Detection of Fraudulent Usage in Wireless Networks

Bo Sun, *Member, IEEE*, Yang Xiao, *Senior Member, IEEE*, and Ruhai Wang, *Member, IEEE*

***Abstract*—The complexity of cellular mobile systems renders prevention-based techniques not adequate to guard against all potential attacks. An intrusion detection system has become an indispensable component to provide defense-in-depth security mechanisms for wireless networks. In this paper, by exploiting regularities demonstrated in users' behaviors, we present a suite of detection techniques to identify fraudulent usage of mobile telecommunication services. Specifically, we explore users' behaviors in terms of calling and mobility activities because they are two of the most important components of mobile users' profiles. To utilize users' calling activities, we formulate the intrusion detection problem as a multifeature two-class pattern-classification problem. Parameters including call-duration time, call inactivity period, and call destination are extracted to form a feature vector to reflect users' calling activities. A nonparametric technique known as the Parzen window with a Gaussian kernel, is used to estimate a class-conditional probability density function. A Bayesian decision rule is applied in order to achieve a desirable error rate. To effectively exploit movement patterns demonstrated by mobile users, we first propose a realistic network model integrating geographic road-level granularities. Based on this model, an instance-based learning technique is presented to construct mobile users' movement patterns. A user's movement history is stored and compared against newly observed movement instances. We then define a novel similarity threshold to classify users' current movement activities. We simulate users' various behaviors and provide simulation results.**

***Index Terms*—Bayesian decision rule, instance-based learning (IBL), intrusion detection, wireless network.**

I. Introduction

THE UBIQUITOUS infrastructure, while dramatically increasing functionality levels, has posed significant security concerns on cellular mobile networks. Although there are many security protocols that have been proposed for cellular mobile networks, how to design a highly secure cellular mobile network still remains a very challenging issue due to open radio-transmission environment and physical vulnerability of mobile devices.

Generally speaking, the following two complementary classes of approaches exist to protect a system: prevention-based and detection-based approaches. Security research into wired networks indicates that there are always some weak points in the system that are hard to predict. This is particularly true for a wireless network, in which open wireless-transmission media and low physical-security protection of mobile devices pose additional challenges for prevention-based approaches. For example, although security measures are taken into account in the designs of second-generation (2G) and third-generation (3G) digital cellular systems, security flaws keep being reported in the literature [1]–[3]. One of the basic threats is the illegitimate use of services, which can lead to the problem of improper billing and masquerading and can cause drastic damage to service providers. Therefore, in order to provide defense-in-depth security mechanisms, a multilayer/multilevel protection system is necessary. Serving as the first level of protection schemes, prevention-based approaches (such as authentication and encryption) can effectively reduce attacks by keeping illegitimate users from entering the system. However, if a device is compromised, all the secrets associated with a device become open to attackers, rendering all prevention-based techniques helpless and resulting in great damage to the whole system. At this time, intrusion detection systems (IDSs), serving as the second level of protection schemes, if well designed, can effectively identify malicious activities and help offer an adequate protection for the system.

In this paper, by exploiting mobile users' calling patterns and the users' exhibited location history, we present a suite of detection techniques to identify a group of particularly harmful insider attackers—the masqueraders. Our work is based on such an observation that most mobile users demonstrate certain regularities in their daily lives. For example, because of regular working rhythms such as daily or weekly business telephone conferences, most users exhibit certain calling patterns. As another example, a mobile user usually travels with a specific destination in mind and tends to follow the shortest path to it. A user's mobility pattern is a reflection of the routines of his daily life, and most mobile users have favorite routes and habitual movement patterns. Although an attacker can compromise all the secrets associated with a mobile device, he could not intimate the authentic user's profiles. This observation is particularly true if the malicious users intend to cause drastic damages. All these will make the adversary exhibit significant skewed behaviors from that of the authentic user. Therefore, by establishing an accurate profile of the mobile user and by comparing it with the current observed activities, malicious activities can be effectively identified.

Motivated by the above observations, we aim at designing practical intrusion detection techniques for cellular mobile networks. We focus on users' calling and mobility activities because they represent two of the most important components of mobile users' profiles. Specifically, to utilize users' calling

Manuscript received March 17, 2006; revised January 8, 2007 and March 10, 2007. This work was based on work supported by the Texas Advanced Research Program under Grant 003581-0006-2006 and the US National Science Foundation (NSF) under Grants DUE-0633445, CNS-0716211, and CNS-0737325. The review of this paper was coordinated by Dr. K. Martin.

B. Sun is with the Department of Computer Science, Lamar University, Beaumont, TX 77710 USA (e-mail: bsun@cs.lamar.edu).

Y. Xiao is with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487 USA (e-mail: yangxiao@ieee.org).

R. Wang is with the Department of Electrical Engineering, Lamar University, Beaumont, TX 77710 USA (e-mail: wang@ee.lamar.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2007.901875

activities, we first apply the Chebyshev inequality to eliminate obvious malicious calls. This can lead to a reduced number of false alarms. We then formulate the intrusion detection problem as a multifeature two-category pattern-classification problem. Call duration time (CDT), call inactivity period (CIP), and call destination (CD) are extracted to form a feature vector to reflect users' calling activities. A nonparametric technique, known as the Parzen-window approach with a Gaussian kernel, is used to estimate a smooth class-conditional density function. Because of potential fraudulent usage of wireless services, normal and abnormal usages may demonstrate distinct behaviors. Therefore, a model of anomaly detection based on Bayesian decision rule is then introduced, and its performance is discussed in terms of false positive and detection rates. To utilize users' mobility activities accurately and effectively, we first propose a realistic network model integrating geographic road-level granularities. The proposed network model takes into account both users' moving patterns and an actual location management scheme in the current cellular system; i.e., whenever a user crosses a boundary of a location area (LA), a location update operation is performed. Based on this model, we present an instance-based-learning (IBL) technique to construct users' movement profiles. A similarity measure is defined to compare a user's activity with its constructed normal profile. A threshold policy is then used to decide whether the current activity is normal or not.

