Inferring Users' Online Activities Through Traffic Analysis

Fan Zhang[1,3], Wenbo He[1], Xue Liu[2] and Patrick G. Bridges[4]
Department of Electrical Engineering, University of Nebraska-Lincoln, NE, USA[1]
School of Computer Science, McGill University, Montreal, Quebec, Canada[2]
Department of Electronics and Information, Huazhong University of Sci. & Tech., Wuhan, China[3]
Department of Computer Science, University of New Mexico, Albuquerque, NM, USA[4]
fzhang2@unl.edu, wenbohe@engr.unl.edu, xueliu@cs.mcgill.ca, bridges@cs.unm.edu

ABSTRACT

Traffic analysis may threaten user privacy, even if the traffic is encrypted. In this paper, we use IEEE 802.11 wireless local area networks (WLANs) as an example to show that inferring users' online activities accurately by traffic analysis without the administrator's privilege is possible during very short periods (e.g., a few seconds). The online activities we investigated include web browsing, chatting, online gaming, downloading, uploading and video watching, etc. We implement a hierarchical classification system based on machine learning algorithms to discover what a user is doing on his/her computer. Furthermore, we conduct experiments in different network environments (e.g., at home, on university campus, and in public areas) with different application scenarios to evaluate the performance of the classification system. Results show that our system can distinguish different online applications on the accuracy of about 80% in 5 seconds and over 90% accuracy if the eavesdropping lasts for 1 minute.

Categories and Subject Descriptors

C.2.0 [**Information Systems Applications**]: General—*Security and Protection*

General Terms

Experimentation, Security

Keywords

Traffic Analysis, Privacy, Users' Online Activities, Machine Learning

1. INTRODUCTION

Traffic analysis attacks on encrypted traffic are often referred to as *side-channel information leaks*. Although the privacy threat of side-channel information leaks has been discovered in various applications, including web browsing [1, 2], secure shell (SSH) [3], keystroke dynamics [4], video-streaming [5] and voice-over-IP (VoIP) [6, 7], these investigations are based mostly on an implicit assumption that the adversary knows a user's online activity (i.e., the particular network application or service that a user is running). Actually, a user's online activity is highly private and sensitive information. Users usually do not want strangers, their parents, guardians, supervisors, bosses or peers to track their online activities. Furthermore, it is more risky if the technique inferring users' online activities is combined with the previous study on side-channel information leaks.

Nowadays, due to the shared-medium of wireless links and the ease of eavesdropping in WLANs, traffic traces that users sent over wireless links are almost exposed to adversaries. In this paper, we investigate the user privacy breach on *users' online activities* by analyzing *encrypted MAC-layer* traffic. We attempt to infer users' online activities in real time by using no more information than packet size, timing and direction. It is a challenging task to do this accurately with such limited information, especially among a wide range of network applications, such as web browsing, online chatting, online gaming, downloading, uploading, online video and BitTorrent. Although traffic features (e.g., average packet size, frequency of a frame and average interval-arrival time) between low bandwidth consumption and high bandwidth consumption applications are identifiable (e.g., chatting vs. downloading), similar applications have very fine distinction, especially under time-varying network environments, different users' online habits and software. In our work, we show that traffic, even from the same application, varies largely among different environments. We also consider *concurrent* online activities. In this case, traffic features in one application may be submerged by another application; and the changeable features make the accurate identification of users' online activities even more difficult.

To overcome the above challenges, we explore an *online hierarchical classification system* based on machine learning (ML) techniques to map traffic features to the online activities and show that an adversary is able to infer and track what the user is doing during very short periods (e.g. a few seconds) without any information about the protocols, software and servers the user is using. Specifically, our classification system performs *multiclass classification* by taking advantage of both the efficient computation of decision tree structure and the high classification accuracy of Support Vector Machine (SVM) and Neural Network (NN) algorithms. Traffic features adopted in the classification system are only based on packet-level statistical values, such as

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
WiSec'11, June14–17, 2011, Hamburg, Germany.
Copyright 2011 ACM 978-1-4503-0692-8/11/06 ...$10.00.

average packet size and average interarrival time etc., in the MAC layer. We conduct experiments in different network environments (e.g., at home, on university campus and in public areas) with different application scenarios to evaluate the classification system. Results show that the proposed classification system achieves good accuracy in noisy environments, distinguishes online activities with around 80% accuracy in 5 seconds, and with over 90% accuracy if the eavesdropping duration lasts for 1 minute. We hope that our work will alert LAN users, network designers, and administrators that there is a serious privacy breach of users' online activities.

The rest of this paper is organized as follows. We summarize the related work in Section 2. Section 3 overviews the background and challenges. We present the design of the online hierarchical classification system in Section 4. Then we demonstrate the experiments conducted at home, on campus and in public networks to evaluate the accuracy of our classification system in Section 5. Section 6 discusses the implication issues. Finally, we conclude the paper in Section 7.

2. RELATED WORK

Side-channel Information Leaks: Side-channel information leaks have been researched widely. Encrypted traffic does not prevent traffic analysis attacks, thus user privacy is still vulnerable. Liberatore, et al. [8] present a straightforward traffic analysis attack against encrypted HTTP streams to identify the source of the traffic, and the authors in [2, 9] do similar webpage fingerprinting. Chen, et al. [1] find that significant traffic distinctions of different webpages help an adversary to wiretap what the user is browsing. Moreover, the lengths of encrypted VoIP packets can be used to identify the phrases spoken within a call [7]. In addition, adversaries may adopt wireless signal strength in multiple monitoring locations to obtain an accurate estimation of a user's location and motion behind walls [10, 11]. Srinivasan, et al. [12] show that a Fingerprint And Timing-based Snooping (FATS) attack can observe private activities, such as cooking, showering, and using the toilet, by eavesdropping on the wireless transmissions of sensors in a home. However, the above research rarely concerns the privacy regarding users' online activities.

Traffic Classification: Traffic classification is mostly employed by network administrators to monitor network traffic and identify Internet applications. These applications are mostly described based on protocol behaviors, such as HTTP, SMTP, FTP, SSH and DNS, etc. But nowadays, many applications are able to run over one protocol. For example, web browsing, chatting, online gaming, downloading, watching online video, etc., can be executed in HTTP protocol. Hence, we focus on the users' online activities which may have more sensitive information than protocol behaviors.

In addition, traffic classification usually uses traffic features in or beyond the IP layer, such as IP address, TCP port, protocol fingerprinting, etc. Few are implemented only by features in the MAC layer. But compared with the difficulty of getting traffic from the routers, gateways or servers without the administrator's privilege, the easy way is for an adversary to eavesdrop on the traffic in the MAC layer. Thus in this paper, we investigate traffic classification on encrypted traffic in the MAC layer.

