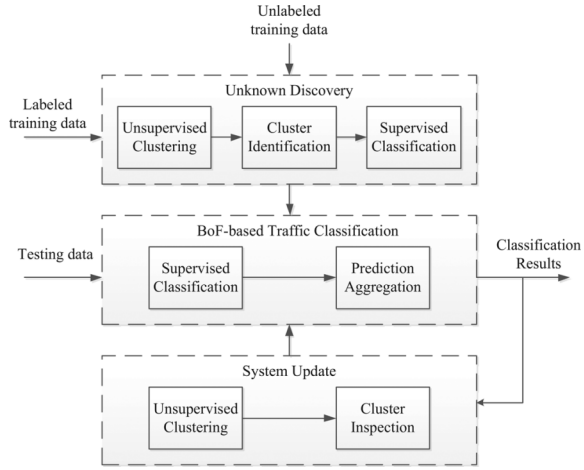


Fig. 1. RTC framework.

Algorithm 1: Zero-day samples extraction

Require: labeled sets $\{\psi_1, \ldots, \psi_N\}$, unlabeled set T_u
Ensure: zero-day sample set U
1: $T_l \leftarrow \bigcup_{i=1}^{N} \psi_i$
2: $T \leftarrow T_l \cup T_u$
3: Perform clustering on T to obtain clusters $\{C_1, \ldots, C_k\}$
4: $V \leftarrow \emptyset$
5: **for** $i = 1$ **to** k **do**
6: **if** C_i does not contain any labeled flows from T_l **then**
7: $V \leftarrow (V \cup C_i)$
8: **end if**
9: **end for**
10: Combine $\{\psi_1, \ldots, \psi_N\}$ and V to train a $(N+1)$-class classifier f_{c1} {V is for a generic unknown class}
11: $U \leftarrow \emptyset$
12: **while** Classify all flows in T_u by f_{c1} **do**
13: **if** $\mathbf{x}$ is predicted to the unknown class **then**
14: Put $\mathbf{x}$ into U
15: **end if**
16: **end while**

A. Unknown Discovery

We propose a two-step method of unknown discovery to extract zero-day traffic samples from a set of unlabeled network traffic crucial to the RTC scheme. The two-step method is summarized in Algorithm 1. The first step is the k-means based identification of zero-day traffic clusters. The second step is zero-day sample extraction using random forest.

Given the prelabeled training sets $\{\psi_1, \ldots, \psi_N\}$ and an unlabeled set T_u, we roughly filter out some zero-day samples out from T_u by using a semi-supervised idea for the first step. The labeled and unlabeled samples are merged to feed

the clustering algorithm, k-means. The k-means clustering aims to partition the traffic flows into k clusters ($k \leq |T|$), $C = \{C_1, \ldots, C_k\}$, to minimize the within-cluster sum of squares. The traditional k-means algorithm uses an iterative refinement technique. Given an initial set of randomly selected k centroids, the algorithm proceeds by alternating between the assignment step and the update step [40]. In the assignment step, each flow is assigned to the cluster with the closest mean

$$C_i^t = \{\mathbf{x}_j : \|\mathbf{x}_j - \mathbf{m}_i^t\| \leq \|\mathbf{x}_j - \mathbf{m}_l^t\| \text{ for all } l = 1, \ldots, k\}. \quad (1)$$

In the update step, the new means are calculated to be the centroid of flows in the cluster. By choosing a large k [25], [41], we obtain the high-purity traffic clusters, $\{C_1, \ldots, C_k\}$. The prelabeled training samples can then be used to identify zero-day traffic clusters. The rule is as follows.

- If a cluster does not contain any prelabeled samples, it is a zero-day traffic cluster.

However, simply put, a large k will lead to a high TP rate as well as a high FP rate of unknown detection that will seriously affect the purity of the detected unknown samples.

In the second step, we propose creating a random forest classifier in order to address this issue. A generic unknown class is proposed to represent the mixture of zero-day applications. The zero-day sample set V obtained in the first step is temporally used as the training set for this generic unknown class. Thus, we have a specific multiclass classification problem involving N known classes and one unknown class. Then, prelabeled training sets $\{\psi_1, \ldots, \psi_N\}$ and temporal zero-day sample set V combine to train a random forest classifier, f_{c1}. Random forest with good generalization capability displayed excellent classification performance in previous work on traffic classification. We further apply f_{c1} to classify flows in T_u to obtain a high-purity set of zero-day samples, U. In particular, to guarantee the purity of zero-day samples, we apply a new classification method that considers flow correlation [4] in real-world traffic. This will be described in detail in Section III-B.

B. BoF-Based Traffic Classification

For robust traffic classification, we further propose a new classification method that considers flow correlation in real-world network traffic and classifies correlated flows together rather than in single flows.

Algorithm 2 presents the proposed method of BoF-based traffic classification. Given the prelabeled training sets $\{\psi_1, \ldots, \psi_N\}$ and the zero-day sample set U produced by the module of unknown discovery, we can build classifier f_{c2} for the $(N+1)$-class classification. f_{c2} is able to categorize zero-day traffic into a generic unknown class. Following our previous work [4], we incorporate flow correlation into the traffic classification process in order to significantly improve identification accuracy. Flow correlation can be discovered by the 3-tuple heuristic [30], [42], [43]. That is, in a short period of time, the flows sharing the same destination IP, destination port, and transport protocol are generated by the same application/protocol. For convenience of traffic classification, we use "bag of flows" to model flow correlation. A BoF can be described by $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_g\}$, where $\mathbf{x}_i$ represents the

Algorithm 2: BoF-based Traffic Classification

Require: labeled sets $\{\psi_1, \ldots, \psi_N\}$, zero-day sample set U, testing set Ω_t
Ensure: label set L_t for testing flows
1: Combine $\{\psi_1, \ldots, \psi_N\}$ and U to train a $(N+1)$-class classifier f_{c2} {U represent a generic unknown class}
2: Construct BoFs $\mathbf{X} = \{X_i\}$ from Ω_t according to 3-tuple heuristic {consider flow correlation in traffic classification}
3: **while** $\mathbf{X} \neq \emptyset$ **do**
4: Take a BoF X_i from $\mathbf{X}$
5: **for** $j = 1$ **to** $|X_i|$ **do**
6: Classify $\mathbf{x}_{ij}$ by f_{c2}
7: **end for**
8: Make final decision by aggregating the predictions of flows in BoF X_i
9: Assign the label of X_i to all flows in this BoF
10: **end while**

ith flow in the BoF. Classification of a BoF can be addressed by aggregating the flow predictions produced by a conventional classifier. In this paper, the aggregated classifier $f_{bof}(X)$ can be expressed as

$$f_{\text{bof}}(X) = \Theta_{\mathbf{x} \in X}(f_{c2}(\mathbf{x})) \quad (2)$$

where f_{c2} denotes the random forest classifier and Θ is the majority vote method [44]. For BoF X, we have g flow predictions $y_{x1}, \ldots, y_{xg}$ produced by f_{c2} (g is the number of flows in X). The flow predictions can be simply transformed into votes

$$v_{ij} = \begin{cases} 1, & \text{if } y_{xj} \text{ indicates the } i\text{th class,} \\ 0, & \text{otherwise.} \end{cases} \quad (1 < j \leqslant g) \quad (3)$$

Then, the compound decision rule is

$$\text{assign } X \rightarrow \omega_l \text{ if} \sum_{j=1}^{g} v_{lj} = \max_{i=1,\ldots,q} \sum_{j=1}^{g} v_{ij}. \quad (4)$$

Consequently, all flows in X are classified into ω_l. The BoF-based traffic classification is also used for unknown discovery in Section III-A.