The rest of the paper is organized as follows. In Section II, we further describe our motivations to develop IDSs for cellular mobile networks. Section III describes the related work. Section IV presents our threat model, network model, and assumptions in developing intrusion detection schemes for wireless networks. In Section V, we present Bayesian-decision-rule-based detection algorithms, which utilize calling activities. In Section VI, we present IBL-based detection algorithms, which utilize mobility activities. Simulation results of both algorithms are shown in Section VII. We conclude this paper and point out a future work in Section VIII.

II. Motivation

For most mobile users in wireless networks, their profiles, in terms of calling and mobility activities, demonstrate some regularity. Some examples include the following.

1) A user may have daily or weekly business telephone conferences because of regular working rhythms, and conferences may last a certain period of time because of his schedule made in advance.
2) Billing plans subscribed by mobile users motivate them to make short calls during daytime, whereas relatively longer calls with friends and family members are made on nights and weekends to avoid extra charges.
3) Daily commuting patterns of public-transportation users are very regular. For example, studies in [24] conducted experiments over a period of six weeks to study trajectories that users follow and found out that users tend to follow regular trajectories more than 70% of time.

Behaviors of fraudsters, on the other hand, demonstrate a skewed distribution. All of these motivate us that we can learn a mobile user's behavior to construct his normal profile based on detection techniques constructed to identify whether the user is an intruder or not by comparing his current activities with his established patterns.

Because of the potential wide variety of users' behaviors, there are a certain number of users who do not exhibit regular patterns. For example, it is not easy to model the movement patterns of taxi drivers. This kind of highly irregular yet legitimate behaviors may result in inaccuracy in their established normal profiles. Therefore, we should not expect that our detection based on users' profiles is accurate for all users in all situations. We realize that, even for a normal user who demonstrates very regular patterns, it is still possible to have deviations of his normal profiles.

Based on these considerations, our paper is not motivated to build a system to accurately detect all intrusions. We do not expect our system to have zero false positives either. IDSs will provide a complementary layer of protection for a system. We do not expect an IDS to be available for all users under any situation. Instead, our objective is to provide an optional service to end users, as well as a useful administration tool for service providers. A similar strategy has been used in credit-card companies. For example, a customer will be called or alarmed if an abnormal usage of his credit card is detected. For example, his card was used in another country that is not his residence and that he does not frequently visit.

In order to model users' behaviors comprehensively, our detection technique based on mobility activities needs to track users' locations. This will give rise to users' location-privacy issues. Therefore, our system provides a user with an option to turn off this service. Privacy concerns must be properly addressed before we can deploy this kind of service. It is worth noticing that location-privacy issues have attracted much attention from the research community [23]. Therefore, it is promising to integrate our proposed service with other existing location-privacy protection schemes.

With a prominent growth of mobile users and the ubiquity of wireless networks, the approaches proposed in this paper are general and can be used in many different applications. For example, mobile users calling in a car fit with the application naturally. The mobility-based approach can also fit the applications of other wireless networks. For example, in a campus wireless network, the location of each user can be recorded and compared with his established normal profile in order to identify potential intruders.

III. Related Work

As mentioned before, there are two important intrusion detection techniques as follows: misuse detection and anomaly detection. A good taxonomy of existing technologies is presented in [4]. The research of intrusion detection began with Denning's seminal paper [5]. Since then, many research efforts have been devoted to different detection techniques, for example, expert system [7], colored Petri nets [8], state-transition analysis [9], neural networks [10], and so on. There are also some fraud-detection systems in telecommunication systems. Data mining [17], machine learning [18], etc. have been

utilized to detect fraud data in telecommunication networks. All existing approaches take into consideration domain-specific knowledge to build suitable detection systems.

Relatively few research efforts have been devoted to intrusion detection research of wireless networks. In [12], Samfat and Molva proposed an intrusion detection architecture for mobile networks, which includes two algorithms to model the behavior of users in terms of both telephony activity and migration patterns. Lin *et al.* [2] proposed an excellent study to detect the potential fraudulent usage of cloned phones in cellular mobile networks. Sun *et al.* [1], [6] proposed a mobility-based detection system to identify potential masqueraders in cellular mobile networks. Büschkes *et al.* [20] presented an approach applying the Bayesian decision rule to user's mobility profile to increase security for wireless networks.

IV. Assumptions

All research into intrusion detection is based on the following assumptions: 1) activities of a subject are observable via auditing mechanisms of some systems, and 2) normal and malicious activities should demonstrate distinct behaviors. Our paper is no exception. Besides these, we make the following assumptions.

First, we assume that our proposed detection schemes can integrate with the existing databases in wireless networks, such as a calling history database to describe users' calling activities and a mobility database to describe users' mobility behaviors. This information may already exist in the system for other services. For example, a database recording calling history is a necessary element to support billing services. We further assume that this information is accurate and secure through adequate protection measures. This is a realistic assumption given the importance of these databases. To support the realistic network model that we propose, we also assume the existence of a path database that illustrates a digital map of the service area. It is worth mentioning that much work has been carried out in this area, aiming at enhancing the various aspects of quality of service (QoS) in wireless networks [25].

Second, we assume that most users' behaviors demonstrate certain regularities. This makes it viable for us to reason that the evidence in data establishes their normal profiles and determine whether the system is currently under attack. This assumption is reasonable if we consider most of the users' regular daily/weekly working rhythms. Our detection algorithms alone are not suitable for users who demonstrate totally random behaviors. Nevertheless, our method is automatically user-selective since the optional warning service mentioned before will tend to give many false warning messages to this type of users and force them to unsubscribe/disable such a service.