The traditional identification techniques, the *port-based approach*, *payload-based approach* and *host-behavior-based approach* [13] are no longer valid in the MAC layer. The *port-based approach* relies on the well-known ports registered by the Internet Assigned Numbers Authority (IANA) [14], and the *payload-based approach* is based on features of the payload [15]. But this information is undetectable in the MAC-layer due to the MAC-layer encryption. Similarly, the *host-behavior-based approach* [16, 17] can not be used without end-to-end information about host connections, such as port, IP address, etc. Instead, we employ a *flow-feature-based approach* which is based on *machine learning (ML)* techniques for the MAC-layer traffic classification.

Recently, Wright, et al. [18, 19] and Dainotti, et al. [20] propose packet-level classification approaches based on the features, packet interarrival time and payload size. Our classification approach is different from the above approaches in a few important ways. (1) The evaluation in [18, 19, 20] is based on traffic flows, which means that packets in a flow belong to an application. In contrast, we do not know which flow or application an encrypted frame belongs to in the MAC layer. (2) In terms of different time-varying wireless environments, the traffic varies largely. Thus, we evaluate our classification system at home, on campus and in a public area. (3) Our classification system, which is based on different machine learning algorithms and features, gives an identification in real time (every 5 seconds). Nowadays, online classification methods [21, 22, 23, 24] rely on features of TCP/UDP and IP traffic. These features are unavailable in the MAC layer and can not be applied to the MAC-layer classification.

Machine Learning: Recently, researchers have adopted ML technologies in the flow-feature-based traffic classification. Nguyen et al. conduct a survey [13] in this area focusing on ML techniques, such as Bayesian techniques [25], k-nearest neighbor algorithm, decision tree (e.g., C4.5), NN and SVM [26]. Hidden Markov Model (HMM) is also employed in traffic analysis [7, 18, 20]. In this paper, we use two intelligent ML algorithms, SVM and NN, to identify seven popular online activities.

3. BACKGROUND AND CHALLENGES

3.1 Adversary Model

The shared-medium nature of WLANs poses privacy vulnerabilities on users' online activities. To track the traffic from and to a user, the adversary only needs to install sniffer software (e.g. Wireshark, Aircrack-ng). In this case, the network adapter passes all packets it receives to the adversary rather than just frames addressed to it. In this paper, we act as an adversary and use the *Intel Wireless WiFi Link 4965AGN* network cards with the *Libpcap* library to intercept specific users' traffic. The adversary does not know any information about the software and encryption schemes adopted by the users.

Figure 1 shows the working scenario of an adversary who adopts traffic classification to infer users' online activities. The adversary sniffs the WLAN in the same channel as the Access Point (AP). The classification system collects traffic samples of the whole network and knows how many users are in this WLAN. Then after the adversary inputs the MAC address of the user he wants to eavesdrop on, the classifica-

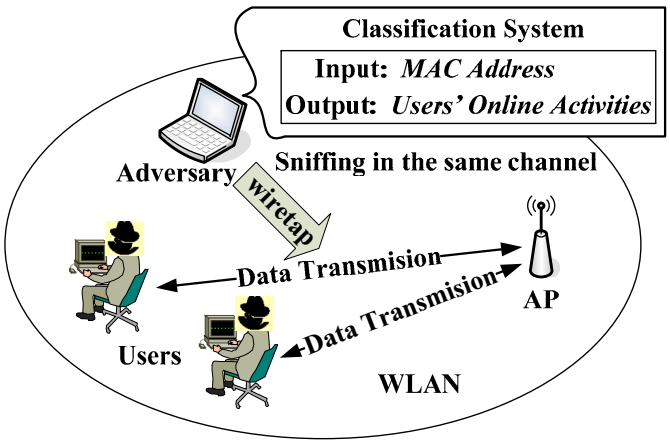


Figure 1: Adversary Model

tion system identifies which online applications the user is running.

3.2 Challenges

Challenges for the MAC-layer traffic classification come from the following factors.

3.2.1 Limited Flow Features

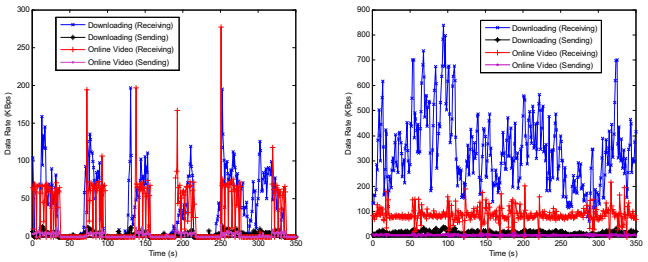
Features for traffic classification in TCP/UDP and the IP layer can be obtained from packet headers, including the TCP port and IP address, payload information, SYN packet and protocol fingerprinting [13]. In contrast, valid information from the MAC layer is very limited, since MAC layer frames are usually encrypted. From the MAC header, we can find the MAC address, SSID (Service Set Identifier), directions of the traffic (receiving or sending), RF signal strength and frame types. Besides the header, we only get the packet-level data, such as frame size and its timestamp.

3.2.2 Noises in Traffic Features

Traffic patterns, even from the same application, can be easily affected by network situations, such as signal strength, available bandwidth, and service provider. Therefore, a MAC-layer flow of a given application may exhibit different features in different time slots, locations and network situations. From Figure 2, we can see that the data rate of the same applications (e.g., downloading or online video) fluctuates tremendously (even from 0 to 1MBps) in a very short time and differs markedly in different network situations. The data rate in Figure 2(b) with better network situations is much larger than that in Figure 2(a). In addition, the flow features may be affected by the attributes of an application, such as "who is running the application"; "which software is used by a user"; "the target content server (websites)", etc. We download the same contents by BitTorrent (BT) and HTTP from different servers at the same time. As shown in Figure 3, the data rates widely diverge. This noise makes a highly accurate traffic classification difficult to achieve. Hence, it is hard to find standard or uniform parameters for classifiers in different situations.

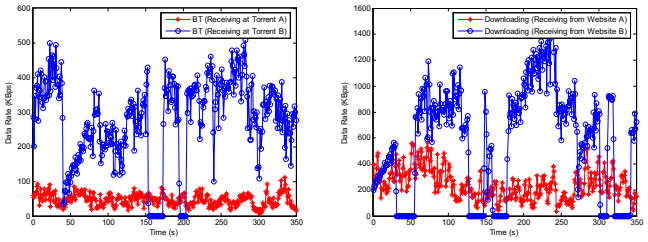
3.2.3 Existence of Concurrent Applications

A user may open multiple application windows and perform multiple tasks on the Internet simultaneously. By wiretapping on the traffic in the MAC layer, we only observe the aggregated traffic to and from a given user. Hence, using traffic analysis, it is very hard to know how many applications are running by a specific user and what bandwidth portion is allocated to individual applications. In addition, because the frames in the MAC layer are transmitted over the last hop, the end-to-end information, such as the relationship of frames between sending and corresponding responses is undetectable.