Here, we provide formal justification on the benefit of flow correlation for traffic classification. In the previous work [4], we found that the accuracy of flow-statistics-based traffic classification can be improved significantly by combining multiple correlated flows. For the theoretical study, and given that BoF $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$, we made a compound prediction using the average of predictions made on each flow. Based on the Bayesian decision theory, the average combination rule can be transformed to the majority vote rule under Kittler's theoretical framework [45].

If we consider a classification problem, where we try to predict *a posteriori* probability, and we suppose a trained predictive model is f, the compound prediction is given by

$$f_{\text{bof}}(X) = \frac{1}{M} \sum_{m=1}^{M} f(\mathbf{x}_m). \quad (5)$$

Suppose the true *posteriori* probability function we are trying to predict is given by $p(\mathbf{x}_m)$, the output of each random flow can be written as the true value plus an error in the form

$$f(\mathbf{x}_m) = p(\mathbf{x}_m) + e(\mathbf{x}_m). \tag{6}$$

The average sum-of-squares error can be described as

$$E[\{f(\mathbf{x}_m) - p(\mathbf{x}_m)\}^2] = E[e(\mathbf{x}_m)^2] \tag{7}$$

where $\mathbf{x}_m$ has its own distribution and $E[\cdot]$ denotes the expectation with respect to its distribution. The average error made by the flows individually is therefore

$$E_{\text{flow}} = \frac{1}{M}\sum_{m=1}^{M} E[e(\mathbf{x}_m)^2]. \tag{8}$$

Similarly, the expected error for the BoFs is given by

$$\begin{aligned} E_{\text{bof}} &= E[\{\frac{1}{M}\sum_{m=1}^{M} f(\mathbf{x}_m) - \frac{1}{M}\sum_{m=1}^{M} p(\mathbf{x}_m)\}^2] \\ &= E[\{\frac{1}{M}\sum_{m=1}^{M} e(\mathbf{x}_m)\}^2]. \end{aligned} \tag{9}$$

We assume errors have a zero mean and are uncorrelated, i.e., $E[e(\mathbf{x}_m)] = 0$, and $E[e(\mathbf{x}_m)e(\mathbf{x}_l)] = 0, m \neq l$. Then, we obtain

$$= \frac{1}{M} E_{\text{flow}}. \tag{10}$$

This result suggests the flow prediction error can be reduced by a factor of M by using a simple BoF-based model.

A further study on E_{bof} will be presented in Section VI by relaxing the independent assumption.

C. System Update

With unknown discovery and BoF-based traffic classification, the proposed scheme has identified zero-day traffic when performing traffic classification. The module of system update is proposed to achieve fine-grained classification of zero-day traffic. The purpose is to learn new classes in identified zero-day traffic and to complement the system's knowledge. The capability of learning new classes makes the proposed scheme different to the conventional traffic classification method.

The procedure of learning new classes is shown in Algorithm 3. Given a set of zero-day traffic, Z, which is the outcome of BoF-based traffic classification, we perform k-means clustering to obtain the clusters $\{C_1, \ldots, C_k\}$. For each cluster, we randomly select several sample flows (e.g., three) for manual inspection. To guarantee high purity of new training sets, the consensus strategy is adopted to make a prediction. If all the selected flows indicate a new application/protocol, we create a new class and use the flows in the cluster as its training data. For a new class that has been created during the system update, the flows in the cluster will be added to the training set of that class. Once the cluster inspection is completed, the new detected classes will be added into the set of known classes, and the training dataset will be extended accordingly. This means the classification system is able to learn new classes. The updated system can deal with more applications and achieve further fine-grained classification.

Algorithm 3: New class detection

Require: zero-day traffic Z
Ensure: training samples for new classes
1: Perform clustering on Z to obtain k clusters $\{C_1, \ldots, C_k\}$
2: **for** $i = 1$ **to** k **do**
3: Randomly select A flows from C_i
4: Manually inspect these A flows {Involve a little human effort}
5: **if** All of the selected flows are generated by the same application **then**
6: **if** This is a new application **then**
7: **if** It has been identified **then**
8: Merge C_i and its training set
9: **else**
10: Create a training set ψ' for this new application
11: **end if**
12: **end if**
13: **end if**
14: **end for**

Frequent system update is not necessary according to previous research [35]. If the classified zero-day traffic indicates any significant change to the applications, the system update will be triggered to retrain the RTC classifier. Some discussions on classifier retraining are provided in Section VI-B.

In the above-mentioned procedure, training samples for new classes may include noise because traffic clusters are not 100% pure. This issue may affect the classification accuracy of known classes. To tackle this issue, we propose the application of a two-level classification strategy.

In the first level, the $(N+1)$-classes classifier obtained before the system update can be utilized to perform traffic classification. Ideally, zero-day traffic will be classified into a generic unknown class. In the second level, training samples for new classes obtained during a system update can be used to train a new classifier, and this classifies traffic in the generic unknown class into fine-grained new classes. The advantage of the two-level classification strategy is the performance of known classes will not be affected. In this sense, the robustness of the traffic classification system can be improved.

IV. Parameter Optimization

The setting of a parameter is a significant challenge for a traffic classification method that applies machine learning techniques. We observe the performance of the proposed RTC scheme relies on the effectiveness of unknown discovery. In unknown discovery, there are two parameters: k determining the number of clusters produced by k-means, and T_u indicating the size of an unlabeled training set. Fig. 2 reports the true positive rate (TPR) and the false positive rate (FPR) of zero-day

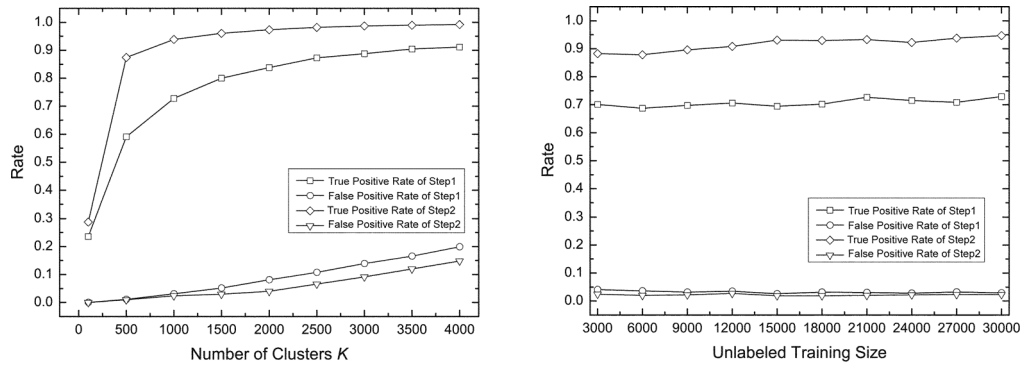


Fig. 2. Impact of parameters to unknown discovery. (a) TPR and FPR for various number of clusters K. (b) TPR and FPR for various unlabeled training sizes.

sample detection produced by unknown discovery. The experiment setup we used here is consistent with the one we used in Section V-B. TPR is the rate of the sum of correctly detected zero-day traffic to the sum of all actual zero-day traffic. FPR is the rate of the sum of the traffic inaccurately detected as zero-day to the sum of traffic of known applications. Fig. 2(a) shows the results with a fixed $T_u = 30\,000$ and various k. It is clear that while the FPR produced in the first step was low, the corresponding TPR was not high either. The second step significantly improved TPR and further reduced FPR. TPR of unknown discovery changed from about 28% to 99% when k increased from 100 to 4000. Meanwhile, its FPR increased from 0% to 20%. The final classification performance will have a big difference if k changes dramatically. It is necessary to select a good k to balance TPR and FPR in order to achieve high classification accuracy. By fixing k to 1000 and varying T_u from 3000 to 30 000, we obtain Fig. 2(b). This figure shows that increasing T_u can slightly affect TPR and FPR. Compared to the first step, the second step can effectively improve TPR by about 20%. If we consider T_u is out of control in practical applications, our parameter setting focuses on k.

