Network Traffic Classification Using Correlation Information

Jun Zhang, *Member*, *IEEE*, Yang Xiang, *Member*, *IEEE*, Yu Wang, Wanlei Zhou, *Senior Member*, *IEEE*, Yong Xiang, *Senior Member*, *IEEE*, and Yong Guan, *Member*, *IEEE*

Abstract—Traffic classification has wide applications in network management, from security monitoring to quality of service measurements. Recent research tends to apply machine learning techniques to flow statistical feature based classification methods. The nearest neighbor (NN)-based method has exhibited superior classification performance. It also has several important advantages, such as no requirements of training procedure, no risk of overfitting of parameters, and naturally being able to handle a huge number of classes. However, the performance of NN classifier can be severely affected if the size of training data is small. In this paper, we propose a novel nonparametric approach for traffic classification, which can improve the classification performance effectively by incorporating correlated information into the classification process. We analyze the new classification approach and its performance benefit from both theoretical and empirical perspectives. A large number of experiments are carried out on two real-world traffic data sets to validate the proposed approach. The results show the traffic classification performance can be improved significantly even under the extreme difficult circumstance of very few training samples.

Index Terms—Traffic classification, network operations, security

1 Introduction

Network traffic classification has drawn significant attention over the past few years [1], [2], [3], [4], [5]. Classifying traffic flows by their generation applications plays very important role in network security and management, such as quality of service (QoS) control, lawful interception and intrusion detection [6]. Traditional traffic classification methods [1], [7], [2] include the port-based prediction methods and payload-based deep inspection methods. In current network environment, the traditional methods suffer from a number of practical problems, such as dynamic ports and encrypted applications. Recent research efforts have been focused on the application of machine learning techniques to traffic classification based on flow statistical features [2]. Machine learning can automatically search for and describe useful structural patterns in a supplied traffic data set, which is helpful to intelligently conduct traffic classification [8], [7]. However, the problem of accurate classification of current network traffic based on flow statistical features has not been solved.

The flow statistical feature-based traffic classification can be achieved by using supervised classification algorithms or unsupervised classification (clustering) algorithms [2]. In unsupervised traffic classification, it is very difficult to construct an application oriented traffic classifier by using the clustering results without knowing the real traffic classes [9], [10]. Given a set of prelabeled training data, the supervised traffic classification can be divided into two categories: parametric classifiers and nonparametric classifiers. Parametric classifiers, such as C4.5 decision tree [11], Bayesian network [11], SVM [3], and neural networks [12], require an intensive training procedure for the classifier parameters. Nonparametric classifiers, e.g., k-Nearest Neighbor (k-NN) [13], usually require no training phase and make classification decision based on closest training samples in the feature space [14]. When $k = 1$, the NN-based traffic classifier assigns a testing traffic flow into the class of its nearest training sample. As reported in [3], the NN classifier can achieve superior performance similar to that of the parametric classifiers, SVM and neural nets. They are the top three out of seven evaluated machine learning algorithms. In contrast to the parametric classifiers, the NN classifier has several important advantages [14]. For example, it does not require training procedure, immunizes overfitting of parameters and is able to handle a huge number of classes. In this point of view, the NN classifier is more suitable for traffic classification in current complex network environment.

However, the performance of the NN classifier is severely affected by a small size of training data which cannot accurately represent the traffic classes. We have observed that the classification accuracy of the NN-based traffic classifier decreases by approximate 20 percents when the number of training samples reduces from 100 to 10 for each class (see Section 3.1 for detail). Other supervised classification methods, such as SVM and neural nets, are not robust to training data size either. In practical, we may only manually label very few samples as supervised training data since traffic labeling is time consuming, especially for encrypted applications. It is essential that traffic classification can work with very few manually labeled training samples for some

- *J. Zhang, Y. Xiang, Y. Wang, W. Zhou, and Y. Xiang are with the School of Information Technology, Deakin University, 221 Burwood Highway, Burwood Victoria 3125, Australia. E-mail: {jun.zhang, yang.xiang, y.wang, wanlei, yong.xiang}@deakin.edu.au.*
- *Y. Guan is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011. E-mail: yguan@iastate.edu.*

Manuscript received 2 Oct. 2011; revised 26 Jan. 2012; accepted 1 Mar. 2012; published online 9 Mar. 2012.
Recommended for acceptance by K. Wu.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2011-10-0744.
Digital Object Identifier no. 10.1109/TPDS.2012.98.

specific purposes. This observation and the need for such traffic classifiers motivate our work.

In this paper, we propose a new framework, **Traffic Classification** using **Correlation** (TCC) information, to address the problem of very few training samples. The correlation information in network traffic can be used to effectively improve the classification accuracy. The major contributions of this work are summarized as follows:

- We propose a novel nonparametric approach which incorporates correlation of traffic flows to improve the classification performance.
- We provide a detailed analysis on the novel classification approach and its performance benefit from both theoretical and empirical aspects.
- The performance evaluation shows that the traffic classification using very few training samples can be significantly improved by our approach.

All data related to this work are available at http://anss.org.au/tc.

The remainder of the paper is organized as follows: Section 2 reviews related work in traffic classification. A novel classification approach and the theoretical analysis are proposed in Section 3. Section 4 presents a large number of experiments and results for performance evaluation. Some discussions related to this work are provided in Section 5. Finally, the paper is concluded in Section 6.

2 Related Work

In the last decade, considerable research works were reported on the application of machine learning techniques to traffic classification. These works can be categorized as supervised methods or unsupervised methods.

2.1 Supervised Methods

The supervised traffic classification methods analyze the supervised training data and produce an inferred function which can predict the output class for any testing flow. In supervised traffic classification, sufficient supervised training data is a general assumption. To address the problems suffered by payload-based traffic classification, such as encrypted applications and user data privacy, Moore and Zuev [7] applied the supervised naive Bayes techniques to classify network traffic based on flow statistical features. Williams et al. [11] evaluated the supervised algorithms including naive Bayes with discretization, naive Bayes with kernel density estimation, C4.5 decision tree, Bayesian network, and naive Bayes tree. Nguyen and Armitage [15] proposed to conduct traffic classification based on the recent packets of a flow for real-time purpose. Auld et al. [12] extended the work of [7] with the application of Bayesian neural networks for accurate traffic classification. Erman et al. [16] used unidirectional statistical features for traffic classification in the network core and proposed an algorithm with the capability of estimating the missing features. Bernaille and Teixeira [17] proposed to use only the size of the first packets of an SSL connection to recognize the encrypted applications. Bonfiglio et al. [18] proposed to analyze the message content randomness introduced by the encryption processing using Pearson's chi-Square test-based technique. Crotti et al. [19] proposed the probability density function (PDF)-based protocol fingerprints to express three traffic statistical properties in a compact way. Their work is extended with a parametric optimization procedure [20]. Este et al. [21] applied one-class SVMs to traffic classification and presented a simple optimization algorithm for each set of SVM working parameters. Valenti et al. [22] proposed to classify P2P-TV traffic using the count of packets exchanged with other peers during the small time windows. Pietrzyk et al. [23] evaluated three supervised methods for an ADSL provider managing many points of presence, the results of which are comparable to deep inspection solutions. These works use parametric machine learning algorithms, which require an intensive training procedure for the classifier parameters and need the retraining for new discovered applications.

There are a few works using nonparametric machine learning algorithms. Roughan et al. [13] have tested NN and LDA methods for traffic classification using five categories of statistical features. Kim et al. [3] extensively evaluated ports-based CorelReef method, host behavior-based BLINC method and seven common statistical feature-based methods using supervised algorithms on seven different traffic traces. The performance of the NN-based traffic classifier is comparable to two outstanding parametric classifiers, SVM and neural nets [3]. Although nonparametric methods have several important advantages which are not shared by parametric methods, their capabilities have been considered undervalued in the area of traffic classification.

Besides, supervised learning has also been applied to payload-based traffic classification. Although traffic classification by searching application signatures in payload content is more accurate, deriving the signatures manually is very time consuming. To address this problem, Haffner et al. [8] proposed to apply the supervised learning algorithms to automatically identify signatures for a range of applications. Finamore et al. [24] proposed application signatures using statistical characterization of payload and applied supervised algorithms, such as SVM, to conduct traffic classification. Similar to the supervised methods based on flow statistical features, these payload-based methods require sufficient supervised training data.

2.2 Unsupervised Methods

The unsupervised methods (or clustering) try to find cluster structure in unlabeled traffic data and assign any testing flow to the application-based class of its nearest cluster. McGregor et al. [25] proposed to group traffic flows into a small number of clusters using the expectation maximization (EM) algorithm and manually label each cluster to an application. Zander et al. [26] used AutoClass to group traffic flows and proposed a metric called intraclass homogeneity for cluster evaluation. Bernaille et al. [9] applied the k-means algorithm to traffic clustering and labeled the clusters to applications using a payload analysis tool. Erman et al. [27] evaluated the k-means, DBSCAN and AutoClass algorithms for traffic clustering on two empirical data traces. The empirical research showed that traffic clustering can produce high-purity clusters when the number of clusters is set as much larger than the number of real applications. Generally, the clustering techniques can be used to discover traffic from previously unknown

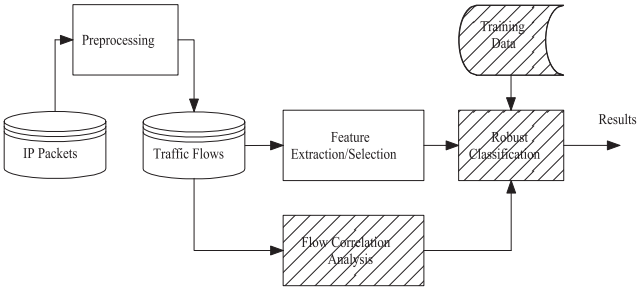
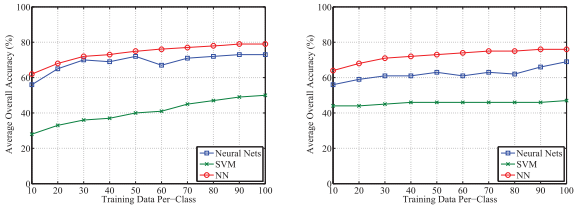


Fig. 1. A new traffic classification system model.