(a) Poor Network Situation (b) Good Network Situation

Figure 2: Data rate of the same applications in different network situations



(a) BT Application (b) Download Application

Figure 3: Data rate of the same applications from different servers

3.2.4 Dynamic Task Switch by Users

Because a user may continually switch his/her applications and each application may last for a short period of time, tracking users' activities requires that the system should map the traffic patterns to specific applications quickly and accurately without complete knowledge about the traffic. In addition, the classification model must be updated dynamically according to the time-varying network environments.

4. CLASSIFICATION METHODOLOGY

4.1 Data Collections

4.1.1 Data Set

We investigate seven popular network applications, including web browsing, chatting, online gaming, downloading, uploading, online video and BT, which are labeled from ① to ⑦. We list the applications and their variations in Table 1.

Multiple concurrent applications are also studied in this paper. In most of the cases, users do not run more than two applications in each short time (e.g., 5 seconds). Even if users run more than two, we can identify two main applications which have larger traffic loads in the concurrent traffic.

Table 1: Attributes of network applications

Applications	Software	Server (Website)
Browsing ①	IE, Mozilla Firefox	Yahoo, CNN, Amazon, Google, etc.
Chatting ②	MSN, QQ, Google Talk	MSN, Tencent, Google
Online Game ③	QQ Three Kingdoms, World of Warcraft	Tencent, WoW Servers
Downloading ④	HTTP	Ubuntu, Microsoft, Sun, etc.
Uploading ⑤	HTTP	YouTube, Facebook, Google, etc.
Online Video ⑥	HTTP or specific client	YouTube, MSN, etc.
BitTorrent ⑦ (Typical P2P File-sharing)	Bitcomet, Flashget, Xunlei, Linux BitTorrent Client	http://torrent.ubuntu.com, Xunlei, http://www.verycd.com/, etc.

Table 2: Features used in classification

data frames, control frames, management frames	data rate (receiving and sending)
	frame size (categories, mean, median, variance, ...)
	frame interarrival time (mean, median, variance, ...)
	frame size distribution ($> m$ bytes or $< m$ bytes) (m = 100, 500, 1000, ...)
	number of frames (receiving and sending)

Hence, we consider only two concurrent applications in this paper. They can be divided into three types: a large bandwidth consumption application plus a low bandwidth consumption application (e.g., downloading and chatting), two large bandwidth consumption applications (e.g., downloading and online video) and two small bandwidth consumption applications (e.g., browsing and chatting). The *six typical* combinations of the concurrent applications we selected in our experiments are browsing and chatting {①,②}, browsing and BT {①,⑦}, browsing and uploading {①,⑤}, chatting and downloading {②,④}, downloading and video {④,⑥}, video and BT {⑥,⑦}.

The traffic samples we collected in the experiments are divided into the training set and the testing set. The former is used to train the classification system to build the model, and the latter is used to evaluate classification accuracy.

4.1.2 Scenarios

We consider three dominant WLAN deployments: public network, home network, and university campus (or enterprise) network. The traffic in public networks is usually not encrypted at the link layer [27]. Hence, it is very easy to obtain the identifying features of a specific user's traffic in such an environment. We carry out tests in airports, cafe houses, McDonald's, etc. Home and small business networks are small and more likely to adopt link-layer encryption, such as Wired Equivalent Privacy (WEP) or WiFi Protected Access (WPA). We conduct the experiments in a home environment with *Comcast Internet* as the Internet Service Provider. Campus networks usually support a large population of users and employ the link-layer encryption. We also conduct experiments on the university campus.

To achieve better accuracy, the classification system must adapt to dynamic network conditions and tolerate the noises caused by variant versions of applications, users' habits and interferences among concurrent applications.

To evaluate the network condition, we collect traffic data samples in different *received signal strength indication (RSSI)*, with various *channel utilization*, in different time slots (morning, afternoon and evening, respectively) and networks. In this way, the training data in a certain network condition indicated by RSSI and channel utilization are used to build the classifier models under different network conditions. The system will select the most appropriate classifier model, which is under similar network conditions as the testing data.

Table 3: Features of seven applications (from AP to the user)

Applications	Average frame size (byte)	Average interarrival time (s)	Frame size distribution (>500bytes)
Browsing ①	1013.20	0.0284	64.617%
Chatting ②	269.06	0.9901	9.357%
Online Game ③	459.53	0.3084	34.501%
Downloading ④	1575.3	0.0028	99.951%
Uploading ⑤	132.76	0.0301	0.0307%
Online Video ⑥	1547.6	0.0119	99.560%
BitTorrent ⑦	962.04	0.0247	60.650%

To tolerate noise, we consider various attributes of applications in our data collection efforts. We select commonly used applications and their attributes listed in Table 1. More than 10 people participate in the test with their choices of different operating systems, software and laptops.

4.2 Feature Extraction and Selection

4.2.1 Feature Extraction

The observed traffic traces are time-series data. Individual frames contain very little information, but they are correlated with their neighboring frames in a certain pattern for a given application. Therefore, the statistical features of frames may disclose information leaks. For example, chatting and gaming have a small number of frames with relatively small size for both sending and receiving. Browsing contains bursty traffic. Downloading and uploading are both high bandwidth consumption applications with large frame size in downlink and uplink, respectively. Online video demonstrates a relatively stable data rate which is usually between BT and downloading. BT may be a high bandwidth consumption application in bidirectional directions. Its traffic variance is also very large.

For statistical analysis, we need to demarcate traffic flows into a series of measurements. We use an "*observation window*" to represent a segment of a flow and extract the flow features in each individual window. The window size, W, can be either expressed in time domain or measured by events. Intuitively, if the window size reflects the periodic

Table 4: Similarity distances of seven applications

D_M	①br.	②ch.	③ga.	④do.	⑤up.	⑥vo.	⑦bt.
①br.	-	1854.3	361.14	374.48	767.63	21.707	**3.7258**
②ch.	1854.3	-	**8.9232**	4258.1	1556.3	1625.1	1439.3
③ga.	361.14	**8.9232**	-	55341	33480	2176.9	205.03
④do.	374.48	4258.1	55341	-	100037	**93.777**	576.74
⑤up.	767.63	1556.3	33480	100037	-	5699.2	**189.79**
⑥vo.	21.707	1625.1	2176.9	93.777	5699.2	-	**15.852**
⑦bt.	**3.7258**	1439.3	205.03	576.74	189.79	15.852	-

components of the traffic trace, it will be useful for feature extraction. We have attempted to find the periodicity of the traffic traces by the *fast Fourier transform*. Unfortunately, the periodicity is fuzzy and undetectable. In our experiments, we use a fixed time domain to describe W. For example, the traffic is a time series denoted as $\{T_1, T_2, \cdots, T_W, T_{W+1}, \cdots, T_N\}$. Without using a sliding window, the traffic data will be divided into *flow segments*: $\{T_1, T_2, \cdots, T_W\}$,$\{T_{W+1}, \cdots, T_{2W}\}$, $\cdots$. But with the sliding window technique, the flow segments to be considered will be $\{T_1, T_2, \cdots, T_W\}$, $\{T_2, \cdots, T_{W+1}\}$, $\cdots$. In this way, we can obtain more instances of features.