We propose a new optimization method combining a 10-fold cross validation and binary search to find an optimal k. The advantage of the optimization method is twofold: accuracy and speed. This method is applied in the proposed RTC scheme for performance evaluation, as mentioned in Section V-B. In 10-fold cross validation, the original training set, including labeled and unlabeled traffic flows, is randomly partitioned into 10 equal-size subsets. Of the 10 subsets, a single subset is retained as validation data for testing the model of unknown discovery. The remaining nine subsets are used as training data. The cross-validation process is then repeated 10 times, with each of the 10 subsets used exactly once as the validation data. The 10 results from the folds are then averaged to produce a single estimation.

A new problem of which metric can be used to evaluate the results of unknown discovery in cross validation is critical to the optimization accuracy. Accuracy is a single value common for measuring the overall performance of traffic classification. However, accuracy calculated using the labeled training data for known classes cannot measure the performance of zero-day traffic detection. Based on the empirical results as shown in Fig. 2(a), we find FPR is a good measure for cross validation. From a theoretical point of view, our original idea is the following:

- to search for a maximum k that does not produce false positives.

This refers to our ability to detect as many accurate zero-day samples as possible without introducing any errors. However, experimental results show the TPR obtained using this idea is low. An observation from Fig. 2(a) is that TPR dramatically increases if FPR slightly increases from 0. Practically, the threshold of the false positive for parameter optimization can be set to a small value. Based on our experiments, we find that 3% is a good value for FPR.

Another problem is that searching for an optimal k is time-consuming. For example, if the training set has 10 000 flows, k may change from 1 to 10 000. Fortunately, we find FPR is monotone and increases as k increases. Therefore, a binary search is helpful to quickly find k, and the corresponding FPR is closest to 3%. Algorithm 4 describes the procedure of automatic parameter selection. A binary search of k takes logarithmic time, which is very efficient. Fig. 3 shows the results of this intelligent method for different T_u. It is clear that a bad k can severely affect classification accuracy. This optimization method can successfully find an optimized k and produce excellent traffic classification accuracy. The traffic dataset used in this section also refers to Section V.

V. Performance Evaluation

A large number of experiments were carried out on real-world traffic datasets to compare the RTC scheme with four state-of-the-art traffic classification methods. This section reports the experiments and results.

A. Dataset

In this paper, four Internet traffic traces are used for our experimental study, as shown in Table I. They are captured from three Internet positions located around the world, so the sampling points are heterogeneous in terms of link type and capacity. The collection time ranges from 2006 to 2010, covering five recent years in which the Internet has grown and evolved rapidly. Since either partial or full packet payload is preserved in these traffic traces, we build the ground truth (i.e., the actual classes of

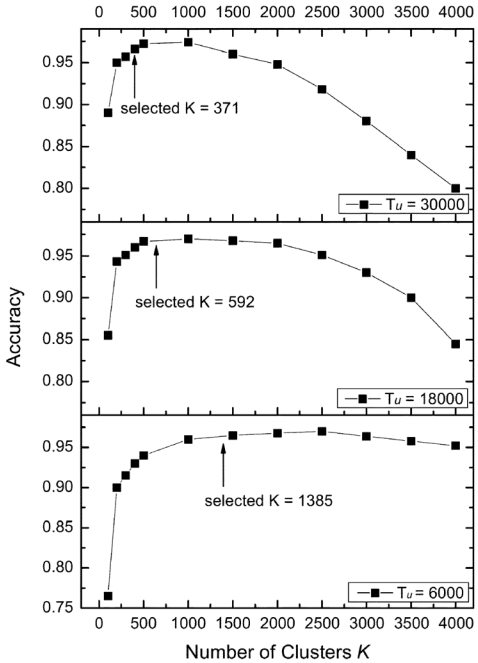


Fig. 3. Automatic selection of k.

Algorithm 4: Parameter optimization

Require: the module of unknown discovery with cross validation, $D_{\text{fpr}}(k)$; the size of mixed training data, N_t; FPR threshold T_f
Ensure: an optimal k

```
T_f ← 3% {default setting to stop searching}
imin ← 1 and imax ← N_t {searching range}
k = 0
while imax ≠ imin do
    k ← (imin + imax)/2
    if D_fpr(k) < T_f then
        imin ← k + 1
    else if D_fpr(k) > T_f then
        imax ← k − 1
    else
        break
    end if
end while
```

traffic flows) with high confidence. The KEIO and WIDE traces are provided by the public traffic data repository, maintained by the MAWI working group (http://mawi.wide.ad.jp/mawi/). The KEIO trace is captured at a 1-Gb/s Ethernet link in Keio University's Shonan-Fujisawa campus in Japan and was collected in August 2006. The WIDE-08 and WIDE-09 traces are taken from a US–Japan trans-Pacific backbone line (a 150-Mb.s Ethernet link) that carries commodity traffic for WIDE organizations. The original traces were collected as part of the "A Day in the Life of the Internet" project, which lasted 72 h from March 18 to 20, 2008, and 96 h from March 30 to April 4, 2009. Forty bytes of application-layer payload were kept for each packet, while all IP addresses were anonymized in KEIO and WIDE traces. In addition, the ISP data set is a trace we captured using a passive probe at a 100-Mb/s Ethernet edge link from an Internet service provider located in Australia. Full packet payloads are preserved in the collection without any filtering or packet loss. The trace is 7 days long and began on November 27, 2010.

TABLE I
TRAFFIC TRACES

Trace	Data	Duration	Link Type	Volume
KEIO	2006-08-06	30 mins	edge	16.99 GB
WIDE-08	2008-03-18	5 hours	backbone	197.2 GB
WIDE-09	2009-03-31	5 hours	backbone	224.2 GB
ISP	2010-11	7 days	edge	665.7 GB

Following the significant work of [3], [8], and [35], we focus exclusively on the vast majority of traffic (up to 95%) in the observed networks: TCP traffic. Note the proposed RTC scheme is independent to the transport-layer protocol. In consideration of practical uses, we adopt a 900-s idle timeout for flows terminated without a proper teardown. To establish the ground truth in datasets, we develop a DPI tool matching regular expression patterns against payloads. Two distinct sets of application signatures are developed based on previous experience and some well-known tools, such as the l7-filter (http://l7-filter.sourceforge.net) and Tstat (http://tstat.tlc.polito.it). The first set is designed to match against the full flow payload (for the ISP trace). For the remaining traces, in which only 40 B of payload are available for each packet, we tune the second set of signatures to match early message keywords. Some efforts of manual inspection were also made to investigate the encrypted and emerging applications.

We create a combined dataset to study the impact of various factors on traffic classification performance. Merging multiple real-world traces into one for evaluation can minimize the effects of data bias [3]. The combined dataset contains more classes than individual datasets, which is helpful in challenging the classification methods. Since we merged the traffic captured at various locations and time periods, the target applications display strong and different behaviors, which cannot be observed in individual traffic traces. Our work focuses on dealing with zero-day applications. To reduce the impact of class imbalance on experiments, four traffic traces were merged together to form the experiment dataset. Then, for the classes that contain more than 100 000 flows, we randomly sampled 100 000 flows of each class; for the classes that contains less than 100 000 flows, we included all flows of these classes in the experiment dataset. Unrecognized traffic of the DPI tool is excluded from the combined dataset. Finally, the combined dataset was constituted by over 638 000 traffic flows from 10 major traffic classes and 16 small traffic classes. Fig. 4 shows distribution of traffic classes.