(a) on *wide* dataset (b) on *isp* dataset

Fig. 2. Impact of training data size.

applications [28]. Wang et al. [29] proposed to integrate statistical feature-based flow clustering with payload signature matching method, so as to eliminate the requirement of supervised training data. Finamore et al. [30] combined flow statistical feature-based clustering and payload statistical feature-based clustering for mining unidentified traffic. However, the clustering methods suffer from a problem of mapping from a large number of clusters to real applications. This problem is very difficult to solve without knowing any information about real applications.

Erman et al. [10] proposed to use a set of supervised training data in an unsupervised approach to address the problem of mapping from flow clusters to real applications. However, the mapping method will produce a large proportion of "unknown" clusters, especially when the supervised training data is very small. In this paper, we study the problem of supervised traffic classification using very few training samples. From the supervised learning point of view, several supervised samples are available for each class. Without the process of unsupervised clustering, the mapping between clusters and applications can be avoided. Our work focuses on nonparametric classification methods and address the difficult problem of traffic classification using very few training samples. The motivations are twofold. First, as mentioned in Section 1, nonparametric NN method has three important advantages which are suitable for traffic classification in current complex network situation. Second, labeling training data is time consuming and the capability of classification using very few training sample is very useful.

3 A Traffic Classification Approach with Flow Correlation

This section presents a new framework which we call **Traffic Classification** using **Correlation** information or ***TCC*** for short. A novel nonparametric approach is also proposed to effectively incorporate flow correlation information into the classification process.

3.1 System Model

Fig. 1 shows the proposed system model. In the preprocessing, the system captures IP packets crossing a computer network and constructs traffic flows by IP header inspection. A flow consists of successive IP packets having the same five-tuple: $\{src\_ip, src\_port, dst\_ip, dst\_port, protocol\}$. After that, a set of statistical features are extracted to represent each flow. Feature selection aims to select a subset of relevant features for building robust classification models. Flow correlation analysis is proposed to correlate information in the traffic flows. Finally, the robust traffic classification engine classifies traffic flows into application-based classes by taking all information of statistical features and flow correlation into account.

We observe that the accuracy of conventional traffic classification methods are severely affected by the size of training data. Fig. 2 reports the average overall accuracy of three classification algorithms [31] when a small size of training data is available. The experimental conditions are described in detail in Section 4. The classification performance of all algorithms are very poor when only 10 or 20 training samples are available for each class. In our experiments, NN classifier has the best classification performance. However, in the case of 10 training samples, the average overall accuracy of NN classifier is only about 60 percent on two data sets, which is very low.

The novelty of our system model is to discover correlation information in the traffic flows and incorporate it into the classification process. Conventional supervised classification methods treat the traffic flows as the individual and independent instances. They do not take the correlation among traffic flows into account. We argue that the correlation information can significantly improve the classification performance, especially when the size of training data is very small. In the proposed system model, flow correlation analysis is a new component for traffic classification which takes the role of correlation discovery. Robust classification methods can use the correlation information as input.

In this paper, we use "bag of flows" (BoF) to model correlation information in traffic flows.

- *A BoF consists of some correlated traffic flows which are generated by the same application.*

A BoF can be described by

$$Q = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, \tag{1}$$

where $\mathbf{x}_i$ is a feature vector representing the ith flow in the BoF Q. The BoF Q explicitly denotes the correlation among n flows, $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. The power of modeling correlation information with a bag has been demonstrated in our preliminary work for image ranking [32]. In this paper, the proposed flow correlation analysis will produce and analyze a large number of BoFs (see Section 3.3). A robust classification method should be able to deal with BoFs instead of individual flows. We will comprehensively study traffic classification with the BoF model from both theoretical and empirical perspectives.

3.2 Probabilistic Framework

In this section, we present a probabilistic framework for BoF model-based traffic classification. Given a BoF as the query, $Q = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, all flows in the BoF Q will be classified into the predicted class for Q.

According to the Bayesian decision theory [14], the maximum-a-posteriori (MAP) classifier aims to minimize the average classification error. For the query Q, the optimal class given by the MAP classifier is $\omega^* = \arg\max_\omega P(\omega|Q)$. With the assumption of uniform prior $P(\omega)$, we have the Maximum-Likelihood (ML) classifier

$$\omega^* = \arg\max_\omega P(\omega|Q) = \arg\max_\omega p(Q|\omega). \tag{2}$$

We consider the Naive-Bayes assumption in this study: $p(Q|\omega) = p(\mathbf{x}_1, \ldots, \mathbf{x}_n|\omega) = \prod_{\mathbf{x}\in Q} p(\mathbf{x}|\omega)$. And the log probability of the ML classifier is

$$\begin{aligned}\omega^* &= \arg\max_\omega \log(p(Q|\omega)) \\ &= \arg\max_\omega \frac{1}{\|Q\|}\sum_{\mathbf{x}\in Q}\log(p(\mathbf{x}|\omega)).\end{aligned} \tag{3}$$

Taking practical use into account, we uses an NN classifier to approximate the above optimal MAP Naive-Bayes classifier [33]. First, the Parzen likelihood estimation $\hat{p}(\mathbf{x}|\omega)$ is:

$$\hat{p}(\mathbf{x}|\omega) = \frac{1}{\|\omega\|}\sum_{\mathbf{x}'\in\omega} K(\mathbf{x}-\mathbf{x}'), \tag{4}$$

where $K(\cdot)$ is a Parzen kernel function and $\mathbf{x}'$ is a supervised training sample. We choose the Gaussian function for this study

$$K(\mathbf{x}-\mathbf{x}') = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{x}'\|^2\right). \tag{5}$$

The summation in (4) can be approximated using the r largest elements in the sum, which correspond to the r nearest neighbors of a flow $\mathbf{x} \in Q$ within the training data of the class ω

$$\hat{p}(\mathbf{x}|\omega) = \frac{1}{\|\omega\|}\sum_{j=1}^{r} K(\mathbf{x}-\mathbf{x}'_{NN_j}). \tag{6}$$

When the kernel function is Gaussian and $r = 1$, one can obtain

$$\log(\hat{p}(\mathbf{x}|\omega)) = -\frac{1}{2\sigma^2\|\omega\|}\min_{\mathbf{x}'\in\omega}\|\mathbf{x}-\mathbf{x}'\|^2, \tag{7}$$

and

$$\log(\hat{p}(Q|\omega)) = -\frac{1}{2\|Q\|\sigma^2\|\omega\|}\sum_{\mathbf{x}\in Q}\min_{\mathbf{x}'\in\omega}\|\mathbf{x}-\mathbf{x}'\|^2. \tag{8}$$

With the assumption of uniform prior $P(\omega)$, the scale parameter $\frac{1}{2\sigma^2\|\omega\|}$ will not affect the classification result. We have $\log(\hat{p}(Q|\omega)) \propto -\frac{1}{\|Q\|}\sum_{\mathbf{x}\in Q}\min_{\mathbf{x}'\in\omega}\|\mathbf{x}-\mathbf{x}'\|^2$. Therefore, the classifier for BoFs is

$$\begin{aligned}\omega^* &= \arg\max_\omega \log(p(Q|\omega)) \\ &= \arg\min_\omega \frac{1}{\|Q\|}\sum_{\mathbf{x}\in Q}\min_{\mathbf{x}'\in\omega}\|\mathbf{x}-\mathbf{x}'\|^2.\end{aligned} \tag{9}$$

Equation (9) shows a new nonparametric approach for BoF model-based traffic classification, which is derived from the Bayesian decision theory.

3.3 Correlation Analysis

We conduct correlation analysis using a three-tuple heuristic, which can quickly discover BoFs in the real traffic data.

- Three-tuple heuristic: in a certain period of time, the flows sharing the same three-tuple $\{dst\_ip, dst\_port, protocol\}$ form a BoF.

The correlated flows sharing the same three-tuple are generated by the same application. 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. The three-tuple heuristic about flow correlation has been considered in several practical traffic classification schemes [34], [35], [36]. Ma et al. [34] proposed a payload-based clustering method for protocol inference, in which they grouped flows into equivalence clusters using the heuristic. Canini et al. [35] tested the correctness of the three-tuple heuristic with real-world traces. In our previous work [36], we applied the heuristic to improve unsupervised traffic clustering. In this paper, we use BoF to model the correlation information obtained by the three-tuple heuristic and study the BoF model-based supervised classification, which is different from the exiting works [34], [35], [36]. Our new research problem is how to effectively use the correlation information in a supervised classification framework, which has been addressed in Section 3.2 from the theoretical perspective.

We measure the efficiency of the BoF discovery method by calculating some statistics of five real-world traffic data sets. The traffic data sets cover a wide range of link types such as backbone, internal, edge and wireless. The *sigcomm* traces [37] contains a detailed trace of wireless network activity at *SIGCOMM* 2008. The *lbnl* traces [38] are captured at two internal network locations of the Lawrence Berkeley National Laboratory in America. The *keio* trace [39] is collected from a 1 Gbps Ethernet link in Keio University Shonan-Fujisawa campus in Japan. The *wide* traces [39] are captured at a 150 Mbps Ethernet trans-Pacific backbone link that carries commodity traffic for the WIDE member organizations. The *isp* trace is a full payload traffic data set we collected at a 100 Mbps edge link of a small-medium ISP located in Melbourne, Australia from 11/2010 to 12/2010. For all the data sets, we focus exclusively on TCP traffic in this work and leave the non-TCP traffic for future work.