We list the features used in our classification in Table 2. Therein, "frame size distribution (>500bytes)" means the percentage of frames which are larger than 500 bytes in each window size. Table 3 illustrates features in home scenarios with *RSSI* around 55, which is equivalent to -50dBm of RF Signal Strength according to Cisco Standard [28]. We can see that different applications have very different features.

4.2.2 Feature Selection

Since the flow features are not equally important for inferring specific applications, we need to identify representative features and remove irrelevant and redundant ones to improve classification accuracy. We use a *best first search* to generate candidate sets of features, since it provides higher classification accuracy than *greedy search* [29]. We also use the correlation-based filter (CFS) to examine the relevance of each feature, i.e., those highly correlated to a specific class but with minimal correlation to each other [26]. CFS is practical and outperforms the other filter method (consistency-based filter) in terms of classification accuracy and efficiency [29]. For every trace, the CFS selected three categories of features: frame size, number of frames and frame distribution information. The aptness of the feature selection is evaluated in Section 5.

4.3 ML Algorithms

In our classification system, the relationships of frames in the MAC layer are vaguely understood and difficult to describe adequately with conventional approaches, so straightforward classification methods, such as k-nearest neighbor algorithm, decision tree (e.g., C4.5) and Naive Bayes, etc., may not yield high classification accuracy. HMM is a powerful statistical technique based on the Markovian assumption. But the number of parameters that need to be set in an HMM is huge. As a result, the amount of data that is required to train an HMM is also very large. Hence, we use SVM and NN algorithms, which are widely used in intelligent data mining applications [26], to design the classifiers. These two methods can model complex relationships between inputs and outputs and find patterns in data. NN is a universal approximation tool and able to tolerate noise. SVM performs better with small training sets, which is very useful for online classification. Kim et al. present in [26] that SVM outperforms other classification methods with more than 98% overall accuracy. Considering that the relationships between features seem to be nonlinear, we choose radial basis function (RBF) for both SVM and NN algorithms to achieve non-linear classification. RBF is one of the most commonly used in SVM. Similarly, we also use a popular artificial NN, radial basis function network (RBFN), to do the classification.

4.4 Hierarchical Classification Structures

Classifying seven applications in aggregated traffic belongs to the category of multiclass classification. The main technique for multiclass classification is to decompose the multiclass problem into several binary problems, especially for SVM. The common methods to build such binary classifiers are (i) one of the classes to the rest (one-versus-all, OVA) (ii) between every pair of classes (one-versus-one, OVO), or (iii) directed acyclic graph SVM (DAGSVM) [30]. For a K-class problem ($K = 7$ in our classification for seven popular online applications), the disadvantage of OVA is that it needs K classifiers and each classifier needs to be trained by training samples of all the classes. OVO and DAGSVM have to construct $K(K-1)/2$ classifiers, which incur large computational overhead.

To decrease the number of classifiers and improve the accuracy, we use the decision tree structure [31, 32, 33] in our classification system. We present hierarchical classification structures, which take advantage of both the efficient computation of the tree structure and the high classification accuracy of SVM and RBFN, in Figure 4 and 5, respectively. We use K ($K = 7$) classifiers for SVM and $K - 1$ classifiers in the RBFN model. Each classifier only needs to be trained to a subset of the training samples, and the hierarchical classification allows individual classifiers in the structure to be updated flexibly and independently. This is a notable improvement when the number of classes is large. At each node of the tree, a decision is made to assign the input to several possible groups which are the subtrees. Each of these groups may contain multiple classes. This is repeated down the tree until the sample reaches a leaf node that represents the class it has been assigned to.

In the design of the classification system, the basic rule we obey is to separate the most different and independent groups or classes first and distinguish the most similar classes last. We measure the similarity of different applications by using the *Mahalanobis distance*. It is a multivariate distance measure for several modeling algorithms, such as k-nearest

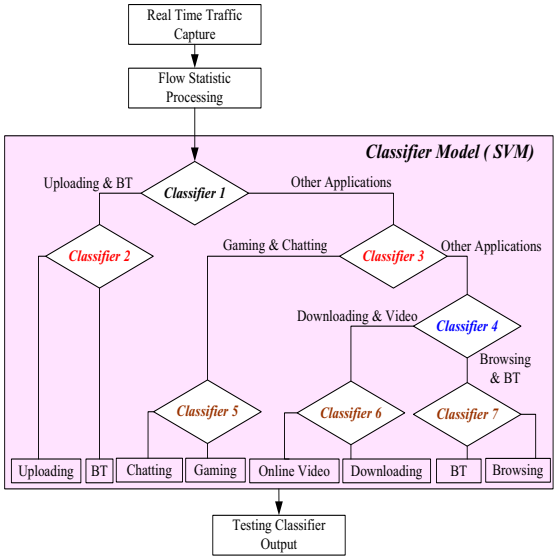


Figure 4: The hierarchical classification structure of SVM algorithms

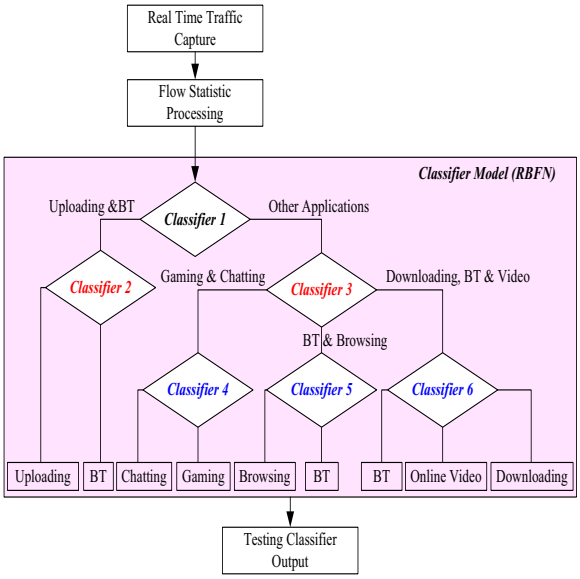


Figure 5: The hierarchical classification structure of RBFN algorithms

neighbors and RBFN. The Mahalanobis distance from a multivariate matrix Y to X is shown as follows:

$$D_M(Y) = \sqrt{(Y-\mu)^T S^{-1}(Y-\mu)}$$

where μ and S are the mean and covariance of X.