In experiments, 20 unidirectional flow statistical features, as listed in Table II, were extracted to represent traffic flows. We applied feature selection to further remove irrelevant and redundant features from the feature set [46]. The process of feature selection yields nine features. These are client-to-server number of packets, client-to-server maximum packet bytes, client-to-server minimum packet bytes, client-to-server average packet bytes, the standard deviation of client-to-server packet bytes, client-to-server minimum interpacket time,

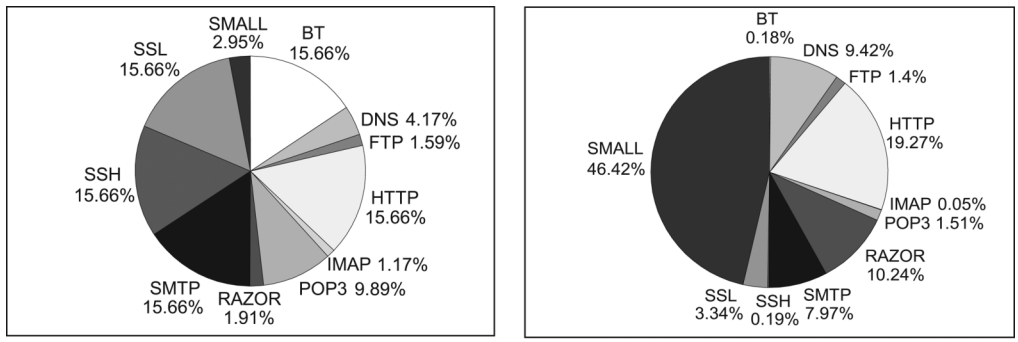


Fig. 4. Class distribution of the combined dataset. (a) Flow. (b) Byte.

TABLE II
UNIDIRECTIONAL STATISTICAL FEATURES

Type of features	Feature description	Number
Packets	Number of packets transferred in unidirection	2
Bytes	Volume of bytes transferred in unidirection	2
Packet Size	Min., Max., Mean and Std Dev. of packet size in unidirection	8
Inter-Packet Time	Min., Max., Mean and Std Dev. of Inter Packet Time in unidirection	8
	Total	20

server-to-client number of packets, server-to-client maximum packet bytes, and server-to-client minimum packet bytes.

During experiments, we simulated the problem of zero-day applications. On the combined dataset, we manually set a few major classes and all small classes to "unknown." In the experiments, the dataset was divided into four disjointed parts: a prelabeled set, an unlabeled set, and two testing sets. For known classes, a small percentage of flows were randomly selected to form a labeled training set. It is important to note that no samples of unknown classes were available for the classification system. Some flows were randomly selected from the unlabeled set and used in the RTC scheme and Erman's semi-supervised method. Two testing sets were used to evaluate the RTC scheme with or without a system update.

Furthermore, we also performed a number of experiments on individual datasets of ISP and WIDE-09, in which the traffic unrecognized by DPI were considered zero-day traffic. In these experiments, the unknown classes were not manually selected, which is different to the combined dataset.

B. Evaluation With Synthetic Zero-Day Traffic

1) Experiments and Goals: For performance evaluation, a large number of experiments were conducted on the combined dataset. We present the average performance of over 100 runs and also provide the error bars to show how the results were stable.

We compare the proposed RTC scheme with four state-of-the-art traffic classification methods: random forest [47], the BoF-based method [4], the semi-supervised method [35], and one-class SVM [8]. Note that features used in experiments were different to those in [35]. However, to be fair, all comparison methods/schemes used the nine selected features.

The proposed RTC scheme without system update was evaluated in experiments. We take random forest as a representative of conventional supervised traffic classification methods. In our empirical study, random forest displays superior performance over other supervised algorithms, such as k-NN and support vector machine. The BoF-based method [4] was able to effectively incorporate flow correlation into supervised classification. Our previous work shows the BoF-based method outperforms conventional supervised methods. We implemented the BoF-based method by employing the random forest algorithm and majority vote rule. In addition, we test Erman's semi-supervised method [35], which has the capability of unknown identification. Theoretically speaking, one-class SVM can avoid the problem of zero-day applications because it can train an SVM classifier for each known class. Ideally, the traffic rejected by all known classes is generated by unknown applications. Therefore, the modified one-class SVM [8] is also selected for our comparison study.

The proposed RTC scheme and Erman's semi-supervised method share two parameters: the number of clusters in k-means and the number of unlabeled training flows. In the empirical study, we used 30 000 unlabeled flows in the training set. According to our experimental results, we set $k = 2000$ for Erman's semi-supervised method in order to achieve its highest classification accuracy.

We developed an automatic method to select k in the proposed RTC scheme. The method of parameter setting combines a 10-fold cross validation and binary search described in detail in Section IV. Two common metrics were used to measure the traffic classification performance, accuracy, and F-measure.

Three sets of experiments were performed to compare the traffic classification performance of the five methods/schemes.

2) Impact of Labeled Training Data: Fig. 5 shows the overall accuracy of the five methods with various labeled training sizes. During experiments, the major classes of BT, DNS, and SMTP were set to unknown. The small classes work as noise to challenge the traffic classification methods. Therefore, the modified dataset includes seven known major classes and three unknown

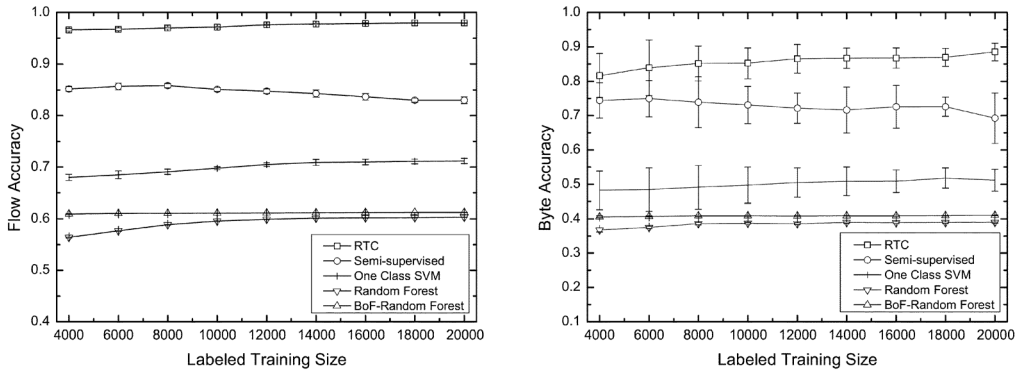


Fig. 5. Overall accuracy. (a) Flow. (b) Byte.

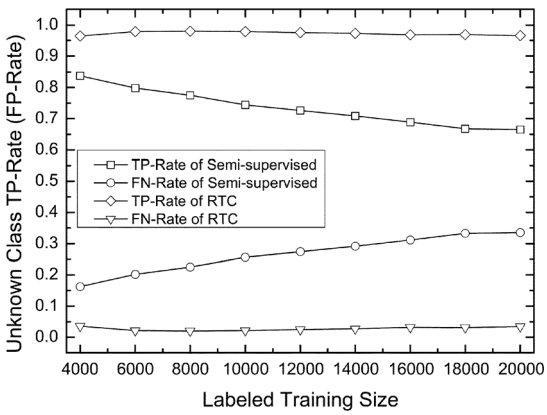


Fig. 6. Impact of labeled training data.

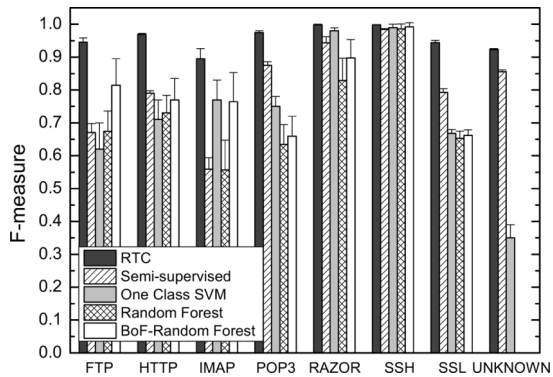


Fig. 7. F-measure of each application class.

major classes. The flows from the unknown classes compose zero-day traffic.