The statistical results are reported in Table 1. From the results, we observe that the correlation information is widely available in real network traffic. For instance, correlation occurs among 98 percent of flows in *wide* data set and 99 percent of flows in *isp* data set. The correlation information is valuable and can be exploited to enhance the performance of traffic classification.

TABLE 1
Statistics of Traffic Data Sets

Dataset	TCP Flows	Correlated Flows
sigcomm	7k	88%
lbnl	14k	91%
keio	170k	98%
wide	182k	98%
isp	765k	99%

3.4 Performance Benefit

We study performance benefit of the proposed approach by providing the theoretical and empirical analysis in a binary classification case. The analysis can be extended to the multiclass classification case in a straightforward way. Fig. 3 illustrates the performance benefit using both simulation data and real network traffic data.

Considering a binary classification problem, the decision rule of the NN classifier [14] for two classes, ω_1 and ω_2, is

$$\omega^* = \begin{cases} \omega_1, & \text{for } \min\limits_{\mathbf{x}'\in\omega_1}\|\mathbf{x}-\mathbf{x}'\|^2 < \min\limits_{\mathbf{x}'\in\omega_2}\|\mathbf{x}-\mathbf{x}'\|^2 \\ \omega_2, & \text{for } \min\limits_{\mathbf{x}'\in\omega_1}\|\mathbf{x}-\mathbf{x}'\|^2 > \min\limits_{\mathbf{x}'\in\omega_2}\|\mathbf{x}-\mathbf{x}'\|^2, \end{cases} \tag{10}$$

where $\min_{\mathbf{x}'\in\omega_i}\|\mathbf{x}-\mathbf{x}'\|^2, i\in\{1,2\}$ is the distance of $\mathbf{x}$ to class ω_i. We define the "**distance divergence**" of $\mathbf{x}$ as

$$\delta_{\mathbf{x}} = \min_{\mathbf{x}'\in\omega_1}\|\mathbf{x}-\mathbf{x}'\|^2 - \min_{\mathbf{x}'\in\omega_2}\|\mathbf{x}-\mathbf{x}'\|^2. \tag{11}$$

The value of $\delta_{\mathbf{x}}$ determines which class $\mathbf{x}$ is close to. Accordingly, the decision rule of the NN classifier expressed by (10) becomes

$$\omega^* = \begin{cases} \omega_1, & \text{for } \delta_{\mathbf{x}} < 0 \\ \omega_2, & \text{for } \delta_{\mathbf{x}} > 0. \end{cases} \tag{12}$$

In this study, it is assumed that the distance divergence of traffic flows in any class is independent and identically distributed (i.i.d.). The normal distribution is used for theoretical analysis. The probability density function (PDF) [14] of flow distance divergence is

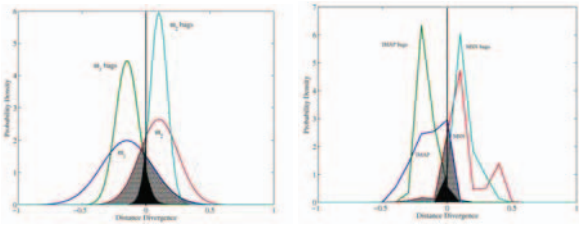
$$p(\delta_{\mathbf{x}}) \sim \begin{cases} \mathcal{N}(\mu_1,\sigma_1^2) & \text{for } \omega_1 \\ \mathcal{N}(\mu_2,\sigma_2^2) & \text{for } \omega_2, \end{cases} \tag{13}$$

where parameters μ_1 and μ_2 denote means, and σ_1^2 and σ_2^2 stand for variances. With the NN classifier, an individual flow $\mathbf{x}$ is classified according to its distance divergence $\delta_{\mathbf{x}}$ using the decision rule in (12). The classification error can occur if

$$\begin{cases} \mathbf{x}\in\omega_1 & \text{with } \delta_{\mathbf{x}} > 0, \\ \text{or} & \\ \mathbf{x}\in\omega_2 & \text{with } \delta_{\mathbf{x}} < 0. \end{cases} \tag{14}$$

Fig. 3a shows the simulation using the normal distributions. It is generally acceptable that the NN classifier is better than random guessing, so we set $\mu_1 < 0$ and $\mu_2 > 0$. In the figure, the classification error of the NN classifier is highlighted by the large lined area.

Let us consider the BoF model-based approach, which is expressed by (9). Given a query BoF, $Q=\{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$, the distance of Q to class ω is calculated by $\frac{1}{n}\sum_{\mathbf{x}\in Q}\min_{\mathbf{x}'\in\omega}\|\mathbf{x}-\mathbf{x}'\|^2$. The distance of Q is used for each of the flows in Q. We define the distance divergence of Q as



(a) simulation (b) real data

Fig. 3. Performance benefit illustration.

$$\begin{aligned} \delta_Q &= \frac{1}{n}\sum_{\mathbf{x}\in Q}\min_{\mathbf{x}'\in\omega_1}\|\mathbf{x}-\mathbf{x}'\|^2 - \frac{1}{n}\sum_{\mathbf{x}\in Q}\min_{\mathbf{x}'\in\omega_2}\|\mathbf{x}-\mathbf{x}'\|^2 \\ &= \frac{1}{n}\sum_{\mathbf{x}\in Q}\left(\min_{\mathbf{x}'\in\omega_1}\|\mathbf{x}-\mathbf{x}'\|^2 - \min_{\mathbf{x}'\in\omega_2}\|\mathbf{x}-\mathbf{x}'\|^2\right) \\ &= \frac{1}{n}\sum_{\mathbf{x}\in Q}\delta_{\mathbf{x}}. \end{aligned} \tag{15}$$

Consequently, the distance divergence of Q is also used for each of the flows in Q. Then, the decision rule of our proposed method becomes

$$\omega^* = \begin{cases} \omega_1, & \text{for } \delta_Q < 0 \\ \omega_2, & \text{for } \delta_Q > 0. \end{cases} \tag{16}$$

It is assumed that the distribution of flow distance divergence is normal, as shown in (13). Then, taking into account (15), the distribution of flow distance divergence produced by the BoF model-based approach is also normal. The PDFs for the BoF model-based flow distance divergence are

$$p(\hat{\delta_Q}) \sim \begin{cases} \mathcal{N}\left(\mu_1,\frac{\sigma_1^2}{n}\right) & \text{for } \omega_1 \text{ bags}, \\ \mathcal{N}\left(\mu_2,\frac{\sigma_2^2}{n}\right) & \text{for } \omega_2 \text{ bags}. \end{cases} \tag{17}$$

Similar to the NN classifier, the classification error of the proposed approach can occur if

$$\begin{cases} Q\in\omega_1 & \text{with } \delta_Q > 0, \\ \text{or} & \\ Q\in\omega_2 & \text{with } \delta_Q < 0. \end{cases} \tag{18}$$

The classification errors are highlighted by the small shaded area in Fig. 3a. We can find that the error area of the proposed method is smaller than that of the NN classifier. In other words, the classification performance can be improved when the correlation among traffic flows is utilized in the classification scheme.

Furthermore, we investigate the classification error based on the normal distribution assumption. According to (13) and (14), the probability of error for the NN classifier is

$$\begin{aligned} P_{NN}(error) &= \int_0^{+\infty} p_{\omega_1}(t)dt + \int_{-\infty}^{0} p_{\omega_2}(t)dt \\ &= 1 - \Phi\left(\frac{-\mu_1}{\sigma_1}\right) + \Phi\left(\frac{-\mu_2}{\sigma_2}\right), \end{aligned} \tag{19}$$

where $\Phi(x)$ is the cumulative distribution function (cdf) of the standard normal distribution, which is defined as

$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-t^2/2}dt. \tag{20}$$

Since $\Phi(x)$ is an increasing function, $\mu_1 < 0$ and $\mu_2 > 0$, we have $P_{NN}(error) \propto \sigma_1$ and $P_{NN}(error) \propto \sigma_2$. The probability of error will decrease if the standard deviations of two classes decrease simultaneously. According to (17), the probability of error for our proposed approach is

$$P_{BoF}(error) = 1 - \Phi\left(\frac{-\sqrt{n}\mu_1}{\sigma_1}\right) + \Phi\left(\frac{-\sqrt{n}\mu_2}{\sigma_2}\right). \quad (21)$$

Specifically, the improvement is

$$\begin{aligned} P_{improve} &= P_{NN}(error) - P_{BoF}(error) \\ &= \Phi\left(\frac{-\sqrt{n}\mu_1}{\sigma_1}\right) - \Phi\left(\frac{-\mu_1}{\sigma_1}\right) + \Phi\left(\frac{-\mu_2}{\sigma_2}\right) - \Phi\left(\frac{-\sqrt{n}\mu_2}{\sigma_2}\right). \end{aligned} \quad (22)$$

Obviously, we have $P_{improve} > 0$ and the improvement is related to n. Our proposed approach can effectively reduce the probability of error since the standard deviations decrease simultaneously from σ_1 to $\frac{\sigma_1}{\sqrt{n}}$ for ω_1 and from σ_2 to $\frac{\sigma_2}{\sqrt{n}}$ for ω_2.