Third, we assume that mobile devices can be compromised and that all secrets associated with the compromised devices are open to attackers. Under this assumption, we do not need to assume or apply tamper-resistant hardware and software, which are still costly and impractical to handheld devices. This assumption justifies our research in anomaly detection since all prevention-based techniques will be rendered helpless once the mobile device is captured and compromised. If we could assume tamper resistance of hardware/software, the whole security research could become much easier.

A. Threat Model

The complexity of a wireless mobile network system could incur software and design errors. This could make many attacks possible. One example is cell-phone cloning: a mobile phone card of an authenticate user A is cloned by an attacker B, which enables B to use a cloned phone card to make fraudulent telephone calls. The legitimate phone user A gets billed for the cloned phone's calls. In addition, the masquerader can fake the International Mobile Equipment Identifier and Subscriber Identity Module card in order to get the service illegally. In subscription fraud, fraudsters can also subscribe to the service using the authentic user's name and obtain an account without the intention of paying the bill.

B. Network Model

Most of the previous work on intrusion detections for wireless cellular networks uses structured-graph network topology models, such as hexagonal or square-cell configurations. The cells are usually determined by the architecture of the cellular networks. However, in practice, considering the fact that a mobile user usually drives along the road, cell-based models may not precisely locate a mobile user and model the trajectory of a user because they do not support the fine granularity of road network [13]. Moreover, most users tend to follow speed limit signs when driving. In addition, each user has his preference of traveling speed. Therefore, it takes a user roughly the same amount of time to travel a specific path (we will not consider the possible traffic jam in this paper). In reality, there also exists a road network, which is overlapped with a location area (LA), which consists of several or many cells. Considering all these, we propose the following network model, as shown in Fig. 1.

Fig. 1(a) shows the network topology in one LA, which consists of seven cells. In Fig. 1(a), straight bold lines represent the road network. Each hexagon represents one cell. v_1, v_2, v_3, and v_4 represent the intersection points of the road network and the boundary of the LA. For current mobile systems, location updates happen when a user enters or leaves one LA. This is one of the most common ways to track the cellular mobile phones. It is true whenever a user is making a phone call or not. Considering these, we propose the network model, as shown in Fig. 1(b).

In Fig. 1(b), each intersection of the road segment and the LA is modeled as a vertex. In our example, we have four vertices, v_1, v_2, v_3, and v_4. These vertices form a fully connected graph, meaning that there is one path between any two vertices. In this way, we can ignore the complex internal road network inside one LA.

It is possible that, in one LA, there is more than one possible path connecting two vertices. We assume that, in one LA, one user prefers one specific path. This means that, in Fig. 1(b), for a specific user, it will take him roughly the same amount of time to travel between any two vertices. If one user has variations in

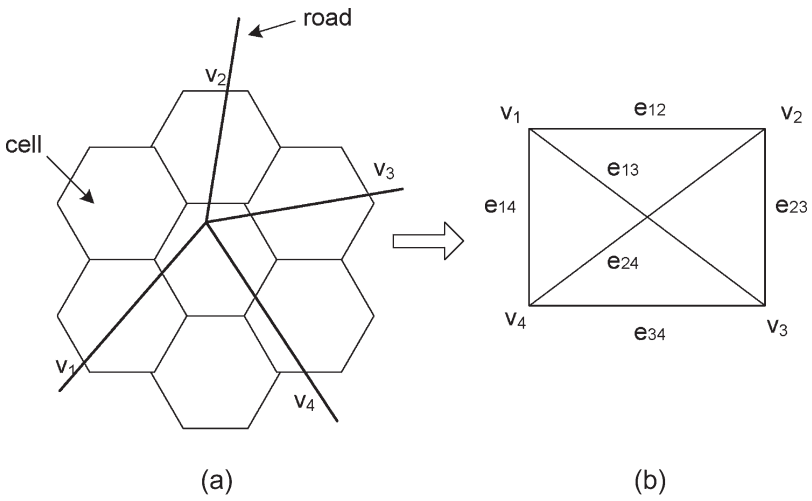

Fig. 1. Topology model. (a) One LA. (b) Network model.

his traveling habit, i.e., if he takes two different paths between the same two vertices, we can have two edges connecting these two vertices in Fig. 1(b).

The network model shown in Fig. 1 is more accurate than the model only considering the cell list traversed by each user, considering the current mechanisms that mobile networks use to track users' location information. Furthermore, most road segments have associated speed limits, and most users have a driving habit. For example, some users want to strictly follow the speed limit, whereas others tend to drive 10 mi/h faster. This will take different users different amounts of time to traverse a specific road segment (edge).

This network model is also more realistic than a model considering the actual path topology. For example, Karimi and Liu [13] proposed a network model that uses edges to model routes and vertices to model traffic lights. Although this model is very accurate, however, in practice, it is difficult for us to track a user's location information based on this model in current 2G/3G cellular networks. A location update often happens when a user traverses the LA border. When the user is inside the LA, if the user is not making a phone call, the user's location information is not visible to the system. Therefore, although our model ignores potential different routes between two vertices, it fully takes into consideration the information that can be provided by current cellular mobile networks, which makes it suitable for our IDS.

V. Calling-Activity-Based Detection Algorithm

In this section, we present our detection scheme based on users' calling activities. Features such as CDT, CIP, and CD are extracted to form the feature vector to reflect users' calling activities. Because of the potential wide variety of user's calling activities, we first adopt Chebyshev inequality to eliminate obvious abnormal calls. This helps in decreasing the false-positive rate. We then formulate the intrusion detection problem as a multifeature two-class pattern-classification problem. A nonparametric technique, known as the Parzen-window approach with a Gaussian kernel, is used to estimate the smooth class-conditional density function of the feature-vector values.

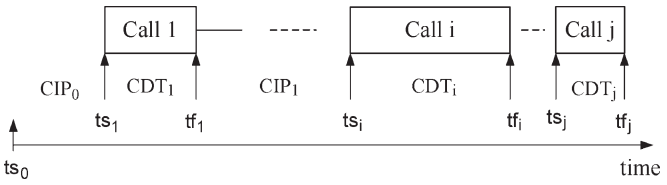

Fig. 2. Call vectors.