For example, to calculate the Mahalanobis distances between *browsing* and *chatting*, we use the data rate in downlink and uplink as the multivariate matrix Y for *browsing* and denote X for *chatting*. $D_M(Y)$ is the Mahalanobis distance from *browsing* to *chatting*. Similarly, $D_M(X)$ is the Mahalanobis distance from *chatting* to *browsing*. After that, we utilize, $E\left(D_M\left(X\right) + D_M\left(Y\right)\right)$, the *average* of these two distances as the distance between *browsing* and *chatting*. We show the similarity in distances of seven applications in Table 4 (in home scenarios with *RSSI* around 55).

Among seven applications, only *upload* and *BT* may have continuous large traffic in uplink. Hence, we adopt *Classifier 1* to distinguish *upload* and *BT* from the rest of the applications in Figure 4 and Figure 5. From Table 4, we notice that the following application pairs have relatively small distances: *browsing* and *BT*, *chatting* and *online gaming*, and *downloading* and *online video*. Therefore, we group each pair and design *Classifiers 3* and *4* to separate the application pairs in SVM algorithms in Figure 4, where *Classifiers 2*, *5*, *6* and *7* are leaf nodes in the binary classification tree. The output of the system is an application for the input flow segment.

Likewise, the hierarchical classification structure of RBFN is shown in Figure 5. A major difference between RBFN and SVM is that a RBFN classifier is able to separate more than two classes. Therefore, we use *Classifiers 3* and *6* for multiple class separation in order to reduce the number of classifiers. *Classifiers 2*, *4* and *5* are binary classifications. Because *BT* has a large range of data rates, sometimes it looks like *downloading* with a large data rate, and sometimes it is like *browsing*. In order to improve the accuracy of detection *BT*, we identify the BT application by *Classifiers 2* and *7* in SVM algorithms, and by *Classifiers 2*, *5* and *6* in RBFN algorithms.

4.5 Classification for Concurrent Applications

A user may open multiple windows and run multiple online applications simultaneously. In this case, the traffic we observe in the MAC layer is a mixture of frames from multiple applications. However, it is hard to separate frames of one application from the others when the frames are encrypted.

Our strategy is to use the proposed hierarchical structure in Section 4.4 to identify the dominating application at first. Then we classify the possible concurrent applications. We denote the window size as W and give a classification result of concurrent applications after L window sizes. For each W, we identify the flow segment of the application it belongs to. Then we get L identified sub-flows to compute the proportion that each application occupied in the L sub-flows. The application with the highest proportion will be the dominating application. Similarly, we can get the application which is the second largest proportion. At the same time, we need to set a threshold for an application to identify its existence. Only if the proportion of the application is larger than its threshold, the application may be regarded as a concurrent application. According to the above strategy, we may infer the possible concurrent applications in the aggregated traffic.

However, the low traffic applications (e.g., chatting and online gaming) are hardly detected. They may be inundated by the dominating applications. In this case, we adjust features to tell if the aggregated traffic includes low bandwidth applications. For example, we adopt *the frame size distribution* and *the number of frames* in small size (e.g., $\leq$ 400bytes) as features to tell if the downloading frames are mixed with chatting. The reason is that downloading has few frames smaller than 400 bytes, and most chatting frames are smaller than 400 bytes. To efficiently identify multiple online applications, we have to reduce the window size W.

Table 5: Overall accuracy for seven applications ($W = 5s$)(%)

Overall accuracy of SVM algorithms: *The Same Location, User in One Day*								
Scenarios	①br.	②ch.	③ga.	④do.	⑤up.	⑥vo.	⑦bt.	Mean
Home	42.690	79.935	98.017	91.285	95.969	91.283	81.734	**82.987**
Public	65.934	70.450	74.041	94.052	87.172	64.319	91.054	**78.146**
University	45.623	66.967	84.399	85.526	91.901	83.984	70.172	**75.510**
Overall accuracy of RBFN algorithms: *The Same Location, User in One Day*								
Home	37.767	77.932	88.181	99.877	95.922	93.321	89.683	**83.240**
Public	48.738	61.488	81.031	94.005	84.748	96.227	91.381	**79.657**
University	28.533	64.442	61.160	95.703	91.901	71.584	90.847	**72.024**
Overall accuracy: *Similar RSSI, Totally Different Location, User and Time*								
Mixed (SVM)	53.164	61.108	67.873	92.740	91.866	72.402	58.218	**71.053**
Mixed (RBFN)	36.362	68.367	58.901	95.386	93.827	90.894	56.023	**71.394**

Table 6: Accuracy and FP in Different Window Sizes (%)

Metrics	Window Sizes	①br.	②ch.	③ga.	④do.	⑤up.	⑥vo.	⑦bt.	Mean
Accuracy	$W = 5s$ (SVM)	42.690	79.935	98.017	91.285	95.969	91.283	81.734	**82.987**
Accuracy	$W = 5s$ (RBFN)	37.767	77.932	88.181	99.877	95.922	93.321	89.683	**83.240**
Accuracy	$W = 60s$ (SVM)	53.571	99.427	100.000	100.000	95.969	100.000	99.692	**92.666**
Accuracy	$W = 60s$ (RBFN)	72.936	85.293	93.742	100.000	95.922	100.000	95.137	**91.861**
FP	$W = 5s$ (SVM)	2.583	1.815	2.930	1.257	0.773	2.621	7.871	**2.836**
FP	$W = 5s$ (RBFN)	2.734	2.212	3.287	0.932	0.020	1.047	9.322	**2.793**
FP	$W = 60s$ (SVM)	0.055	0.734	0.662	0.131	0.000	0.000	6.975	**1.222**
FP	$W = 60s$ (RBFN)	1.507	1.448	1.861	0.129	0.000	0.297	4.255	**1.356**

Thus, the majority of the frames in individual windows are from a single application, and we have more chances to detect concurrent applications in a fixed time duration.

As a user usually does not actively run more than two online applications at the same time, we test our classification system based on two concurrent applications. We show in Section 5 that we can successfully detect the two concurrent applications (the main application and the hidden application) in the aggregated traffic.