The size of supervised training data changes from 4000 to 20 000. The results show the proposed RTC scheme is significantly superior to the other four methods. The second best is the semi-supervised method. The accuracy difference between RTC and semi-supervised can reach 15%.

The accuracy of the other three methods—random forest, BoF-random forest, and one-class SVM—is poor. The cause of the low accuracy exhibited by BoF-random forest and random forest is the inaccurately classification of zero-day traffic into known classes. One-class SVM cannot produce a discriminative boundary in a multiclass space without a large amount of labeled training data. In addition, its unknown detection capability is limited without zero-day information.

An interesting observation was the accuracy of Erman's semi-supervised method slightly decreasing as the size of the labeled training data increased. To investigate the causes, we report the TPR and false negative rate (FNR) of zero-day sample detection, as shown in Fig. 6. TPR is the rate of the sum of correctly detected zero-day traffic compared to the sum of all actual zero-day traffic. FNR is the rate of the sum of zero-day traffic inaccurately detected as "known" compared to the sum of all actual zero-day traffic. The results of our RTC scheme are also shown for comparison.

We notice the number of clusters produced by k-means in semi-supervised is fixed to 2000. The results show that for Erman's method, as the labeled training flows increase in size, the true positive rate declines and the false negative rate quickly rises. This will significantly affect its unknown detection capability. Consequently, the overall accuracy of the semi-supervised method is limited and becomes worse. Our RTC scheme can successfully solve this problem by automatically optimizing k for different sizes of supervised training data. The figure shows the TPR and FNR of the RTC scheme has only slight changes.

In addition, we tested the classification speed of the five competing methods. The results (flows/second) were 3.2×10^4 for RTC, 4.5×10^3 for one-class SVM, 3.77×10^4 for BoF-random forest, 3.28×10^5 for random forest, and 6.8×10^3 for semi-supervised. In our experiments, the RTC scheme displays the comparable classification speed of existing methods.

3) Performance of Traffic Classes: Fig. 7 reports the flow F-measures from five competing traffic classification methods. In general, the results indicate the proposed RTC scheme significantly outperforms other methods when zero-day applications are present. Other methods do not work as well due to poor performance in predefined known classes or failure to identify zero-day traffic.

Let us further investigate the F-measures in each class. In class FTP, the F-measure of our scheme was higher than the second best method, BoF-random forest, by about 0.13.

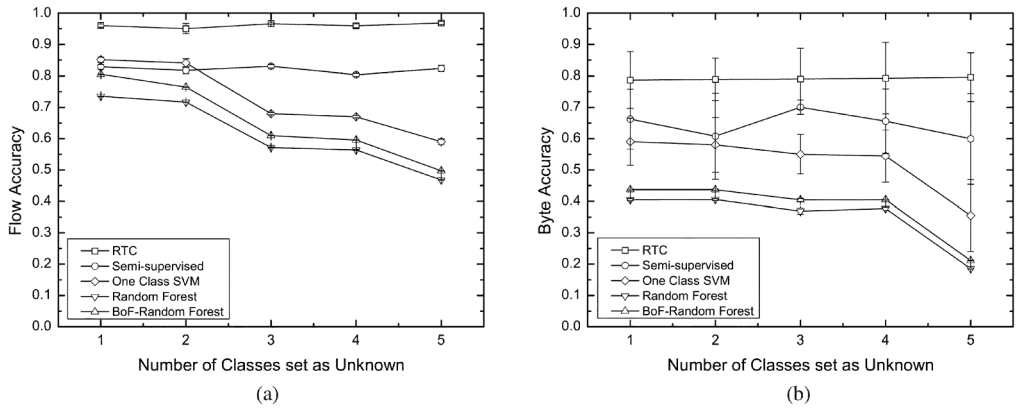


Fig. 8. Classification result with varying zero-day applications. (a) Flow. (b) Byte.

Random forest was slightly better than the semi-supervised method, however both were worse than our scheme by about 0.27. In class HTTP, the improvement of our scheme was about 0.18; with semi-supervised, the second best method, is about 0.18. There were no significant differences among methods of random forest, BoF-random forest, and semi-supervised. In class IMAP, the F-measure of our scheme achieved 0.9, which is higher by about 0.12 than the second best method, one-class SVM. In class POP3, the F-measure of our scheme was about 0.97. The F-measure of the second best method, semi-supervised, was about 0.87, which is much higher than the other three methods. In class RAZOR, the ranking list was our scheme, one-class SVM, semi-supervised, BoF-based, and random forest. In class SSH, all methods displayed excellent performance. In class SSL, the F-measure of our scheme was higher than the second best method, semi-supervised, by over 0.15. The performance of one-class SVM was similar to that of BoF-random forest and random forest. The three methods were less than the semi-supervised method by about 0.14. Finally, our scheme was superior to the methods semi-supervised and one-class SVM in terms of zero-day traffic identification. The difference of F-measures between our scheme and the second best method, semi-supervised, was 0.08. One-class SVM had very low zero-day traffic identification performance due to its poor classification boundary for zero-day applications.

We observed the superiority of the proposed RTC scheme was due to its excellent functionality of unknown discovery. As described in Section III-A, a new two-step unknown discovery was applied for robust traffic classification. The first step borrows the idea of the semi-supervised method to roughly detect some zero-day samples. The experimental results show the true positive rate of zero-day traffic detection in the first step was 72%, and the false positive rate was 6%. The second step constructs a random forest classifier by using the outcome of the first step, which can further improve the effectiveness of zero-day sample extraction. In the experiment, the true positive rate was raised to 94%, and the false positive rate was reduced to 3%. Thus, zero-day samples can be combined with prelabeled training data to train a super classifier that has the capability of identifying zero-day traffic.

4) Impact of Zero-Day Applications: Fig. 8 displays the impact of zero-day application classes to traffic classification performance. In this figure, we amplify the pool of zero-day traffic by adding one to five major classes. One can see the accuracy of RTC and semi-supervised was stable when the number of zero-day application classes increased. Meanwhile, the accuracy of one-class SVM, random forest, and BoF-random forest decreased dramatically.

These results further confirm the robustness of the proposed RTC scheme. In detail, RTC outperformed semi-supervised in terms of accuracy and reliability.

The accuracy of RTC is always significantly higher than semi-supervised, with a difference of proximately 12%. With a different number of zero-day applications, semi-supervised's accuracy changed by 3%, while for RTC, it was only 1% .

Compared to the supervised methods, random forest and BoF-random forest, RTC exhibited the excellent capability of dealing with zero-day traffic. However, the accuracy of supervised methods was strictly limited by the amount of traffic generated by known applications, which they can correctly classify. For example, the accuracy of BoF-random forest declined from 80% to 50% when the number of zero-day application classes increased from 1 to 5. The accuracy of one-class SVM was higher than random forest and BoF-random forest because it identified a small portion of zero-day traffic. However, one-class SVM has very limited zero-day traffic identification ability that cannot be improved by increasing the supervised training size. The reason is one-class SVM does not explore zero-day information in the classification procedure.

5) Performance of System Update: A set of experiments were carried out to evaluate the function of the system update. We tested the classification performance of our scheme, with and without a system update. In the experiments, the labeled and unlabeled training data consisted of 4000 and 30 000 flows, respectively. During the system update, the identified zero-day traffic was categorized into 100 clusters. We randomly selected three flows from each cluster and manually inspected them for new class construction. It was assumed the three unknown major classes could be recognized at this stage since their traffic was statistically significant. A two-level classification strategy was applied to perform traffic classification. An F-measure was used to evaluate the classification results.