Based on this, the Bayesian decision rule is then applied in order to achieve the desirable error rate in terms of false-alarm and detection rates.

A. Feature Selection

The first step in intrusion detection is to extract effective features. Features are security-related measures that can be used to construct suitable detection algorithms. Effective features must be selected to reflect the activities of a subject. If the dimension of the feature vector is too small, they may not reflect a variety of calling activities. On the other hand, if the dimension of the feature vector is too large, it results in a heavy computational overhead.

Basically, the calling activities of an observed user can be represented using Fig. 2. When the mobile station initiates and terminates call i, the system can record ts_i (the time when the ith call starts) and tf_i (the time when the ith call terminates). From the collected data based on Fig. 2, we can compute the statistical call vector of the user. One advantage in defining ts_i and tf_i is that they are easily defined with the help of existing signaling messages. A similar approach is also adopted in [12], which can be used to characterize the calling activities of each user. In this way, the incurred overhead in the mobile networks can be minimized.

Based on the above considerations, we adopt a tuple definition as follows to represent calling activities of a mobile user: CDT, CIP, and CD. We adopt these features because they can be used to represent users' calling activities. It is worth mentioning that feature selection is domain-specific and infamously hard to handle. How to select effective features turns out to be a very

challenging research topic. We leave it as one of our important future works.

1) CDT represents the duration that a call lasts. For any ith call of a user, let ts_i and tf_i denote the time when the call starts and the time when it finishes, respectively. The ith CDT is defined as $(\mathrm{tf}_i - \mathrm{ts}_i)$.
2) CIP represents the time period between the time instant when a new call is initiated and the time instant when the previous call was finished. The ith CIP is defined as $(\mathrm{ts}_{i+1} - \mathrm{tf}_i)$.
3) CD represents the destination of a call.

Based on a user's calling habit, we can treat a group of CDs with the same properties of CDT and CIP as one CD. For example, calls that have different country codes may have the same statistical properties of CDT and CIP. We can combine the different CDs of these international calls into one CD. In this way, the number of states maintained for the user can be effectively reduced. Similar strategies can also be further applied to national calls based on the user's behaviors.

B. Bayesian Decision Rule

Let $\mathbf{x} = (\mathrm{CDT}, \mathrm{CIP}, \mathrm{CD})$ denote the feature vector $\mathbf{x}$, which is in a 3-D Euclidean space $\mathbf{R}^3$ (feature space). The feature space includes all possible states of a user's calling activities. We then apply the Bayesian decision rule to classify these calling activities.

Let $\{\omega_1, \ldots, \omega_c\}$ denote the finite set of c states (categories). Let the feature vector $\mathbf{x}$ be a d-dimensional vector-valued random variable. We use an uppercase $P(.)$ to denote a probability-mass function and a lowercase $p(.)$ to denote a probability-density function (pdf). Bayesian formula could be used to determine the *a posteriori* probability $P(\omega_j|\mathbf{x})$ in the following way:

$$P(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j)P(\omega_j)}{p(\mathbf{x})} \tag{1}$$

where $p(\mathbf{x}|\omega_j)$ is the class-conditional probability density of $\mathbf{x}$ for a given class ω_j. It is also called the likelihood of ω_j with respect to $\mathbf{x}$. $P(\omega_j)$ is the *a priori* probability of class ω_j. $p(\mathbf{x})$ is the probability density of the observed feature vector $\mathbf{x}$, and it is constant for every class ω_j for a particular $\mathbf{x}$.

Given the sample $\mathcal{D}$, (1) then becomes

$$\begin{aligned} P(\omega_j|\mathbf{x}, \mathcal{D}) &= \frac{p(\mathbf{x}|\omega_j, D)P(\omega_j|D)}{\sum_{i=1}^{c} p(\mathbf{x}|\omega_i, D)P(\omega_i|D)} \\ &= \frac{p(\mathbf{x}|\omega_j, D_j)P(\omega_j)}{\sum_{i=1}^{c} p(\mathbf{x}|\omega_i, D_i)P(\omega_i)}. \end{aligned} \tag{2}$$

Here, we separate the training samples by class into c subsets $D_1, D_2, \ldots, D_c$. Equation (2) suggests that we can use the information provided by the training sample to help determine both the class-conditional densities and the *a priori* probabilities.

In (1), $p(\mathbf{x})$ is unimportant as far as making a decision is concerned. As we can see from (2), the purpose of $p(\mathbf{x})$ is to normalize $P(\omega_j|\mathbf{x}, \mathcal{D})$ to ensure that the *a posteriori* distribution $P(\omega_j|\mathbf{x}, \mathcal{D})$ integrates or sums to one. We can calculate the *a priori* $P(\omega_i)$ based on the collected training data. $P(\omega_i)$ can also be assigned based on some domain knowledge. For example, if we have n vectors and n_i of them are of class ω_i, then the empirical probability of $P(\omega_i)$ is estimated as $P(\omega_i) = n_i/n$.

$p(\mathbf{x}|\omega_j)$ is more difficult to compute. In order to compute $p(\mathbf{x}|\omega_j)$, we can divide the feature-vector space into intervals and count the number of vectors falling into every interval. This approach only works when the number of intervals and the dimensions of the vectors are both small. In our case, because of the continuous nature of CDT and CIP, it is very difficult to decide a proper interval for each feature value. Other possible approaches include the classical maximum-likelihood estimation or the Bayesian estimation if the forms of density functions of $p(\mathbf{x}|\omega_j)$ are known [15]. However, this assumption is suspicious given the fact that common parametric forms rarely fit the densities of the potential wide variety of a user's calling activities.

Considering all these, we utilize a nonparametric approach based on a Parzen window to estimate the class-conditional PDF $p(\mathbf{x}|\omega_i)$, as detailed in Section V-D.