5. EVALUATION

Our prototype classification system has been tested at a home, on a university campus, and in public areas. Wireless LANs in these environments all support 802.11a/b/g modes, and the data rate may fluctuate from 1Mbps to 54Mbps. The encryption scheme of WLANs at home and on campus is WPA/WPA2, and the traffic collected in public areas is unencrypted. More than 10 volunteers run the same applications on different devices, operating systems or laptops. An application runs for about 10 minutes each time. In total, we get more than 50 hours of traffic data. In our experiments, we measure the traffic and collect features when the RSSI is larger than 40 (i.e., RF Signal Strength > -70dBm) and the maximum data rate is larger than 50KB/s. Otherwise, users will suffer from poor network quality. We divide the collection data into many groups according to similar RSSI values or scenarios. A similar RSSI means ±10dBm, and scenarios are determined by many factors, such as location, user and time duration of measurement. For each group of data, the training data is randomly selected from the collected data and the rest are used for evaluation, called testing data. In the experiments, the testing data set is a factor of 3 to 10 times larger than the training data set.

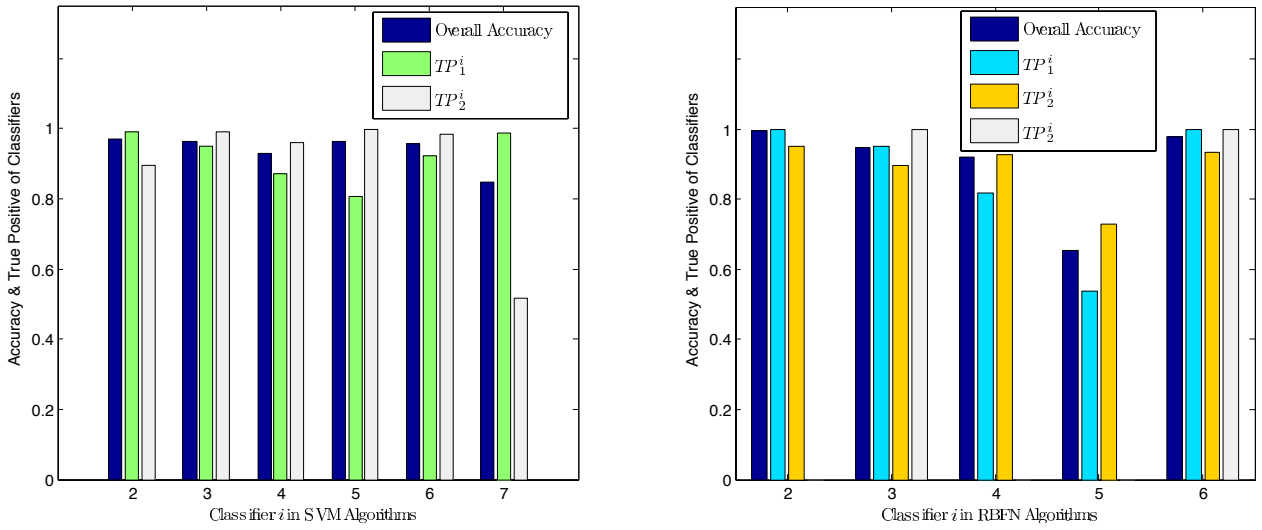
5.1 Performance Metrics

A key criterion of performance evaluation for classification techniques is the accuracy (i.e., how accurately the technique or model classifies the flows) [13], which can be measured by three metrics: *overall accuracy*, *true positive (TP)*, and *false positive (FP)*. The *overall accuracy* is defined as the percentage of correctly classified instances among the total number of instances, and *true positive* means the percentage of members of a given class $\mathscr{X}$ is correctly classified to class $\mathscr{X}$. FP reflects the percent of non-class $\mathscr{X}$ packets incorrectly classified as belonging to class $\mathscr{X}$.

5.2 Overall Accuracy in Different Scenarios

The overall accuracy of different applications and scenarios is listed in Table 5. We set the window size to $W = 5$ seconds. The overall accuracy of classification for seven applications in different scenarios is around 80% when we select the same location and user in one day. We achieve the best accuracy in the home scenario, 82.987%, and the lowest accuracy in the university scenario, 75.510%. The relatively low accuracy in the university environment is caused by traffic interference and collision, which lead to large jitters in the data rate. In contrast, the number of users is very limited in the home scenario, so the interference and collision caused by sharing wireless bandwidth is less than in the public and university scenarios. Public networks likely have a low and restricted data rate. This rate limiting reduces the classification accuracy. The mixed scenario is only distinguished by the similar RSSI. It may include the traffic trace from home, public area, and university. Hence, it has much noise and different traffic patterns in different environments. Therefore, the accuracy in the mixed scenario is the lowest, about 71%, among all scenarios.

Among different applications, browsing has the lowest ac-



(a) SVM (b) RBFN

Figure 6: Overall accuracy and TP of classifiers

curacy. This is caused by two factors. First, browsing applications have a large variance in the data rate. Flash, GIF and video files of advertisements are embedded in many websites which may generate bursty traffic. Second, in a hierarchical classification structure, browsing is in the lowest layer of the classifier model, shown in Figure 4 and 5. Errors from previously layers have accumulated to the classifier of browsing. In summary, high bandwidth applications have better accuracy than low bandwidth applications.

SVM algorithms achieve similar overall accuracy to RBFN algorithms. For high traffic applications, RBFN performs better than SVM, such as downloading, online video and BT. On the contrary, SVM gets better accuracy in low traffic applications, especially in browsing and chatting.

5.3 Accuracy and False Positive in Different Window Sizes (W)

Changing the window size from 5 to 60 seconds, the classification accuracy of each application is presented in Table 6. The accuracy increases simultaneously with the rise in W. This is because the features extracted from a larger window can tolerate more noise. If we collect features every 60 seconds, it achieves more than 90% accuracy; and five applications can be classified with almost 100% accuracy. SVM is more sensitive to increasing window size and performs better than RBFN.

The FP of each application is also listed in Table 6. We can see that the FPs of low traffic applications are mostly higher than those of high traffic applications. BT always has the largest FP in both SVM and RBFN algorithms. That means other applications are misclassified as belonging to BT, especially for browsing. The reason is that the traffic of BT varies very much. It may resembles low traffic applications, or looks like high traffic applications according to network resources.