Fig. 9 reports the F-measures of our scheme before and after the update. In this figure, the performance of the

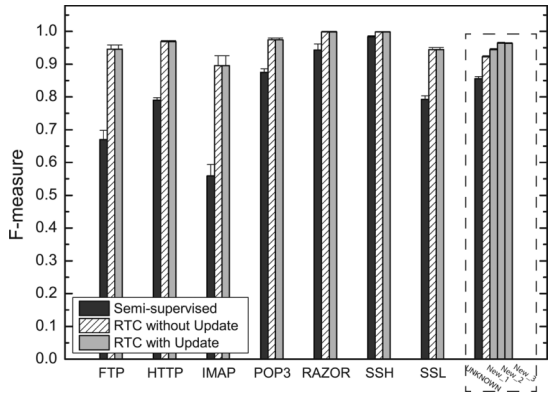


Fig. 9. Performance of system update.

semi-supervised method was used as the baseline. The results show the proposed RTC scheme with system update can achieve fine-grained classification of zero-day traffic. For example, zero-day traffic can be identified with a F-measure of 0.91 before an update. After an update, the zero-day traffic can be perfectly classified into three new classes. The F-measures of new classes, new_1(BT), new_2(DNS), and new_3(SMTP), can achieve about 0.94, 0.96, 0.96, respectively. In the known classes, the performance of our scheme did not change after the system update because of the two-level classification strategy. We can draw an initial conclusion that the system update can achieve fine-grained classification of zero-day traffic without affecting the performance of known classes.

In the experiments, there were about 60 000 flows identified as zero-day traffic. According to the experimental setting, the rate of manual inspection was $0.5\%[= (100 * 3)/60\,000]$. This rate was very low, thus making it possible for the practical use of the module for a system update. For example, in attack detection, fine-grained identification of zero-day traffic is well worth it and only uses minimal human effort.

C. Evaluation With DPI Unrecognized Traffic

We have used only DPI recognized flows to study the impact of different "unknown" settings on traffic classification. In this section, we report additional experiments and the results on individual datasets by considering DPI unrecognized traffic as zero-day traffic.

The experiments were carried out on ISP and WIDE-09 traffic traces. The ISP experiment dataset contained over 650 000 flows, with approximately 296 000 as zero-day traffic (i.e., unrecognized by DPI). We identified the known classes BT, DNS, EDONKEY, FTP, HTTP, IMAP, MSN, POP3, SMB, SMTP, SSH, SSL, and XMPP. The zero-day traffic constituted 55% of flows and 12% of bytes. In experiments on the ISP dataset, 4000 labeled flows and 30 000 unlabeled flows were randomly sampled for training. The WIDE-09 experiment dataset contained over 439 000 flows, in which about 158 000 were zero-day traffic. The known classes in WIDE-09 were BT, DNS, FTP, HTTP, POP3, SMTP, and SSL. The zero-day traffic constituted 36% of flows and 25% of bytes. In experiments on the WIDE-09 dataset, 2500 labeled flows and 20 000 unlabeled flows were randomly sampled for training.

Fig. 10 shows classification results on the ISP and WIDE-09 datasets. The flow and byte accuracy of traffic classification on the ISP are reported in Fig. 10(a). One can see RTC always displays the highest flow and byte accuracy among all competing methods. For flow accuracy, RTC is better than the second best method, semi-supervised, by about 10%. In addition, semi-supervised and one-class SVM significantly outperformed random forest and BoF-random forest. The differences are from 30% up to 50%. The byte accuracy of RTC was about 15% higher than the other four methods with a similar byte accuracy. It should be noted the byte accuracy was independent to the flow accuracy due to the presence of elephant and mice flows. The results on WIDE-09, as shown in Fig. 10(b), are similar to those on ISP. Regarding flow accuracy, RTC, semi-supervised, and one-class SVM, which have the potential to deal with zero-day applications, are much better than random forest and BoF-random forest. However, there are big differences among the byte accuracy of the five competing methods. RTC outperformed other methods by up to 25%.

VI. Discussion

A. Sub-Bag of Flows

Here, we present a further study on flow correlation in the context of traffic classification. As mentioned previously, (10) suggests the flow prediction error can be reduced by a factor of M by using a simple BoF-based model. For estimating E_{bof} in the experiments, M can be calculated by

$$M = n_{\mathrm{flow}}/n_{\mathrm{bof}} \tag{11}$$

where n_{flow} is the number of testing flows, and n_{bof} is the number of BoFs constructed by the testing flows. Unfortunately, E_{bof} in (10) depends on the key assumption that errors due to individual flows in any BoF are independent.

A novel factor of our study was to accurately estimate the reduction in the overall error when the flow errors were highly dependent in practice. We observed a number of sub-bags constitute a BoF. A sub-bag consists of flows sharing 4-tuples: source IP, destination IP, destination port, and transport protocol. One can see flows in a sub-bag are likely generated by the same user in a short period of time. The flows in a sub-bag have high dependency, while the flows in different sub-bags have low dependency. We propose M in (10) be replaced with the number of sub-bags in a BoF to alleviate the problem of error dependency. Equation (10) can be rewritten as

$$E'_{\mathrm{bof}} = \frac{1}{M'} E_{\mathrm{flow}}. \tag{12}$$

In practice, M' is the average number of sub-bags in a BoF. This can be calculated by

$$M' = n_{\mathrm{sbof}}/n_{\mathrm{bof}} \tag{13}$$

where n_{sbof} and n_{bof} are the number of sub-bags and the number of BoFs in the testing set. One can see E'_{bof} in (12) is estimated under the weak assumption that errors due to individual sub-bags are independent.

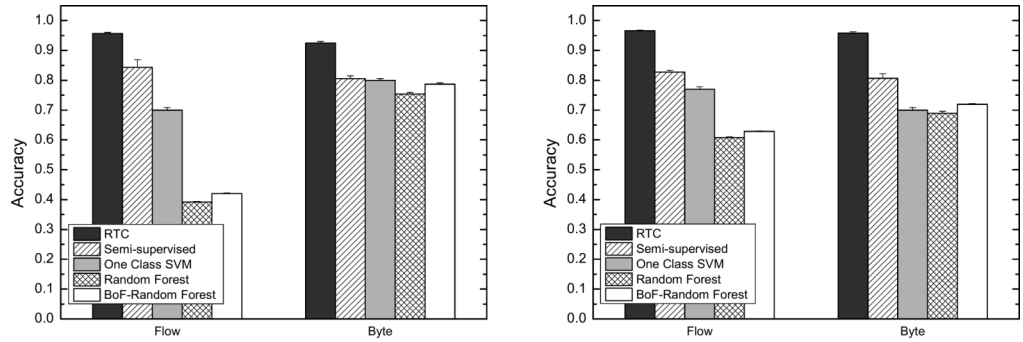


Fig. 10. Classification result with DPI unrecognized unknown traffic. (a) ISP. (b) WIDE-09.

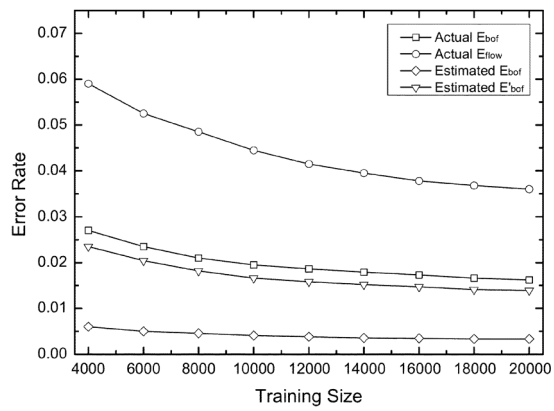


Fig. 11. Error estimation.