We use ω_1 to denote the normal state and ω_2 to denote the abnormal state. In this way, the intrusion detection problem can be formulated as a multifeature two-state pattern-classification problem. In order to classify, we resort to the Bayesian decision rule [15]

$$\textbf{If } P(\omega_1|\mathbf{x}) > P(\omega_2|\mathbf{x}) \quad \textbf{Decide } \omega_1 \quad \textbf{Else} \quad \textbf{Decide } \omega_2. \tag{3}$$

C. Classification Error Rate

When a new observation (a new phone call) $\mathbf{x}$ is made, the probability of error could be defined as

$$P(\mathrm{error}|\mathbf{x}) = \begin{cases} P(\omega_1|\mathbf{x}), & \text{if we decide } \omega_2 \\ P(\omega_2|\mathbf{x}), & \text{if we decide } \omega_1. \end{cases} \tag{4}$$

The decision rule illustrated in (3) could minimize the average probability of error. Let us consider the example shown in Fig. 3. Here, we use the 1-D feature vector for the purpose of illustration. Its probability of error is defined as

$$\begin{aligned} P(\mathrm{error}) &= \int_{-\infty}^{\infty} P(\mathrm{error}, \mathbf{x}) d\mathbf{x} \\ &= P(\mathbf{x} \in \Re_2, \omega_1) + P(\mathbf{x} \in \Re_1, \omega_2) \\ &= \int_{\Re_2} p(\mathbf{x}|\omega_1)P(\omega_1)d\mathbf{x} + \int_{\Re_1} p(\mathbf{x}|\omega_2)P(\omega_2)d\mathbf{x}. \end{aligned}$$

That is, given the observed feature vector $\mathbf{x}$, we can use the Bayesian decision rule to classify the calling activities.

D. Density Estimation Using Parzen Window

As we stated before, it is very difficult to calculate $p(\mathbf{x}|\omega_i)$. Due to the potential diversified user behaviors, it is unrealistic to

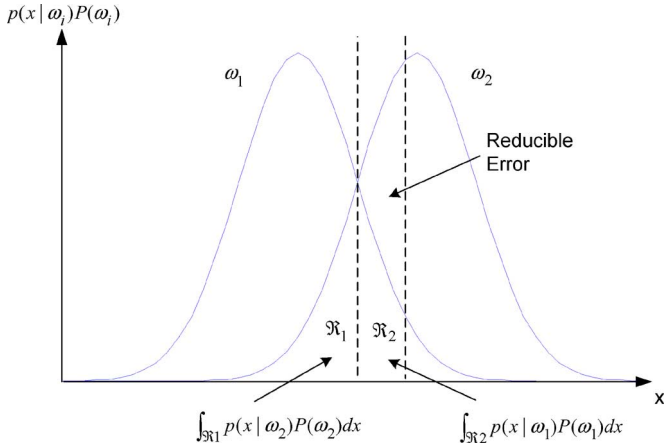

Fig. 3. Classification error rate of Bayesian decision rule.

assume that the underlying pdf is known. Therefore, we resort to the Parzen window [15]—a very popular nonparametric approach used to estimate probability densities.

Let $p(\mathbf{x})$ be the pdf to be estimated. One of the most fundamental techniques to estimate any unknown density relies on the fact that the probability P that a vector $\mathbf{x}$ falls in a region $\Re$ is given by $P = \int_{\Re} p(\mathbf{x}')d\mathbf{x}'$.

In this way, we can estimate the smooth value of p by estimating the probability P. The Parzen-window approach to estimate densities can be assumed that the region $\Re$ is a d-dimensional hypercube. If h_n is the length of an edge of the hypercube $\Re$, its volume is given by $V_n = h_n^d$.

To find the number of samples that fall within the hypercube $\Re$, we define the following window function:

$$\varphi(\mathbf{u}) = \begin{cases} 1, & |u_j| \le 1/2; \quad j = 1, \ldots, d \\ 0, & \text{otherwise.} \end{cases}$$

$\varphi(\mathbf{u})$ is known as a Parzen window. Thus, the number of samples in this hypercube is given by $k_n = \sum_{i=1}^{n} \varphi((\mathbf{x} - \mathbf{x_i})/h_n)$. The estimate is obtained as $p_n(\mathbf{x}) = (1/n)\sum_{i=1}^{n}(1/V_n)\varphi((\mathbf{x} - \mathbf{x_i})/h_n)$.

As we can see, the window function $\varphi(\mathbf{u})$ is being used for interpolation—each sample contributing to the estimate in accordance with its distance from $\mathbf{x}$.

To examine the effect that the window width h_n has on $p_n(\mathbf{x})$, function $\delta_n(\mathbf{x})$ is defined as $\delta_n(\mathbf{x}) = (1/V_n)\varphi(\mathbf{x}/h_n)$. Then, $p_n(\mathbf{x})$ is defined as $p_n(\mathbf{x}) = (1/n)\sum_{i=1}^{n} \delta_n(\mathbf{x} - \mathbf{x_i})$.

The window function $\varphi(\mathbf{u})$ has several drawbacks. It can yield density estimates that have discontinuities. In addition, all the data points in the hypercube $\Re$ centered around the estimation point weight equally, regardless of their distance to the estimation point [15]. To overcome these potential drawbacks, we adopt a commonly used multivariate Gaussian density function as our smooth kernel function $\varphi(\mathbf{x}) = (1/(2\pi)^{d/2})\exp[-(1/2)\mathbf{x}^T\mathbf{x}]$, where $\mathbf{x}$ is a d-dimensional column vector.

It is easy to see that $\int_{\Re^d} \varphi(\mathbf{x})d\mathbf{x} = 1$. In our later simulation, d is set to three. h_n also plays an important effect on the estimation of $p_n(\mathbf{x})$. A too large h_n leads to little resolution on $p_n(\mathbf{x})$, whereas a too small h_n leads to too much statistical variability [15]. In our simulation shown later, we adopt different h_n values to adjust the smoothness of the density estimates until the compromise is acceptable, as suggested in [15].