We also evaluate performance benefit of the proposed approach using real network traffic data. Fig. 3b illustrates the performance benefit using the PDFs of distance divergence on a subset of our *isp* traffic data set (see Section 4.1 for detail), which consists of 5k IMAP flows and 5k MSN flows. The distance divergence is calculated based on a small number of training data. BoFs in the data set are constructed by the correlation analysis method presented in Section 3.3. The lined area represents the classification error of the NN classifier. The shaded area represents the classification error of our proposed approach, which is inside the lined area. Since the shade area is smaller than the lined area, the classification accuracy of the proposed method outperforms that of the NN classifier. The results of using real data are consistent with the simulation results, which further confirms the benefit of flow correlation to the classification performance.

When only very few training samples are available, we observe that the standard deviation of flow distance divergence is normally large, which leads to high classification error. In the proposed approach, the mean of BoF-based flow distance divergence is the same to that of the NN classifier. However, the standard deviation of the BoF-based flow distance divergence is much smaller than that of the NN classifier. For instance, in Fig. 3b, the means of flow distance divergence for IMAP and MSN are -0.14 and 0.12, which do not change for our proposed approach. But the standard deviation of flow distance divergence for IMAP reduces from 0.12 for the NN classifier to 0.04 for the proposed approach. The standard deviation for MSN reduces from 0.14 for the NN classifier to 0.08 for the proposed approach. The classification error declines considerably as the standard deviations of flow distance divergence decrease. Therefore, the performance benefit can be obtained because the proposed approach can use correlation information to effectively reduce the standard deviation of flow distance divergence.

3.5 Classification Methods

Let us revisit the BoF model-based classification approach described by (9). It can be interpreted as that a BoF can be classified by aggregating the prediction values of flows produced by the NN classifier. In (9), the prediction value of a flow $\mathbf{x}$ produced by the NN classifier is defined using its minimum distance to the training samples of class ω

$$d_{\mathbf{x}} = \min_{\mathbf{x}' \in \omega} \|\mathbf{x} - \mathbf{x}'\|^2. \quad (23)$$

The distance of a query BoF Q to class ω is obtained by aggregating the flow distances with the "average" operation as follows:

$$d_Q^{avg} = \frac{1}{\|Q\|} \sum_{\mathbf{x} \in Q} d_{\mathbf{x}}. \quad (24)$$

Finally, the flows in Q are classified into the class with the minimum distance of Q. This classification method described by (9) is named "AVG-NN."

From this viewpoint, we can apply different combination operations [40] to aggregate the flow distances, so as to obtain different classification methods for BoFs. With the "minimum" operation, the distance of Q to ω can be calculated by

$$d_Q^{min} = \min_{\mathbf{x} \in Q} d_{\mathbf{x}}. \quad (25)$$

And the modified decision rule is

$$\omega^* = \arg\min_{\omega} (\min_{\mathbf{x} \in Q} \min_{\mathbf{x}' \in \omega} \|\mathbf{x} - \mathbf{x}'\|^2). \quad (26)$$

The classification method described by (26) is named "MIN-NN."

Furthermore, we can aggregate the binary prediction values produced by the NN classifier to conduct classification of BoFs. The decision rule of the NN classifier is

$$\omega_{\mathbf{x}}^* = \arg\min_{\omega} (\min_{\mathbf{x}' \in \omega} \|\mathbf{x} - \mathbf{x}'\|^2). \quad (27)$$

We define the vote of a flow $\mathbf{x}$ for class ω as

$$v_\omega(\mathbf{x}) = \begin{cases} 1, & \text{for } \omega \text{ is } \omega_{\mathbf{x}}^* \\ 0, & \text{else} \end{cases}. \quad (28)$$

With the "majority vote" rule, the modified decision rule becomes

$$\omega^* = \arg\max_{\omega} (\sum_{\mathbf{x} \in Q} v_\omega(\mathbf{x})). \quad (29)$$

The classification method described by (29) is called "MVT-NN."

Table 2 summaries the proposed classification approach. In Step 4, we can apply different aggregation strategies, which have different meanings.

- AVG-NN: combines multiple flow distances to make a decision for a BoF;

TABLE 2
The Proposed Classification Approach

1. Compute statistical features for all traffic flows.
2. Construct traffic BoFs.
Given a query BoF $Q = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$
3. $\forall \mathbf{x}_i \ \forall \omega$ compute prediction values.
4. Aggregate prediction values to predict the class ω^*.
5. Assign all flows in Q to class ω^*.

TABLE 3
Data Sets for Performance Evaluation

Dataset	Flows	Classes	Encrypted class	Dominant class
wide	182k	6	–	WWW
isp	200k	14	SSH/SSL	–

- MIN-NN: chooses a minimum flow distance to make a decision for a BoF;
- MVT-NN: combines multiple decisions on flows to make a final decision for a BoF.

Our study show that they have different impact for traffic classification.

4 Performance Evaluation

In this section, we evaluate the proposed classification methods on two real-world traffic data sets, *wide* and *isp*. First, there are no IP payload available in the *sigcomm*, *lbnl*, and *keio* traces and the accurate ground truth for these traces cannot be built up. Second, we prefer to perform performance evaluation using the large-scale data sets and *wide* and *isp* are the largest ones among the five traces.

4.1 Data Sets

In order to evaluate the work presented in this paper, we have established the ground truth for the part-payload *wide* trace [39] and our full-payload *isp* trace [36]. To do this, we have developed a deep packet inspection (DPI) tool that matches regular expression signatures against flow payload content. A number of application signatures are developed based on previous experience and some well-known tools such as l7-filter (http://l7-filter.sourceforge.net) and Tstat (http://tstat.tlc.polito.it). Also, several encrypted and new applications are investigated by manual inspection of the unidentified traffic. We note that ongoing work is being undertaken to fully identify the rest of the unidentified traffic. The *isp* trace and a full document will be made publicly available to the research community.

In our experiments, we use two data sets for testing classification methods, as summarized in Table 3. One is the *wide* data set, which is obtained from the *wide* trace. The *wide* data set consists of 182k identified traffic flows except encrypted flows. Following the work in [3], all flows are categorized into six classes, P2P, DNS, FTP, WWW, CHAT, and MAIL. The characteristic of the *wide* data set is that it has a small number of classes and the WWW flows dominates the whole data set. The other is the *isp* data set, which is sampled from our *isp* trace. The *isp* data set consists of 200k flows randomly sampled from 14 major classes: BT, DNS, eBuddy, FTP, HTTP, IMAP, MSN, POP3, RSP, SMTP, SSH, SSL, XMPP, and YahooMsg. The *isp* data set has more classes than the *wide* data set and includes encrypted flows, which makes traffic classification more difficult. To sufficiently take into account the difficulty of multiclass classification, we randomly select 30k flows from each of the dominant classes, such as BT and HTTP. Therefore, no class dominates the *isp* data set.

Like most of the previous works [2], [3], we use only TCP flows to perform the experiments in this paper since TCP flows dominate the *wide* and *isp* traces. Our proposed classification approach concentrates on effectively utilize flow correlation information, which is independent of the transport layer protocol and flow statistical features. Considering the applications using UDP are growing, we plan to further evaluate the proposed approach using UDP flows in the future.

TABLE 4
Simple Unidirectional Statistical Features

Type	Features	Count
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

4.2 Experiments

To measure the classification performance, we use two metrics: overall accuracy and F-measure, which are widely used for performance evaluation in the area of traffic classification [3].

- Overall accuracy is the ratio of the sum of all correctly classified flows to the sum of all testing flows. This metric is used to measure the accuracy of a classifier on the whole testing data.
- F-measure is calculated by

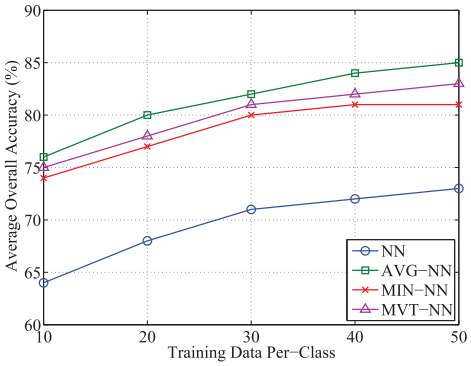
$$F - measure = \frac{2 \times precision \times recall}{precision + recall}, \qquad (30)$$

  where precision is the ratio of correctly classified flows over all predicted flows in a class and recall is the ratio of correctly classified flows over all ground truth flows in a class. F-Measure is used to evaluate the per-class performance.

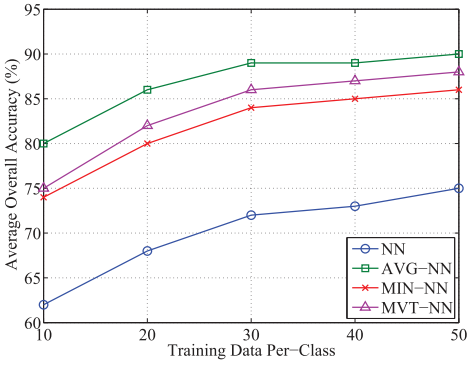
In this work, 20 unidirectional flow statistical features are extracted and used to represent traffic flows, which are listed in Table 4. Feature selection can optimize for high learning accuracy with lower computational complexity. We apply feature selection to remove irrelevant and redundant features from the feature set [41]. We use the correlation-based filter (CFS) [42], [11] and a best first search to generate optimal feature set for each data set. The process of feature selection [31] yields seven and six features for the experiments on the *wide* data set and the *isp* data set, respectively.

We choose the NN classifier representing conventional classification methods for comparison with our proposed approach. First, NN has better performance than Neural Nets and SVM on the *wide* and *isp* data sets, which is shown in Fig. 2. Second, the proposed approach is nonparametric, which shares many advantages with NN. For performance comparison, four classification methods are implemented using the Java language, which are NN, AVG-NN, MIN-NN, and MVT-NN. In these methods, NN does not take into account correlation information in traffic flows. The other three methods proposed in this paper can incorporate correlation information into the classification process.