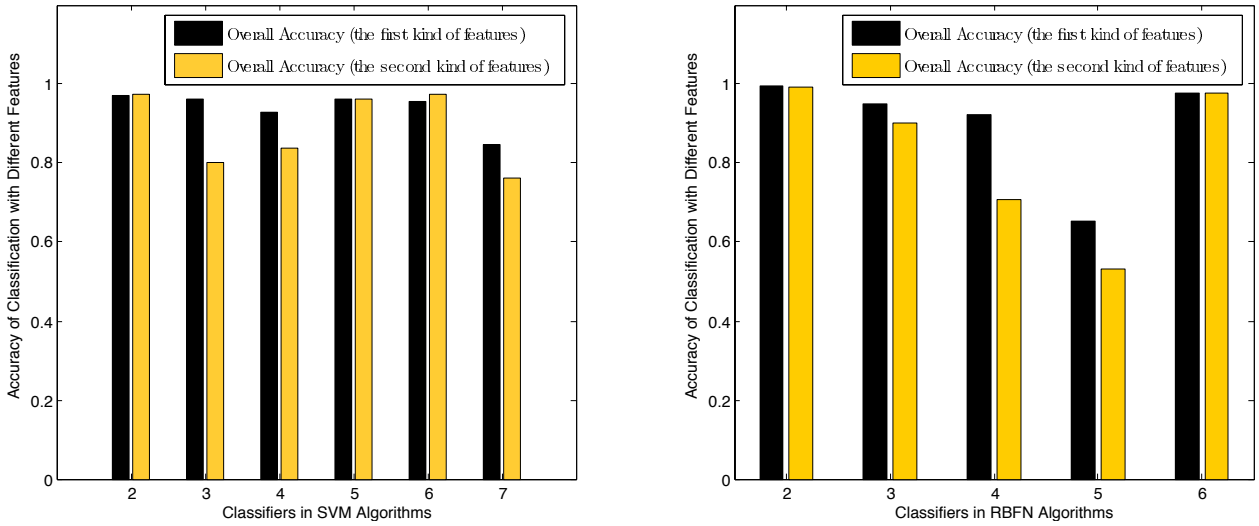
5.4 Overall Accuracy and True Positive of SVM and RBFN Classifiers

We give the overall accuracy and TP of each classifier for SVM and RBFN in Figure 6. The features are generated according to Section 4.2, and W is set as 5 seconds. The classification systems of SVM and RBFN algorithms are shown in Figure 4 and 5. Note that, for simplicity, *Classifier 1* does not use either SVM or RBFN but only employs a threshold of data rate. For example, if the sending data rate is beyond 40KB/s and the average sending rate is above 60KB/s in consecutive 5 seconds, we regard the traffic as probably uploading and then pass it to *Classifier 2*, otherwise to *Classifier 3*.The classifiers, except *Classifier 1*, are shown in Figure 6. TP_j^i indicates the TP of the jth class which is under *Classifier* i in Figure 4 and 5. For example, *Classifier 3* ($i = 3$) of RBFN, TP_1^3 indicates the TP of low traffic applications "chatting & gaming," TP_2^3 shows the TP of "browsing and BT" and TP_3^3 describes the TP of large traffic applications "downloading, online video and BT."

For SVM algorithms, the overall accuracies of the classifiers are beyond 80%; and the TPs are above 85% except browsing, the lowest at 51.534%. This fact is also observed in Section 5.2. Figure 6(b) shows the overall accuracy and TP of RBFN classifiers. All classifiers perform well with high accuracy over 85%, except for *Classifier 5* at 65.380% and the TP of browsing at only 53.946%. It shows TPs are close to accuracies both in SVM and RBFN algorithms. That means our classification system is very balanced. We also observe that RBFN algorithms are good at multiclass classification, since the two triple-category classifiers perform well.

5.5 Feature Selection

Figure 7 compares the overall accuracy of classifications when we use the same hierarchical structure but different features. Note that, classifier 1 is not included because it does not use ML algorithms. For SVM algorithms, we use two pairs of features in one case, *(receiving data rate, sending data rate)* and *(number of receiving frames, number of sending frames)*. In the second case, we adopt one pair of features *(mean receiving frame size, mean sending frame size)*. The



(a) SVM (b) RBFN

Figure 7: Overall accuracy by using different feature selection

Table 7: Detection probability for concurrent applications (%)

Applications	SVM	RBFN
{①br., ⑤up.}	{41.4, 100.0}	{47.8, 100.0}
{①br., ⑦bt.}	{75.7, 100.0}	{54.2, 100.0}
{①br., ②ch.}	{100.0, 36.5}	{100.0, 41.0}
{②ch., ④do.}	{100.0, 100.0}	{100.0, 100.0}
{④do., ⑥vo.}	{100.0, 98.2}	{100.0, 100.0}
{⑥vo., ⑦bt.}	{100.0, 63.7}	{100.0, 65.3}

classification with two pairs of features yields better accuracy.

For RBFN algorithms, we use multiple features, *(mean receiving frame size)*, *the number of frames* and *frame size distribution*, in the first case. The second case only uses one feature, *(mean receiving frame size)*. Figure 7 shows that classifiers using the first kind of features achieve higher accuracy. In summary, more appropriate features will benefit the performance of classification and improve the accuracy.

5.6 Classification for Concurrent Applications

To verify the ability to identify multiple concurrent online applications, we let aggregated traffic (six combinations of two concurrent applications) pass through the proposed classification system. Because we can not be certain that frames from each application will appear in each flow segment, the classification result of each flow segment can not be used to judge the accuracy of concurrent applications. Instead, we use the detection probability to evaluate the performance of our classification system. The result is given every 1 minute when $L = 60$ and $W = 1$ second. If the dominating application is the right aggregated application, we give a detection probability of 100%. Then we compute the detection probability of the second application by dividing its proportion by all applications except the dominating application. Using these methods, we give the detection probability of six combinations in Table 7. We see that our system separates two large bandwidth consumption applications, downloading and online video, with 100% accuracy. A large bandwidth consumption application plus a low bandwidth consumption application (e.g., downloading and chatting) also perform well with 100% accuracy. Because browsing is hard to separate from BT, other combinations have a lower probability of being classified accurately. Chatting has the lowest probability, about 36.474%, when identified from browsing. The performance of SVM algorithms is similar to that of RBFN algorithms.

6. DISCUSSIONS

6.1 Impact of Rate Limiting

For fairness and security, rate limiting software may be used to control the traffic rate of individual users in LANs. Hence, a user can not send or receive frames beyond a specified rate. With such a restriction, if the traffic rate is too low, the rate-related features can not be used to separate high traffic applications from original low traffic applications. However, if the specified rate is not too low (e.g., above 50KBps), the feature-based classification presented in this paper is still valid. In addition, we have examined our classification approach in public networks, where the data rate is limited, and our approach performs well.