E. Chebyshev Inequality

Because of the potential diversified user calling activities, it is possible that a normal user's activity is very skewed. We use the Chebyshev inequality [21] to preprocess these kinds of behaviors.

Chebyshev inequality states as follows: let x be a random variable with a mean μ and a variance σ^2; then, for all positive t, $P(|x - \mu| > t) \le \sigma^2/t^2$.

The Chebyshev inequality provides an upper bound on the probability that the value exceeds a certain threshold t from the mean μ of the variable. A value r is identified as an anomaly if its distance from μ is larger than the threshold. In the training phase, we approximate the mean μ and the variance σ^2. In the detection phase, we can specify a threshold t and use σ^2/t^2 to denote the threshold.

In our context, x represents CDT or CIP. We substitute t with the distance between the current r value of CDT/CIP and its mean μ (i.e., $|r - \mu|$). This gives us an upper bound of the probability that the feature value deviates from its mean

$$P\left(|x - \mu| > |r - \mu|\right) < P(r) = \frac{\sigma^2}{(r - \mu)^2}. \tag{5}$$

The Chebyshev inequality is independent of the underlying distribution. In practice, when σ^2 is large, a very large t is needed in order to specify a reasonable bound. This may admit a very skewed CDT/CIP. Because of this, we only apply the Chebyshev inequality when σ^2 is small. In this way, the false alarms can possibly be reduced [21], [22].

VI. Mobility-Based Detection Algorithms

In this section, we detail our mobility-based detection schemes based on the IBL technique. IBL-based approaches have been used before to detect malicious users' activities. For example, in [26], Lane and Brodley applied IBL-based approaches to user-oriented anomaly detection at the level of shell command input. Different from them, our approach is based on a different network model. In addition, based on our novel network model, we take into consideration the probability with which a user takes each path.

A. Feature Extraction

Based on the network model shown in Fig. 1, we extract (e, t) traversed by the user as the feature. Here, e denotes the edge [represented by the vertex pair to denote the path between two vertices, for example, (v_1, v_4)], whereas t is the time the individual user takes to traverse the edge. Because each user has his own driving habit, if there is no traffic jam, it usually takes each user roughly the same amount of time to traverse a specific path. Therefore, the sequence of (e, t) could be used effectively to reflect the user's movement patterns.

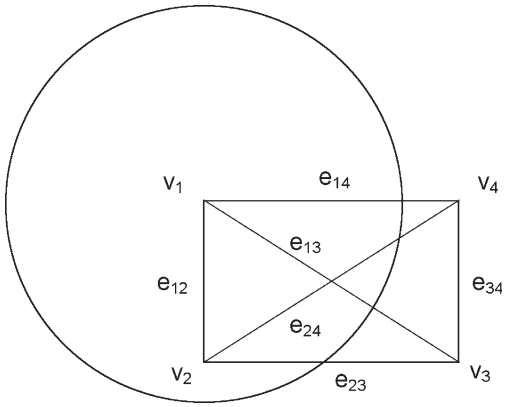

Fig. 4. Vertex prediction.

B. Data Preprocess

It is shown that mobility prediction can significantly improve the performance of mobility management, QoS provisioning, and resource management in cellular mobile networks. Our work can benefit from existing mobility prediction schemes. Yet, they have differences. We can use the existing mobility prediction schemes to help us reduce the false positives. For example, based on the mobility prediction mechanisms, we can predict the potential set of routes that the user will traverse. If the next monitored route is not in this set, we can raise an alarm immediately and will not go through the normal-profile detection process. Some very simple prediction schemes can be used here to avoid the complex modeling process.

In the example shown in Fig. 4, we suppose that the user is traveling from v_1 and that the highest speed limit is S. Based on this, after a period of time T, the location of the user will be limited within the circle, whose radius is equal to $S \times T$. This indicates the longest distance that the user can drive in the time period T. Given an example network topology shown in Fig. 4, the potential set of the next possible vertices is v_2. That is, after T, if the monitored vertices include v_3 or v_4, we can directly generate an alarm. Another example could be that if, 1 h ago, the user is in Houston, it is impossible for the user to be in Los Angeles now. A data preprocessing like this can generate alarms very quickly and accurately.

C. IBL Approach

In contrast to learning methods that construct a general description of the subject activities, IBL methods simply store the training examples. IBL has the advantage that it constructs only a local approximation of the subject behavior, which is suitable when the subject behavior is potentially varied [27].

One example network model and user-mobility behavior is shown in Fig. 5(a). In this example, 0, 1, 2,... are the vertices. At each vertex, the user may have different probabilities to take different paths. Each edge is associated with two numbers. For example, (1/3, 5) at vertex 0, where 1/3 means that the probability that the user takes this path when the user at vertex 0 is 1/3, and five means that it takes the user five units of time to traverse this edge. Fig. 5(b) shows the corresponding path-probability matrix of this user. Each element at (i, j) indicates the probability for this user to take the path (i, j) at vertex i.

A normal profile consists of all the paths traversed by the user. Each path is associated with a probability. Suppose for a given path $R = \{r_1, r_2, \ldots, r_n\}$, where r_i denotes an edge in Fig. 5(a), the probability of each r_i is p_i, and the probability of taking path R is calculated as $P(R) = \prod_{i=1}^{n} p_i$. Here, we omit the problem about how to efficiently store the traversed paths and to retrieve the information to perform anomaly detection. For simplicity, a tree or a matrix can be used in our context.