We perform a number of traffic classification experiments to verify the theoretical analysis. The experiments were conducted on the experimental dataset without considering zero-day traffic. Random forest was applied for supervised traffic classification. The BoF-based method was implemented by combining the random forest algorithm and the majority vote rule. The classification error was used to measure the traffic classification performance. Fig. 11 shows the actual error rates versus the estimated error rates. The results show the estimated BoF error rate using (12) can match the actual BoF error rate a lot better than the error rate estimated using (10). In other words, given the flow error rate E_{flow}, we can accurately estimate the BoF error rate according to the average number of sub-bags in a BoF. We observed that in the four real traffic traces, the average number of sub-bags in BoFs, M' was always larger than 2. Therefore, the BoF model can effectively incorporate flow correlation into traffic identification, thus strongly supporting the new scheme presented in this paper.

Based on the above analysis, one idea is to randomly select a flow to represent a sub-bag to speed up the proposed RTC scheme for practical applications. For example, there are 638 388 flows, 64 444 BoFs, and 165 858 sub-bags in our complex traffic dataset. If we apply the idea of sub-bag, our scheme needs to classify only 165 858 flows instead of the whole dataset (638 388 flows) before prediction aggregation. Therefore, the classification time may reduce to about one fourth of that used by the original scheme. We have evaluated the performance of the RTC scheme with and without considering sub-bags. The results show the classification performance has no significant decrease.

The RTC scheme can be used for real-time classification. We can directly incorporate the ideas of packet milestones [35] and subflows [13] into the RTC scheme. For example, a packet milestone is reached when the count of the total number of packets a flow sends or receives reaches a specific value. What we need to do is extract the statistical features on each packet milestone and train the corresponding RTC classifier. Moreover, we can further speed up traffic classification by considering sub-bags in the RTC scheme.

B. Classifier Retraining

Our work shares a basic assumption with most pattern classification algorithms in that class distribution will not change in the training and testing stages. However, in real-world networks, class distribution may change over a long period of time. For example, one of the N known applications changes, and a cluster appears in a different position to the space. According to the RTC scheme, a new cluster will be identified however this is related to an old application. Therefore, a new ψ_i is not added to the training set, i.e., the new characteristic of the application is not tracked. To address this issue, one possibility is to retrain the traffic classifier by incorporating new samples of old applications.

Erman *et al.* [35] suggested two measures for measuring reliability of classifiers that can be used to indicate when retraining is necessary. The first is the number of flows not assigned a label. If this number increases, it indicates a need for classifier retraining so underrepresented flow types can be captured and classification accuracy improved. The second measure is the average distance of new flows to their nearest cluster mean. A significant increase in the average distance indicates the need for retraining.

We plan to extend this work in the future and address the problem of changing class distribution by developing new strategies for system updates and classifier retraining. One idea is to count the flows of any known classes recognized by semi-automatic identification during a system update. If the number increases, this indicates class distributions of the corresponding known classes have changed and the traffic classifier should be retrained. In other words, when changed

class distributions or new classes are detected, the system update will be triggered.

VII. Conclusion

This paper addresses the new problem of zero-day applications in Internet traffic classification. Conventional traffic classification methods suffer from poor performance when zero-day applications are present due to misclassification of zero-day traffic into predefined known classes. We proposed a novel robust traffic classification scheme, RTC, which can identify zero-day traffic as well as accurately classify the traffic generated by predefined application classes. The proposed scheme has three important modules: unknown discovery, BoF-based traffic classification, and system update. In particular, we presented a formal analysis on the performance benefit of flow correlation compared to traffic classification. A new optimization method was developed to intelligently tune the parameter of the proposed RTC scheme. To evaluate the new scheme, a large number of well-designed experiments were carried out on real-world traffic traces. The results demonstrated that the proposed RTC scheme significantly outperformed four state-of-the-art methods.

References

[1] Cisco Systems, Inc., San Jose, CA, USA, "Cisco WAN and application optimization solution guide," Tech. Rep., 2008 [Online]. Available: http://www.cisco.com/c/en/us/td/docs/nsite/enterprise/wan/wan_optimization/wan_opt_sg.html

[2] T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, 4th Quart., 2008.

[3] H. Kim *et al.*, "Internet traffic classification demystified: myths, caveats, and the best practices," in *Proc. ACM CoNEXT Conf.*, 2008, pp. 1–12.

[4] J. Zhang *et al.*, "Network traffic classification using correlation information," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 104–117, Jan. 2013.

[5] A. Tongaonkar, R. Keralapura, and A. Nucci, "Challenges in network application identification," in *Proc. 5th USENIX Conf. Large-Scale Exploits Emergent Threats*, 2012, pp. 1–3.

[6] A. Moore and D. Zuev, "Internet traffic classification using Bayesian analysis techniques," *Perform. Eval. Rev.*, vol. 33, no. 1, pp. 50–60, 2005.

[7] T. Auld, A. Moore, and S. Gull, "Bayesian neural networks for Internet traffic classification," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 223–239, Jan. 2007.

[8] A. Este, F. Gringoli, and L. Salgarelli, "Support vector machines for TCP traffic classification," *Comput. Netw.*, vol. 53, no. 14, pp. 2476–2490, 2009.

[9] J. Erman, A. Mahanti, M. Arlitt, and C. Williamson, "Identifying and discriminating between web and peer-to-peer traffic in the network core," in *Proc. Int. Conf. World Wide Web*, 2007, pp. 883–892.

[10] L. Bernaille and R. Teixeira, "Early recognition of encrypted applications," in *Proc. Passive Active Netw. Meas.*, 2007, pp. 165–175.

[11] B. Hullár, S. Laki, and A. Gyorgy, "Early identification of peer-to-peer traffic," in *Proc. IEEE Int. Conf. Commun.*, 2011, pp. 1–6.

[12] T. Nguyen and G. Armitage, "Training on multiple sub-flows to optimise the use of machine learning classifiers in real-world IP networks," in *IEEE Conf. Local Comput. Netw.*, 2006, pp. 369–376.

[13] T. Nguyen, G. Armitage, P. Branch, and S. Zander, "Timely and continuous machine-learning-based classification for interactive IP traffic," *IEEE/ACM Trans. Netw.*, vol. 20, no. 6, pp. 1880–1894, Dec. 2012.

[14] P. Bermolen, M. Mellia, M. Meo, D. Rossi, and S. Valenti, "Abacus: Accurate behavioral classification P2P-TV traffic," *Comput. Netw.*, vol. 55, no. 6, pp. 1394–1411, 2011.

[15] E. Glatz and X. Dimitropoulos, "Classifying Internet one-way traffic," in *Proc. ACM SIGMETRICS/PERFORMANCE Joint Int. Conf. Meas. Model. Comput. Syst.*, 2012, pp. 417–418.

[16] Y. Jin *et al.*, "A modular machine learning system for flow-level traffic classification in large networks," *Trans. Knowl. Discov. Data*, vol. 6, no. 1, pp. 4:1–4:34, 2012.

[17] A. Callado, J. Kelner, D. Sadok, C. A. Kamienski, and S. Fernandes, "Better network traffic identification through the independent combination of techniques," *J. Netw. Comput. Appl.*, vol. 33, no. 4, pp. 433–446, 2010.

[18] V. Carela-Español, P. Barlet-Ros, A. Cabellos-Aparicio, and J. Solé-Pareta, "Analysis of the impact of sampling on netflow traffic classification," *Compu. Netw.*, vol. 55, no. 5, pp. 1083–1099, 2011.

[19] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing Skype traffic: when randomness plays with you," *Comput. Commun. Rev.*, vol. 37, no. 4, pp. 37–48, 2007.

[20] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Traffic classification through simple statistical fingerprinting," *Comput. Commun. Rev.*, vol. 37, pp. 5–16, 2007.

[21] S. Valenti, D. Rossi, M. Meo, M. Mellia, and P. Bermolen, "Accurate, fine-grained classification P2P-TV applications by simply counting packets," in *Proc. 1st Int. Workshop Traffic Monitoring Anal.*, 2009, pp. 84–92.