Taking into account the requirement of very few training samples, the size of training samples for each experiment is



(a) on *isp* dataset



(b) on *wide* dataset

Fig. 4. Average overall accuracy.

less than 50 per class. We repeat each experiment for 100 times with different random seeds for selecting supervised training samples and report the average result on all runs.

4.3 Results

To comprehensively evaluate the proposed approach, the experimental results are reported as overall performance, per-experiment performance and per-class performance.

4.3.1 Overall Performance

The overall performance is evaluated in terms of average overall accuracy against varying training data size from 10 to 50 per class. Fig. 4 shows the overall performance of the four classification methods on the two data sets.

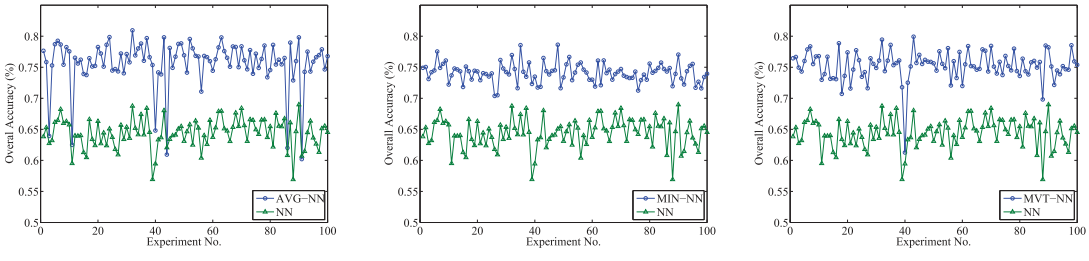
First, a general observation is that all of the proposed methods outperform the NN classifier significantly when only a small number of training data are available. The improvement on overall accuracy is from 10 to 20 percent. For instance, when 10 training samples are available for each class, the average overall accuracy of AVG-NN is higher than that of NN by approximately 15 and 20 percent for the *isp* data set and the *wide* data set, respectively. The reason is that the proposed methods can incorporate correlation information into the class prediction, which is much helpful to improve the classification accuracy.

Second, in terms of overall performance, AVG-NN is the best one among the three proposed methods. The average overall accuracy of AVG-NN is always higher than MIN-NN by about 5 percent on the *wide* data set. On the *isp* data set, the increase is about 3 to 5 percent. MVT-NN is slightly better than MIN-NN on the two data sets. The reason is that the AVG-NN and MVT-NN methods can combine multiple flow prediction values to classify a BoF while the MIN-NN method chooses just one flow prediction value to make a decision.

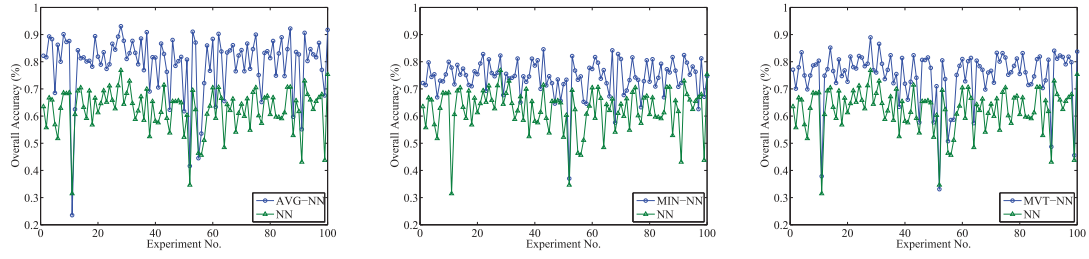
4.3.2 Per-Experiment Performance

Figs. 5 and 6 show the overall accuracy for each experiment using 10 and 20 training samples per class, respectively. Each proposed method is compared with the NN method, so the difference of classification performance in any specific experiment is very clear.

One can see that the classification performance is unstable due to very few training samples, especially in the case of 10 training samples per class. For instance, when 10 training samples are available on the *wide* data set, the performance of NN is between 30 and 80 percent. The gap

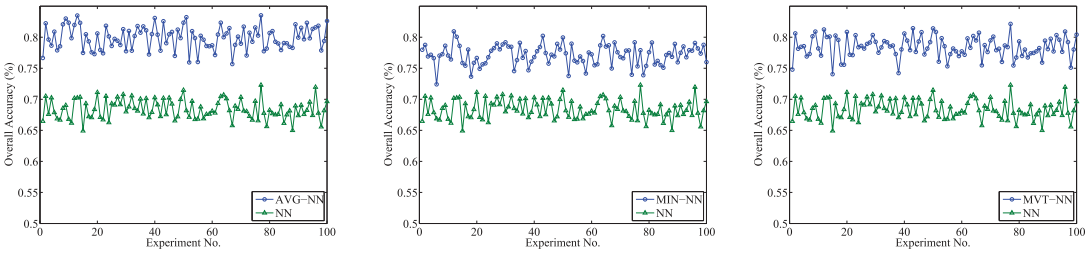


(a) on *isp* dataset

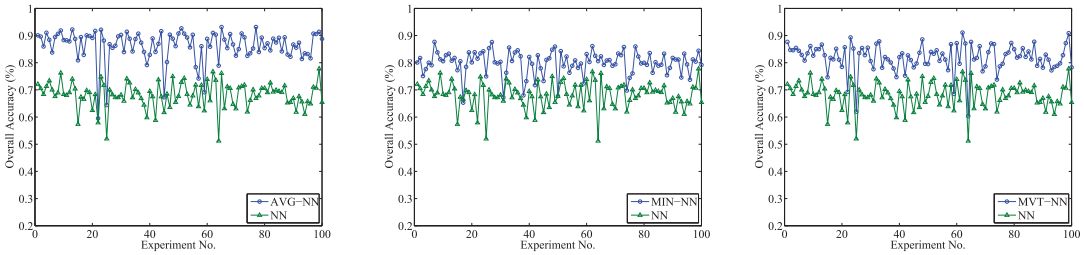


(b) on *wide* dataset

Fig. 5. Overall accuracy per experiment with 10 training samples.



(a) on *isp* dataset



(b) on *wide* dataset

Fig. 6. Overall accuracy per experiment with 20 training samples.

between the best performance and the worst performance can be up to 50 percent. Therefore, the impact of very few training samples to the classification performance is severe.

The figures show that in nearly every experiment each proposed method has much higher accuracy than the NN classifier. The exception, that the overall accuracy of a proposed method is lower than NN, may occur due to the unstable performance, but the number of exceptions is very low. On the *wide* data set, when 10 training samples are available, AVG-NN has two exceptions in 100 experiments. The number of exceptions for MIN-NN and MVT-NN are three and one in 100 experiments, respectively. One exception for MIN-NN occurs for the case of 10 training samples on the *wide* data set. And two exceptions for AVG-NN occur on the *isp* data set when 10 training samples are available for each class. In other cases, the percentage of the proposed methods successfully improving the performance is 100 percent in 100 experiments. The results demonstrate that the proposed methods can improve the classification performance in a robust way. Flow correlation is commonly present in real-world network traffics, which is independent to the supervised training data. The combination of flow correlation and supervised training data can affect the amount of performance improvement.

4.3.3 Per-Class Performance

We use the F-measure metric to measure the per-class performance of the four methods on the two data sets.

Fig. 7 shows the F-measure for each class on the *wide* data set. For the NN classifier, WWW is the easiest to classify. The proposed methods can further improve the performance by about 10 percent in the class. DNS and MAIL are not easy to classify for NN. For these two classes, the proposed methods can significantly improve the classification results. AVG-NN shows the best performance, which can improve the F-measure over NN by about 20 percent for DNS and 15 percent

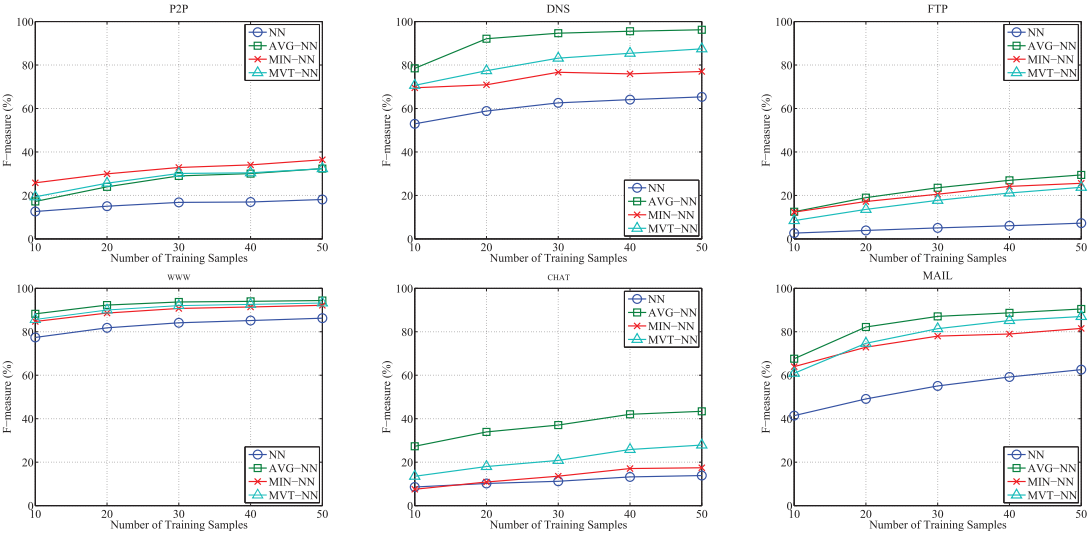


Fig. 7. Average F-measure per class on *wide* data set.