6.2 Resistance to Current Defense Methods

Our system can thwart *pseudonyms* [10], because all packets sent under one pseudonym are trivially linkable. Moreover, pseudonym schemes only change MAC addresses each session or when idle. The infrequent change of MAC addresses can not defend the monitoring of an adversary in a few seconds. A link protocol [34] has implemented the function to obscure identifiers; but it can not obscure the traffic features, such as frame interarrival time and frame size distribution. In addition, high-level mitigation policies, such as packet padding, are likely to be ineffective or incur prohibitively high communication overhead [1, 2]. Traffic

morphing [35] defends against traffic analysis in VoIP and web browsing applications by modifying packet sizes. However, other features (e.g., *data rate*) may still be sufficient for classification. Therefore, efficient defense against side-channel information leaks is a future research topic with strong practical relevance.

7. CONCLUSIONS

In this paper, we propose an online hierarchical classification system to identify users' online activities with high accuracy just by peeping at MAC-layer traffic. The classification system is implemented by using ML algorithms, including SVM and RBFN, and achieves high accuracy in different network situations, such as at home, in university and public network environments. The results show that it can distinguish different online applications with around 80% accuracy in just 5 seconds, and the accuracy is over 90% if the eavesdropping duration lasts for 1 minute. At the same time, our classification system can discover the combination of multiple concurrent online applications. Our work shows that the privacy leak of users' online activities is a severe threat in WiFi networks. We expect that our classification system will invoke public attention to user privacy in online activities.

8. REFERENCES

[1] S. Chen, R. Wang, X. Wang, and K. Zhang. Side-channel leaks in web applications: A reality today, a challenge tomorrow. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 191–206, 2010.

[2] Q. Sun, D.R. Simon, Y. Wang, W. Russell, V.N. Padmanabhan, and L. Qiu. Statistical identification of encrypted web browsing traffic. In *Proceedings of IEEE Symposium on Security and Privacy*, 2002.

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

[4] X. Song, D. Wagner, S. David, and X. Tian. Timing analysis of keystrokes and timing attacks on SSH. In *Proceedings of USENIX Security Symposium*, 2001.

[5] T.S. Saponas, J. Lester, C. Hartung, S. Agarwal, and T. Kohno. Devices that tell on you: Privacy trends in consumer ubiquitous computing. In *Proceedings of USENIX Security Symposium*, 2007.

[6] C.V. Wright, L. Ballard, F. Monrose, and G. M. Masson. Language identification of encrypted VoIP traffic: Alejandra y roberto or alice and bob. In *Proceedings of USENIX Security Symposium*, 2007.

[7] C.V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson. Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations. In *Proceedings of IEEE Symposium on Security and Privacy*, 2008.

[8] M. Liberatore and B. Levine. Inferring the source of encrypted http connections. In *Proceedings of Computer and Communications Security*, 2006.

[9] D. Herrmann, R. Wendolsky, and H. Federrath. Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naive-bayes classifier. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 31–42. ACM, 2009.

[10] T. Jiang, H.J. Wang, and Y. Hu. Preserving location privacy in wireless LANs. In *Proceedings of MobiSys*, pages 246–257, 2007.

[11] J. Wilson and N. Patwari. See through walls: Motion tracking using variance-based radio tomography networks. *IEEE Transactions on Mobile Computing*, 2010.

[12] V. Srinivasan, J. Stankovic, and K. Whitehouse. Protecting your daily in-home activity information from a wireless snooping attack. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 202–211. ACM, 2008.

[13] T.T. Nguyen and G. J. Armitage. A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys and Tutorials*, 10(1-4):56–76, 2008.

[14] Internet Assigned Numbers Authority (IANA), August, 2008. http://www.iana.org/assignments/port-numbers.

[15] T. Karagiannis, A. Broido, and M. Faloutsos. Transport layer identification of P2P traffic. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 121–134. ACM, 2004.

[16] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: multilevel traffic classification in the dark. *ACM SIGCOMM Computer Communication Review*, 35(4):229–240, 2005.

[17] K. Xu, Z.L. Zhang, and S. Bhattacharyya. Profiling internet backbone traffic: behavior models and applications. *ACM SIGCOMM Computer Communication Review*, 35(4):169–180, 2005.

[18] C.V. Wright, F. Monrose, and G. Masson. HMM Profiles for Network Traffic Classification (Extended Abstract). In *Proceedings of Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC)*. Citeseer, 2004.

[19] C.V. Wright, F. Monrose, and G.M. Masson. On inferring application protocol behaviors in encrypted network traffic. *The Journal of Machine Learning Research*, 7:2745–2769, 2006.

[20] A. Dainotti, W.D. Donato, A. Pescape, S. Rossi, et al. Classification of network traffic via packet-level hidden Markov models. In *Proceedings of GLOBECOM*, pages 1–5. IEEE, 2008.

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

[22] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli. Revealing skype traffic: when randomness plays with you. *ACM SIGCOMM Computer Communication Review*, 37(4):37–48, 2007.

[23] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian. Traffic classification on the fly. *ACM SIGCOMM Computer Communication Review*, 36(2):23–26, 2006.

[24] M. Tavallaee, W. Lu, and A.A. Ghorbani. Online Classification of Network Flows. In *Proceedings of Seventh Annual Communication Networks and*

Services Research Conference, pages 78–85. IEEE, 2009.

[25] A. Moore and D. Zuev. Internet traffic classification using Bayesian analysis techniques. *ACM SIGMETRICS Performance Evaluation Review*, 33(1):50–60, 2005.

[26] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee. Internet traffic classification demystified: myths, caveats, and the best practices. In *Proceedings of the 2008 ACM CoNEXT conference*, pages 1–12. ACM, 2008.

[27] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall. 802.11 user fingerprinting. In *Proceedings of MobiCom*, pages 99–110. ACM Press, 2007.

[28] J. Bardwell. Converting signal strength percentage to dBm values. *WildPackets*, 2002.

[29] N. Williams, S. Zander, and G. Armitage. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *ACM SIGCOMM Computer Communication Review*, 36(5):5–16, 2006.

[30] C.W. Hsu and C.J. Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

[31] S. Cheong, S.H. Oh, and S.Y. Lee. Support vector machines with binary tree architecture for multi-class classification. *Neural Information Processing-Letters and Reviews*, 2(3):47–51, 2004.

[32] B. Fei and J. Liu. Binary tree of SVM: a new fast multiclass training and classification algorithm. *Neural Networks, IEEE Transactions on*, 17(3):696–704, 2006.

[33] G. Madzarov and D. Gjorgjevikj. Multi-class classification using support vector machines in decision tree architecture. In *Proceedings of EUROCON 2009*, pages 288–295. IEEE.

[34] B. Greenstein, D. Mccoy, J. Pang, T. Kohno, S. Seshan, and D. Wetherall. Improving wireless privacy with an identifier-free link layer protocol. In *Proceeding of MobiSys*, 2008.

[35] C.V. Wright, S.E. Coull, and F. Monrose. Traffic morphing: an efficient defense against statistical traffic analysis. In *Proceedings of NDSS*, 2009.