D. Similarity Computation

1) Similarity Measure Between Two Equal-Length Paths: We treat each path as a string consisting of a sequence of characters and compute the similarity between two paths. This can help us to determine the anomaly of the observed movement activities. We first calculate the similarity between two equal-length paths. Suppose that we have two strings of equal length l: $X_l = (x_0, x_1, \ldots, x_{l-1})$ (a test string) and $R_l = (r_0, r_1, \ldots, r_{l-1})$ (a string in the normal profile). The similarity of two characters at location i is defined as

$$w(X_l, R_l, i) = \begin{cases} 0, & \text{if } i < 0 \| x_i \neq r_i \\ 1 + w(X_l, R_l, i-1), & \text{if } x_i = r_i. \end{cases}$$

The similarity between X_l and R_l is then defined as $\mathrm{Sim}(X_l, R_l) = \sum_{i=0}^{l-1} w(X_l, R_l, i)$.

This definition of similarity considers the sequential characteristics of the path. The converse measure, the distance between X_l and R_l, is defined as $\mathrm{Dist}(X_l, R_l) = \mathrm{Sim}_{\max} - \mathrm{Sim}(X_l, R_l)$, where $\mathrm{Sim}_{\max}$ is the maximum value of the similarity between X_l and R_l. That is, $\mathrm{Sim}_{\max} = l(l+1)/2$.

By defining the similarity measure this way, the contiguous matching of characters tends to lead to a relatively large similarity value. The mismatch of one character, particularly in the middle of compared strings, can greatly reduce the similarity value. This approach encourages the largest contiguous matching of characters. Fig. 6 shows this when comparing two five-character strings with one mismatch. This phenomenon will become more obvious with a larger l.

2) Similarity Measure Between Test String and Normal Profile: A normal profile N consists of a number of paths R_i. Each path is associated with a probability. It is possible that the number of routes (i.e., the length of the path) traversed by a normal user is larger than l. In order to break the collected path into a set of subpaths of length l, we slide a window of length l over the path, each time by one position.

For example, given a path $R_6 = [(e_2, t_2), (e_7, t_7), (e_1, t_1), (e_9, t_9), (e_8, t_8), (e_5, t_5)]$ with a probability P. Suppose that l is set to four. We have the following l-length subpaths: $[(e_2, t_2), (e_7, t_7), (e_1, t_1), (e_9, t_9)]$, $[(e_7, t_7), (e_1, t_1), (e_9, t_9), (e_8, t_8)]$, and $[(e_1, t_1), (e_9, t_9), (e_8, t_8), (e_5, t_5)]$. Each subpath is associated with a probability P_i.

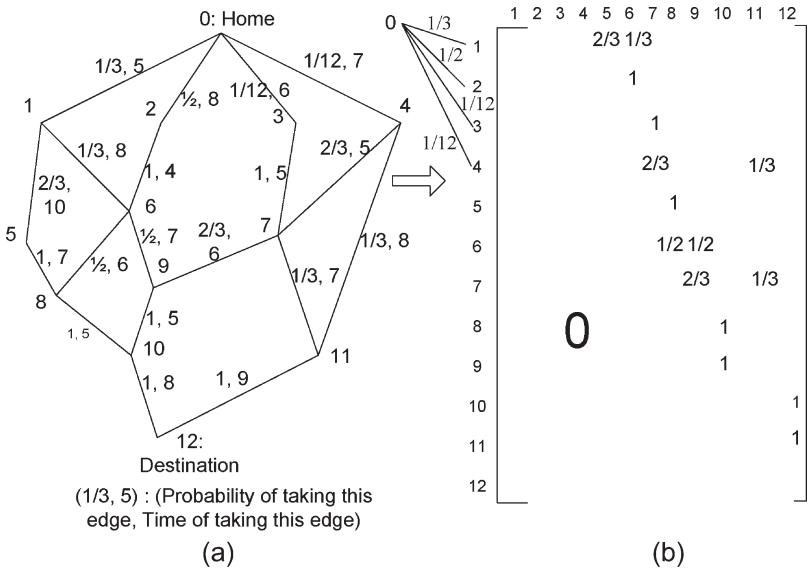

Fig. 5. Path-probability matrix. (a) Example network model. (b) Path-probability matrix.

Mismatched Location	Similarity Value
0	10
1	7
2	6
3	7
4	10
No Mismatch	15

Fig. 6. Example of similarity values.

a) Similarity Measure Between the Test String of Length l and the Normal Profile: Given a test string X of length l (X_l), we first compute its distance to N in the following two steps.

1) For an $R \in N$ of an arbitrary length $L(L > l)$, we first compute the similarity between X_l and R

$$\mathrm{Sim}_R(X_l) = \max\{P_i * \mathrm{Sim}(X_l, R_i)\}, \qquad \forall R_i \text{ of length } l$$
$$R_i \in R, \qquad i = 1, 2, \ldots, L - l + 1$$

where P_i is the path probability of R_i.

2) We then compute the similarity between X_l and N

$$\mathrm{Sim}_N(X_l) = \max\{\mathrm{Sim}_R(X_l)\}, \qquad \forall R \in N.$$

b) Similarity Measure Between the Test String of Arbitrary Length and the Normal Profile: In order to compute the similarity between a test string X of arbitrary length D and the normal profile N, we first break X into l-length substrings $(X_i, X_{i+1}, \ldots, X_{i+l-1})$, $i = 1, 2, \ldots, D - l + 1$. That is, we slide a window of length l over the test trace, each time by one position. In a thin way, the original test string X is broken into $(D - l + 1)$ l-length substrings. We calculate all $\mathrm{Sim}_N(X_i)$ and compute their average

$$\mathrm{Sim}(X) = \frac{\sum_{i=1}^{D-l+1} \mathrm{Sim}_N(X_i)}{(D - l + 1)}. \tag{6}$$

$\mathrm{Sim}(X)$ will be used as the final similarity between a test string of arbitrary length and the normal profile N. A threshold mechanism is then used to decide whether X is normal or not

$$X = \begin{cases} \text{Normal}, & \text{if } \mathrm{Sim}(X) \geq t \\ \text{Abnormal}, & \text{if } \mathrm{Sim}(X) < t. \end{cases}$$

Here, t is the threshold, a system parameter that needs to be tuned in the design phase.