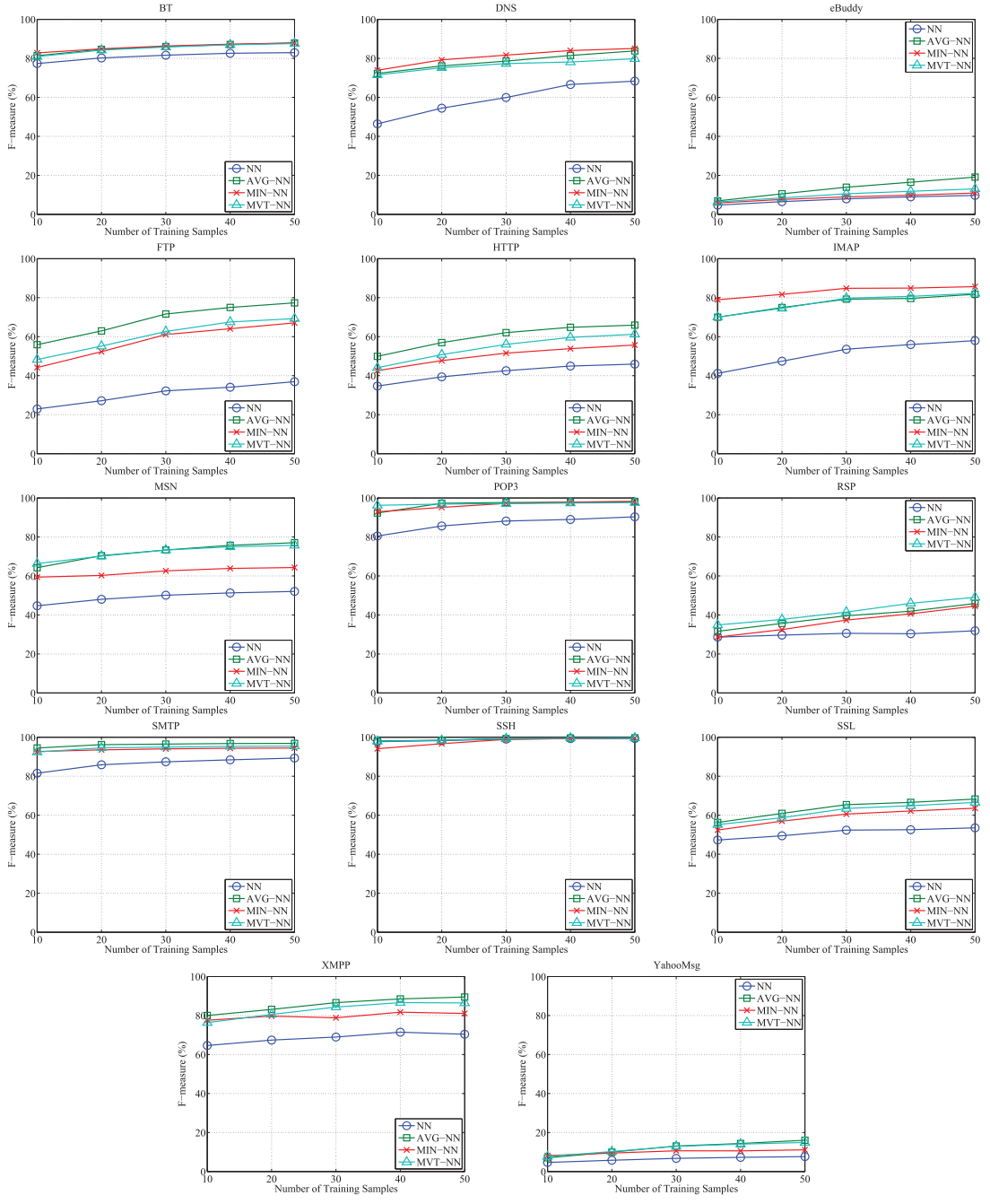
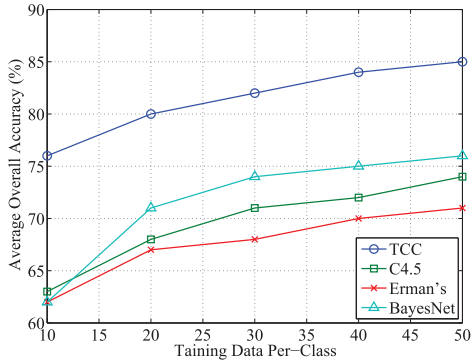


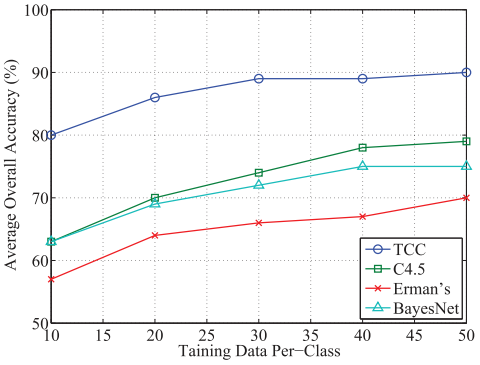
Fig. 8. Average F-measure per class on *isp* data set.

for MAIL. The performance of MVT-NN is slightly better than MIN-NN, both of them outperform NN. In the classes, WWW, DNS, and MAIL, the F-measure of AVG-NN can be over 90 percent. In contrast, it is very hard to classify P2P, FTP, and CHAT, since each class itself may contain multiple applications and communication patterns. Nevertheless, the proposed methods can effectively improve the classification performance for these classes and the improvement is from 10 to 20 percent.

Fig. 8 reports the F-measure for each class on the *isp* data set. It can be seen that the proposed methods can successfully improve the F-measure for each class. The improvement is class based. Similar to the above analysis on the *wide* data set, all applications can be divided into three categories according to the performance of NN, i.e., easy classes, average classes, and hard classes. BT, POP3, SMTP, and SSH are easy classes, in which the F-measure of NN can achieve 80 percent. In the easy classes, although the improvement space is small, the proposed methods, especially AVG-NN, can further improve the performance. For instance, the improvement is 10 percent for POP3. The average classes include DNS, FTP, HTTP, IMAP, MSN, SSL, and XMPP, in which the F-measure of NN is close to 50 percent. The proposed methods, especially AVG-NN, improve the F-measure dramatically for these average classes. For example, the F-measure of AVG-NN is higher than NN by about 40 percent for FTP traffic. The eBuddy, RSP, and YahooMsg traffics are hard for NN to classify. The F-measures of NN for the hard classes are



(a) on *isp* dataset



(b) on *wide* dataset

Fig. 9. Methods comparison.

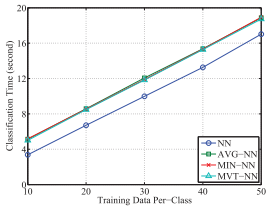
much lower than 50 percent. The proposed methods can improve the F-measure of these hard classes, although the improvement is not very significant.

We observe that the proposed methods do not differentiate much from some of the classes, for example, SSH, eBuddy, and YahooMsg. There may be different reasons for different classes. SSH flows are easy to be classified and there is little space left for improvement. In the two testing data sets, the amount of eBuddy flows and YahooMsg flows are very small, which is insufficient to represent the nature of these classes. The traffic classifier constructed by using the sampled training data has poor generalization for eBuddy and YahooMsg flows. Moreover, based on the ground truth we find that there are not many correlated flows in these classes. The poor training data and the limited correlation information affect the performance improvement of the proposed methods.

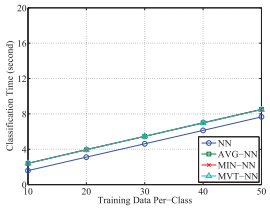
4.3.4 Comparison with Other Existing Methods

A set of experiments are performed to compare the proposed ***TCC*** approach to other recent traffic classification methods including C4.5 [11], BayesNet [11], and Erman's clustering-based method [10]. In these experiments, ***TCC*** adopts the AVG-NN method due to its superior classification performance and Erman's method is implemented without considering unknown classes in the training stage.

Fig. 9 shows the overall performance of four competing methods on the *wide* and *isp* data sets. The results show that ***TCC*** outperforms other three recent traffic classification methods. In the situation of very few supervised training



(a) on *isp* dataset



(b) on *wide* dataset

Fig. 10. Classification time for whole data set.

data, flow correlation can benefit to the traffic classification and ***TCC*** possesses the capability of using flow correlation to effectively improve the traffic classification performance.

4.3.5 Summary

In this paper, we present three methods, AVG-NN, MIN-NN, and MVT-NN, to implement our new approach, ***TCC***. Based on the experimental results, we observe the following.

- With comparison to the NN classifier, the proposed methods can effectively improve the overall performance of traffic classification.
- The proposed methods can improve the classification accuracy in a robust way and consistent improvement is achieved in almost every experiment.
- The proposed methods can improve the F-measure of every class and significant improvements are obtained in most classes.
- AVG-NN shows better performance than MIN-NN and MVT-NN in terms of overall performance, per-experiment performance, and per-class performance.
- ***TCC*** is superior to the existing traffic classification methods since it demonstrates the ability of applying flow correlation to effectively improve traffic classification performance.

5 Discussion

In this section, we provide some discussions on computational performance, system flexibility, and related approaches.

5.1 Computational Performance

The computational performance includes learning time, amount of storage, and classification time. First, the NN classifier does not really involve any learning process, which is shared with our proposed methods. However, other supervised methods, such as neural nets and SVM, need time to learn parameters for their classification model. Second, the proposed methods use the nearest neighbor rule which requires storage for all training data samples. However, the amount of storage is tiny if the training data size is small.