[22] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, "Flow clustering using machine learning techniques," in *Proc. Passive Active Netw. Meas.*, 2004, pp. 205–214.

[23] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," in *Proc. Annu. IEEE Conf. Local Comput. Netw.*, 2005, pp. 250–257.

[24] J. Erman, A. Mahanti, and M. Arlitt, "Internet traffic identification using machine learning," in *Proc. IEEE Global Telecommun. Conf.*, 2006, pp. 1–6.

[25] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proc. SIGCOMM Workshop Mining Netw. Data*, 2006, pp. 281–286.

[26] D. Liu and C. Lung, "P2P traffic identification and optimization using fuzzy c-means clustering," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2011, pp. 2245–2252.

[27] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *Comput. Commun. Rev.*, vol. 36, pp. 23–26, 2006.

[28] Y. Wang, Y. Xiang, and S.-Z. Yu, "An automatic application signature construction system for unknown traffic," *Concurrency Comput., Pract. Exper.*, vol. 22, no. 13, pp. 1927–1944, 2010.

[29] A. Finamore, M. Mellia, and M. Meo, "Mining unclassified traffic using automatic clustering techniques," *Traffic Monitoring Anal.*, vol. 6613, pp. 150–163, 2011.

[30] J. Ma, K. Levchenko, C. Kreibich, S. Savage, and G. M. Voelker, "Unexpected means of protocol inference," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, 2006, pp. 313–326.

[31] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-service mapping for QoS: A statistical signature-based approach to IP traffic classification," in *Proc. 4th ACM SIGCOMM Conf. Internet Meas.*, 2004, pp. 135–148.

[32] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification," *Comput. Commun. Rev.*, vol. 36, pp. 5–16, 2006.

[33] Y. Lim *et al.*, "Internet traffic classification demystified: on the sources of the discriminative power," in *Proc. ACM CoNEXT Conf.*, 2010, pp. 9:1–9:12.

[34] S. Lee *et al.*, "Netramark: a network traffic classification benchmark," *Comput. Commun. Rev.*, vol. 41, no. 1, pp. 22–30, 2011.

[35] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Offline/realtime traffic classification using semi-supervised learning," *Perform. Eval.*, vol. 64, no. 9, pp. 1194–1213, 2007.

[36] P. Casas, J. Mazel, and P. Owezarski, "MINETRAC: Mining flows for unsupervised analysis & semi-supervised classification," in *Proc. 23rd Int. Teletraffic Congr.*, 2011, pp. 87–94.

[37] H. Liu, Z. Wang, and Y. Wang, "Semi-supervised encrypted traffic classification using composite features set," *J. Netw.*, vol. 7, no. 8, pp. 1195–1200, 2012.

[38] B. Schlkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.

[39] G. Xie, M. Iliofotou, R. Keralapura, M. Faloutsos, and A. Nucci, "Subflow: Towards practical flow-level traffic classification," in *Proc. IEEE INFOCOM*, 2012, pp. 2541–2545.

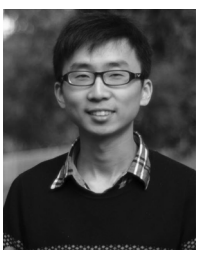
[40] D. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[41] Y. Wang, Y. Xiang, J. Zhang, and S.-Z. Yu, "A novel semi-supervised approach for network traffic clustering," in *Proc. Int. Conf. Netw. Syst. Security*, 2011, pp. 169–175.
[42] M. Baldi, A. Baldini, N. Cascarano, and F. Risso, "Service-based traffic classification: Principles and validation," in *Proc. IEEE SARNOFF*, 2009, pp. 1–6.
[43] N. Cascarano *et al.*, "Comparing p2ptv traffic classifiers," in *Proc. IEEE ICC*, 2010, pp. 1–6.
[44] C. Bishop *et al., Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
[45] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.
[46] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learning Res.*, vol. 3, pp. 1157–1182, 2003.
[47] L. Breiman, "Random forests," *Mach. Learning*, vol. 45, no. 1, pp. 5–32, 2001.

Jun Zhang (M'12) received the Ph.D. degree in computer science from the University of Wollongong, Wollongong, Australia, in 2011.

He is currently with the School of Information Technology, Deakin University, Melbourne, Australia. He has published more than 50 research papers in refereed international journals and conferences, such as the IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS—PART B, and IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT. His research interests include network and system security, pattern recognition, and multimedia retrieval.

Dr. Zhang received the 2009 Chinese government award for outstanding self-financed student abroad.



Xiao Chen (S'13) received the M.IT. degree from The University of Melbourne, Melbourne, Australia, in 2012, and is currently pursuing the Ph.D. degree in computer science at Deakin University, Melbourne, Australia.

His research interests include network traffic classification and online social networks.

Yang Xiang (A'08–M'09–SM'12) received the Ph.D. degree in computer science from Deakin University, Melbourne, Australia.

He is currently a Full Professor with the School of Information Technology, Deakin University. He is the Director of the Network Security and Computing Lab (NSCLab) and the Associate Head of School (Industry Engagement). He is the Chief Investigator of several projects in network and system security, funded by the Australian Research Council (ARC). He has published more than 150 research papers in many international journals and conferences. Two of his papers were selected as the featured articles in the April 2009 and the July 2013 issues of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. He has published two books, *Software Similarity and Classification* (Springer, 2012) and *Dynamic and Advanced Data Mining for Progressing Technological Development* (IGI-Global, 2009). His research interests include network and system security, distributed systems, and networking.

Prof. Xiang has served as the Program/General Chair for many international conferences. He serves as an Associate Editor of the IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and *Security and Communication Networks*, and an Editor of the *Journal of Network and Computer Applications*. He is the Coordinator, Asia, for the IEEE Computer Society Technical Committee on Distributed Processing (TCDP).

Wanlei Zhou (M'92–SM'09) received the B.Eng. and M.Eng. degrees in computer science and engineering from Harbin Institute of Technology, Harbin, China, in 1982 and 1984, respectively, the Ph.D. degree in computer science from The Australian National University, Canberra, Australia, in 1991, and the D.Sc. degree from Deakin University, Melbourne, Australia, in 2002.

He is currently the Alfred Deakin Professor (the highest honor the university can bestow on a member of academic staff) Chair Professor in information technology and head of the School of Information Technology, Deakin University, Melbourne, Australia. Before joining Deakin University, he worked with a number of organizations including the University of Electronic Science and Technology of China, Chengdu, China; Apollo/HP, Chelmsford, MA, USA; National University of Singapore, Singapore; and Monash University, Melbourne, Australia. He has published more than 280 papers in refereed international journals and refereed international conferences proceedings. He has also chaired many international conferences. His research interests include network security, distributed and parallel systems, bioinformatics, mobile computing, and e-learning.

Jie Wu (F'09) received the Ph.D. degree in computer engineering from Florida Atlantic University, Boca Raton, FL, USA, in 1989.

He is the Chair and a Laura H. Carnell Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. Prior to joining Temple University, he was a Program Director with the National Science Foundation and Distinguished Professor with Florida Atlantic University. He regularly publishes in scholarly journals, conference proceedings, and books. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications.

Dr. Wu serves on several editorial boards, including the IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON SERVICE COMPUTING, and *Journal of Parallel and Distributed Computing*. He was General Co-Chair/Chair for IEEE MASS 2006, IEEE IPDPS 2008, and IEEE ICDCS 2013, as well as Program Co-Chair for IEEE INFOCOM 2011 and CCF CNCC 2013. Currently, he is serving as General Chair for ACM MobiHoc 2014. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and Chair for the IEEE Technical Committee on Distributed Processing (TCDP). He is a CCF Distinguished Speaker. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.