Fig. 10 shows the classification time of the four methods versus training data size. Identifying the nearest neighbor of a given flow from among n training flows is conceptually straightforward with n distance calculations to be performed. The nearest neighbor rule is embedded in the proposed methods for traffic classification. With a small training set, the NN classifiers and the proposed methods classify very quickly. For instance, with 10 training samples

TABLE 5
Related Approaches

Approach	Prior Knowledge	Mapping Problem	Correlation Information	Encrypted Traffic
TCC (proposed in this paper)	Very little	No	Use	Handle, flow statistical feature based
NN based [13], [3] (conventional supervised)	Sufficient	No	No	Handle, flow statistical feature based
k-means based [9], [27] (conventional clustering)	No	Suffer from	No	Handle, flow statistical feature based
Erman *et. al* [10]	Some	Partially address, 'unknown' flows	No	Handle, flow statistical feature based
Ma *et. al* [34]	No	Suffer from	Use	No, packet payload based
Wang *et. al* [36]	No	Suffer from	Use	Handle, flow statistical feature based

for each class, the classification time of the proposed methods are about 2 and 5 seconds for the *wide* data set and *isp* data set, respectively. The proposed methods, AVG-NN, MIN-NN, and MVT-NN, have the same classification time, because they follow the same classification approach.

Due to the extra aggregation operation, the classification time of the proposed methods is a little longer than NN but the classification accuracy of our methods is much higher than NN. If NN achieves the same accuracy to our proposed methods, it needs more training samples and must spend more classification time. For instance, to achieve 75 percent classification accuracy on the *wide* data set, NN needs about 100 training samples per class while AVG-NN needs only 10 training samples per class. The classification time of NN is about 15 seconds, which is much longer than 3 seconds of AVG-NN. On the *isp* data set, NN with 100 training samples per class and AVG-NN with 10 training samples per class can achieve the same classification accuracy, 80 percent. However, the classification time of NN is about 36 seconds, while the classification time of AVG-NN is only 5 seconds. From this perspective, the proposed methods are more effective than NN.

5.2 System Flexibility

The proposed system model is open to feature extraction and correlation analysis. First, any kinds of flow statistical features can be applied in our system model. In this work, we extract unidirectional statistical features from full flows. The statistical features extracted from parts of flows [15] can also be used to represent traffic flows in our system model. Second, any new correlation analysis method can be embedded into our system model. We introduce flow correlation analysis to discover correlation information in traffic flows to improve the robustness of classification. In this paper, a three-tuple heuristic-based method is applied to discover flow correlation which are modeled by BoFs. We presented the comprehensive analysis from theoretical and empirical perspectives, which is based on the BoF model instead of the three-tuple method. Therefore, new correlation analysis methods will not affect the effectiveness of the proposed approach. In the future, we will work on developing new methods for flow correlation analysis.

5.3 Related Approaches

Table 5 compares the related approaches by considering four properties, the amount of prior knowledge, the capability of using correlation information, the need of mapping between clusters and applications, and the capability of handling encrypted traffic. NN-based method and k-means-based method are chosen to represent the supervised and unsupervised traffic classification approaches, respectively.

The proposed approach, TCC, has advantages over other related approaches. First, this paper has shown that the proposed approach using correlation information outperforms NN-based method in terms of traffic classification performance. Second, the approaches using clustering algorithms will suffer from the problem of mapping from a large number of clusters to a small number of applications. This mapping problem is difficult to address without any prior knowledge. Considering supervised training data size, the method proposed by Erman et al. [10] is most related to our approach. However, since the former utilizes supervised training data to label traffic clusters, it will produce a large proportion of "unknown" flows, especially when the supervised training data is very small.

Fig. 11 shows the average unknown flow rate of 1,000 experiments using Erma's method with varying training data size. In the experiments, k-means algorithm ($k = 1{,}000$) is applied to build up traffic clusters and some supervised training samples are used to conduct mapping of clusters and applications. The results show that a large number of flows are labeled as "unknown," which is a critical drawback for the mapping method in practice. For instance, if 10 training samples are available for each class, there are over 90 percent flows labeled as "unknown" on the *isp* and *wide* data set. In other words, the correctly classified flows are less than 10 percent of the whole data set. Therefore, our approach is significantly better than Erma's method.

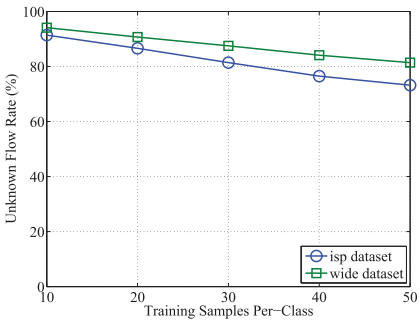


Fig. 11. Unknown flows rate.

6 Conclusion

In this paper, we investigated the problem of traffic classification using very few supervised training samples. A novel nonparametric approach, ***TCC***, was proposed to investigate correlation information in real traffic data and incorporate it into traffic classification. We presented a comprehensive analysis on the system framework and performance benefit from both theoretical and empirical perspectives, which strongly supports the proposed approach. Three new classification methods, AVG-NN, MIN-NN, and MVT-NN, are proposed for illustration, which can incorporate correlation information into the class prediction for improving classification performance. A number of experiments carried out on two real-world traffic data sets show that the performance of traffic classification can be improved significantly and consistently under the critical circumstance of very few supervised training samples. The proposed approach can be used in a wide range of applications, such as automatic recognition of unknown applications from captured network traffic and semi-supervised data mining for processing network packets.

References

[1] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel Traffic Classification in the Dark," *Proc ACM SIGCOMM,* vol. 35, pp. 229-240, Aug. 2005.

[2] T.T. Nguyen and G. Armitage, "A Survey Of Techniques for Internet Traffic Classification Using Machine Learning," *IEEE Comm. Surveys Tutorials,* vol. 10, no. 4, pp. 56-76, Oct.-Dec. 2008.

[3] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices," *Proc. ACM CoNEXT Conf.,* pp. 1-12, 2008.

[4] Y. Wu, G. Min, K. Li, and B. Javadi, "Modelling and Analysis of Communication Networks in Multi-Cluster Systems Under Spatio-Temporal Bursty Traffic," *IEEE Trans. Parallel Distributed Systems,* vol. 23, no. 5, pp. 902-912, May 2012, http://dx.doi.org/10.1109/TPDS.2011.198.

[5] Y.-s. Lim, H.-c. Kim, J. Jeong, C.-k. Kim, T.T. Kwon, and Y. Choi, "Internet Traffic Classification Demystified: on the Sources of the Discriminative Power," *Proc. Sixth Int'l Conf. (Co-NEXT '10),* pp. 9:1-9:12, 2010.

[6] 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 Distributed Systems,* vol. 20, no. 4, pp. 567-580, Apr. 2009.

[7] A.W. Moore and D. Zuev, "Internet Traffic Classification Using Bayesian Analysis Techniques," *ACM SIGMETRICS Performance Evaluation Review (SIGMETRICS),* vol. 33, pp. 50-60, June 2005.

[8] P. Haffner, S. Sen, O. Spatscheck, and D. Wang, "ACAS: Automated Construction of Application Signatures," *Proc. ACM SIGCOMM,* pp. 197-202, 2005.

[9] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic Classification on the Fly," *Proc ACM SIGCOMM,* vol. 36, pp. 23-26, Apr. 2006.

[10] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Offline/Realtime Traffic Classification Using Semi-Supervised Learning," *Performance Evaluation,* vol. 64, nos. 9-12, pp. 1194-1213, Oct. 2007.

[11] N. Williams, S. Zander, and G. Armitage, "A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification," *Proc ACM SIGCOMM,* vol. 36, pp. 5-16, Oct. 2006.

[12] T. Auld, A.W. Moore, and S.F. Gull, "Bayesian Neural Networks for Internet Traffic Classification," *IEEE Trans. Neural Networks,* vol. 18, no. 1, pp. 223-239, Jan. 2007.

[13] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-Service Mapping for QoS: A Statistical Signature-Based Approach to IP Traffic Classification," *Proc. ACM SIGCOMM,* pp. 135-148, 2004.

[14] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification.* Wiley, 2001.

[15] T. Nguyen and G. Armitage, "Training on Multiple Sub-Flows to Optimise the Use of Machine Learning Classifiers in Real-World IP Networks," *Proc. IEEE Ann. Conf. Local Computer Networks,* pp. 369-376, 2006.

[16] J. Erman, A. Mahanti, M. Arlitt, and C. Williamson, "Identifying and Discriminating between Web and Peer-to-Peer Traffic in the Network Core," *Proc. 16th Int'l Conf. World Wide Web,* pp. 883-892, 2007.

[17] L. Bernaille and R. Teixeira, "Early Recognition of Encrypted Applications," *Proc. Eight Int'l Conf. Passive and Active Network Measurement,* pp. 165-175, 2007.

[18] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing Skype Traffic: When Randomness Plays with You," *Proc. Conf. Applications, Technologies, Architectures, and Protocols for Computer Comm.,* pp. 37-48, 2007.

[19] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Traffic Classification through Simple Statistical Fingerprinting," *Proc ACM SIGCOMM,* vol. 37, pp. 5-16, Jan. 2007.

[20] M. Crotti, F. Gringoli, and L. Salgarelli, "Optimizing Statistical Classifiers of Network Traffic," *Proc. Sixth Int'l Wireless Comm. and Mobile Computing Conf.,* pp. 758-763, 2010.

[21] A. Este, F. Gringoli, and L. Salgarelli, "Support Vector Machines for TCP Traffic Classification," *Computer Networks,* vol. 53, no. 14, pp. 2476-2490, Sept. 2009.

[22] S. Valenti, D. Rossi, M. Meo, M. Mellia, and P. Bermolen, "Accurate, Fine-Grained Classification of P2P-TV Applications by Simply Counting Packets," *Proc. Int'l Workshop Traffic Monitoring and Analysis,* pp. 84-92, 2009.

[23] M. Pietrzyk, J.-L. Costeux, G. Urvoy-Keller, and T. En-Najjary, "Challenging Statistical Classification for Operational Usage: the ADSL Case," *Proc. Ninth ACM SIGCOMM,* pp. 122-135, 2009.

[24] A. Finamore, M. Mellia, M. Meo, and D. Rossi, "KISS: Stochastic Packet Inspection Classifier for UDP Traffic," *IEEE/ACM Trans. Networking,* vol. 18, no. 5, pp. 1505-1515, Oct. 2010.

[25] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, "Flow Clustering Using Machine Learning Techniques," *Proc. Passive and Active Measurement Workshop,* pp. 205-214, Apr. 2004.

[26] S. Zander, T. Nguyen, and G. Armitage, "Automated Traffic Classification and Application Identification Using Machine Learning," *Proc. IEEE Ann. Conf. Local Computer Networks,* pp. 250-257, 2005.

[27] J. Erman, M. Arlitt, and A. Mahanti, "Traffic Classification Using Clustering Algorithms," *Proc ACM SIGCOMM,* pp. 281-286, 2006.

[28] J. Erman, A. Mahanti, and M. Arlitt, "Internet Traffic Identification Using Machine Learning," *Proc. IEEE Global Telecomm. Conf.,* pp. 1-6, 2006.

[29] Y. Wang, Y. Xiang, and S.-Z. Yu, "An Automatic Application Signature Construction System for Unknown Traffic," *Concurrency and Computation: Practice and Experience,* vol. 22, pp. 1927-1944, 2010.

[30] A. Finamore, M. Mellia, and M. Meo, "Mining Unclassified Traffic Using Automatic Clustering Techniques," *Proc. Third Int'l Traffic Monitoring and Analysis (TMA),* pp. 150-163, Apr. 2011.

[31] Weka 3: Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka/, 2012.

[32] J. Zhang and L. Ye, "Image Retrieval Based on Bag of Images," *Proc. IEEE Int'l Conf. Image Processing,* pp. 1865-1868, Nov. 2009.

[33] O. Boiman, E. Shechtman, and M. Irani, "In Defense of Nearest-Neighbor Based Image Classification," *IEEE Int'l Conf. Computer Vision and Pattern Recognition,* pp. 1-8, June 2008.

[34] J. Ma, K. Levchenko, C. Kreibich, S. Savage, and G.M. Voelker, "Unexpected Means of Protocol Inference," *Proc. Sixth ACM SIGCOMM,* pp. 313-326, 2006.

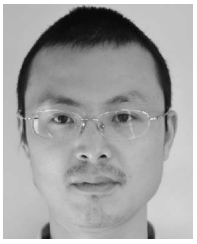
[35] M. Canini, W. Li, M. Zadnik, and A.W. Moore, "Experience with High-Speed Automated Application-Identification for Network-Management," *Proc. Fifth ACM/IEEE Symp. Architectures for Networking and Comm. Systems,* pp. 209-218, 2009.

[36] Y. Wang, Y. Xiang, J. Zhang, and S.-Z. Yu, "A Novel Semi-Supervised Approach for Network Traffic Clustering," *Proc. Int'l Conf. Network and System Security,* Sept. 2011.

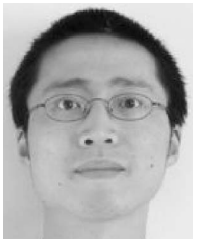
[37] Network Traffic Tracing at SIGCOMM 2008, http://www.cs.umd.edu/projects/wifidelity/tracing, 2012.

[38] R. Pang, M. Allman, M. Bennett, J. Lee, V. Paxson, and B. Tierney, "A First Look at Modern Enterprise Traffic," *Proc. ACM SIGCOMM,* pp. 15-28, 2005.

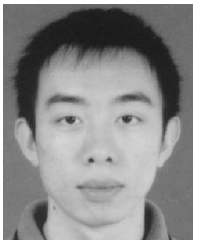
[39] MAWI Working Group Traffic Archive, http://mawi.wide.ad.jp/mawi/, 2012.
[40] A. Webb, *Statistical Pattern Recognition.* John Wiley & Sons, 2002.
[41] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *J. Machine Learning Research,* vol. 3, pp. 1157-1182, Mar. 2003.
[42] M.A. Hall, "Correlation-Based Feature Selection for Machine Learning," PhD Thesis, Department of Computer Science, The Univ. of Waikato, Hamilton, New Zealand, Apr. 1999.



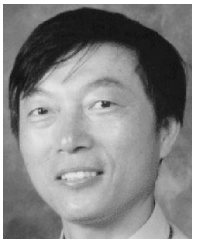
Jun Zhang received the PhD degree from the University of Wollongong, Australia, in 2011. Currently, he is with the School of Information Technology at Deakin University, Melbourne, Australia. His research interests include network and system security, pattern recognition, and multimedia processing. He has published more than 20 research papers in the international journals and conferences, such as *IEEE Transactions on Image Processing*, *The Computer Journal*, and IEEE International Conference on Image Processing. He received 2009 Chinese Government Award for outstanding self-financed student abroad. He is a member of the IEEE.



Yang Xiang received the PhD degree in computer science from Deakin University, Australia. Currently, he is with the School of Information Technology, Deakin University. In particular, he is currently leading in a research group developing active defense systems against large-scale distributed network attacks. He is the chief investigator of several projects in network and system security, funded by the Australian Research Council (ARC). His research interests include network and system security, distributed systems, and networking. He has published more than 100 research papers in many international journals and conferences, such as *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Information Security and Forensics*, and *IEEE Journal on Selected Areas in Communications*. He has served as the program/general chair for many international conferences such as ICA3PP 12/11, IEEE/IFIP EUC 11, IEEE TrustCom 11, IEEE HPCC 10/09, IEEE ICPADS 08, NSS 11/10/09/08/07. He has been the PC member for more than 50 international conferences in distributed systems, networking, and security. He serves as the associate editor of *IEEE Transactions on Parallel and Distributed Systems*and the editor of *Journal of Network and Computer Applications*. He is a member of the IEEE.

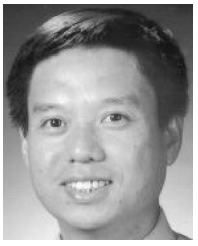


Yu Wang received the BSc degree in electronic information science and technology from Sun Yat-Sen University. Currently, he is working toward the PhD degree with the School of Information Technology at Deakin University. His research interests include traffic classification and network security.



Wanlei Zhou received the PhD degree from the Australian National University, Canberra, Australia, and the DSc degree from Deakin University, Victoria, Australia, in 1991 and 2002, respectively. Currently, He is the chair professor of Information Technology and the head of the School of Information Technology, Deakin University, Melbourne. His research interests include distributed and parallel systems, network security, mobile computing, bioinformatics, and e-learning. He has published more than 200 papers in refereed international journals and refereed international conference proceedings. Since 1997, he has been involved in more than 50 international conferences as the general chair, a steering chair, a PC chair, a session chair, a publication chair, and a PC member. He is a senior member of the IEEE.

Yong Xiang received the BE and ME degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 1983 and 1989, respectively. and the PhD degree from The University of Melbourne, Melbourne, Australia, in 2003. He was with the Southwest Institute of Electronic Equipment of China, Chengdu, from 1983 to 1986. In 1989, he joined the University of Electronic Science and Technology of China, where he was a lecturer from 1989 to 1992 and an associate professor from 1992 to 1997. He was a senior communications engineer with Bandspeed, Inc., Melbourne, Australia, from 2000 to 2002. Currently, he is an associate professor with the School of Information Technology at Deakin University, Melbourne, Australia. His research interests include blind signal/system estimation, information and network security, communication signal processing, multimedia processing, pattern recognition, and biomedical signal processing. He is a senior member of the IEEE.



Yong Guan received the MS and BS degrees in computer science from Peking University, in 1996 and 1990, respectively and the PhD degree in computer science from Texas A&M University, in 2002. He is an associate professor in the Department of Electrical and Computer Engineering at Iowa State University and is the associate director for Research for the Iowa State University's NSA-designated Information Assurance Center. Meanwhile, he is an Ames Lab associate for the Midwest Forensics Resource Center at the US DoE's Ames Lab. Between 1990 and 1997, he worked as an assistant engineer (1990-1993) and lecturer (1996-1997) in Networking Research Group of Computer Center at Peking University, China. In 2002, he joined Iowa State University as a faculty member. His research interests include security and privacy issues, including computer and network forensics, wireless security, and privacy-enhancing technologies for the Internet. He served as the general chair for 2008 IEEE Symposium on Security and Privacy (Oakland 2008), co-organizer for ARO Workshop on Digital Forensics, coordinator of Digital Forensics Working Group at NSA/DHS CAE Principals Meetings, the program co-vice-chairs for ICDCS 2008 (Security and Privacy Area) and the co-organizer for ARO Workshop on Digital Forensics (2009), and on the Program Committees of a number of conferences and workshops (including ACM CCS, INFOCOM, ICDCS, IFIP-SEC, SecureComm, SADFE, DFRWS, ISC, etc.). He received the Best Paper Award from the IEEE National Aerospace and Electronics Conference in 1998, won the second place in graduate category of the International ACM student research contest in 2002, National Science Foundation (NSF) Career Award in 2007, Iowa State University Award for Early Achievement in Research in 2007, the Litton Industries Professorship in 2007, and the Outstanding Community Service Award of IEEE Technical Committee on Security and Privacy in 2008. He is a member of the IEEE.