E. Implementation Issues

Given a training set T, which is the collection of all training examples, suppose that the number of vertices [vertices as shown in Fig. 5(a)] is m and that the number of training instances is n. In order to construct the path-probability matrix as shown in Fig. 5(b), we need $O(n^2)$ storage and $O(mn)$ time in the training phase. When n is large, a large data storage is needed. In addition, in the detection phase, based on the defined similarity measure, only a single historical sequence is selected as most similar to the test sequence. A large n will incur heavy search and comparison operations.

Different strategies can relieve these situations in an operational setting. For example, given the relatively stable user's behaviors, we can apply the principle of locality of reference. Suppose that a linked list is used to store the path probabilities shown in Fig. 5(b), we can organize the vertices based on the decreasing order of the route probability. Suppose that the collection of routes coming out of a vertex v is $\{v \to v_1, v \to v_2, \ldots, v \to v_n\}$ and that the corresponding route probabilities are $\{p_1, p_2, \ldots, p_n\}$, then we can sort $\{v_1, v_2, \ldots, v_n\}$ based

on the decreasing order of $\{p_1, p_2, \ldots, p_n\}$. In this way, search operations can be effectively reduced.

In addition, if the probability of a corresponding route is very small and less than a predefined threshold, we can prune this edge from the storage. A proper threshold value can have important impacts on the system performance. If the threshold is small, the effectiveness of the reduction storage is small. If the threshold is large, the classification performance is impacted. This tradeoff makes the selection of the threshold a site-dependent issue.

VII. Simulation

A. Performance Metrics

We use the following two metrics to evaluate the performance of our proposed detection algorithms.

1) False-positive ratio: For the calling activities, false-positive ratio is measured over normal calls. For the mobility activities, false-positive ratio is measured over normal itineraries. Let m denote the number of measured normal calls (or normal itineraries), and n of them are identified as abnormal. False-positive ratio is defined as n/m.
2) Detection ratio: For the calling activities, detection ratio is measured over abnormal calls. For the mobility activities, detection ratio is measured over abnormal itineraries. Let m denote the number of measured abnormal calls (or abnormal itineraries), and n of them are detected. Detection ratio is defined as n/m.

False-positive and detection ratios are the two most popular metrics in measuring the performance of IDSs. In threshold-based anomaly detection mechanisms, with the adjustment of the threshold, false-positive and detection ratios tend to decrease or increase at the same time. This reflects the tuning process in IDSs. In this respect, receiver-operating-characteristic (ROC) curves can plot the tradeoff between false-positive and detection ratios. In our simulation, we introduce the concept of degree of anomaly, which adds one more factor in measuring the performance. To make the results better plotted, we do not adopt the ROC curves.

B. Data Sets

1) Data Sets of Calling Activities: For the training data regarding a user's calling activities, we assume that the CDT follows the gamma distributions because they reflect the emerging services and are more flexible than the exponential distribution [14]. We also assume that the CIP follows the gamma distributions. Gamma distribution has the pdfs $\gamma = f(x|a,b) = (1/b^a\Gamma(a))x^{(a-1)}e^{x/b}$. Here, Γ is the gamma function defined by the integral $\Gamma(a) = \int_0^\infty e^{-t}t^{a-1}dt$, where a is the shape parameter. b is the scale parameter.

One of the essential assumptions in the whole intrusion detection research is that the normal and the malicious activities should demonstrate distinct behaviors. In our context, we use D to denote the difference between the normal and the abnormal behaviors. Specifically, we use the difference between the location parameter of the normal profile's gamma distribution and the location parameter of the abnormal profile's gamma distribution to represent D. Let μ_n denote the location parameter of the normal profile's gamma distribution and μ_a denote the location parameter of the abnormal profile's gamma distribution. Then, we have $D = |\mu_n - \mu_a|$.

Intuitively, the larger the D is, the more distinct the normal and abnormal profiles are and the better performance the detection algorithm can achieve. It is worth noting that when D is too small, the distinction between the normal and the abnormal profiles is so small that it is very difficult to tell them apart. In this simulation, we generate normal and abnormal profiles using different D values as the training data.

Given a normal and an abnormal profile at a given D, we also generate test data to test the performance of our detection algorithm. It is very normal that user's behaviors may demonstrate some variations. Therefore, we use the difference between the training data's location parameter and the test data's location parameter to represent the users' behavior variation. Let μ_{train} denote the location parameter of the gamma distribution of the training data and μ_{test} denote the location parameter of the gamma distribution of the test data. In order to generate the user test data whose distance is d from the training data, we set $d = |\mu_{\text{train}} - \mu_{\text{test}}|$.

2) Data Sets of Mobility Activities: We use the example network model shown in Fig. 5(a) to generate normal users' behaviors. To test the performance of our proposed schemes, we first introduce a concept—the degree of anomaly—which reflects the degree of variation of a user's mobility profile. In our context, the degree of anomaly could be defined as the percentage of the number of new vertices (vertices not existing in the normal profile) over the number of vertices in the normal profile. In the following simulations' results, we use the number of new vertices to represent the degree of anomaly.

We gradually increase the number of new vertices not included in Fig. 5(a) in order to generate a wide variety of test data. The more the number of new vertices not included in Fig. 5(a), the more abnormal the user's mobility pattern is. For each degree of anomaly, we use the corresponding set of vertices to randomly generate a path set as the test data, which corresponds to the user's behavior at this degree of anomaly. These test data are then used to measure the false-positive and detection rates.

C. Bayes-Based Detection Algorithms

1) A Posteriori Probability Distance: Given an observation $\mathbf{x}$, we first measure the distance of its *a posteriori* probability under different classes following (1). Because $p(\mathbf{x})$ is unimportant as far as making a decision is concerned, we measure the distance as $|p(\mathbf{x}|\omega_1)P(\omega_1) - p(\mathbf{x}|\omega_2)P(\omega_2)|$, as shown in Fig. 7.

First, when d is small, the distance decreases with the increase of d. This is because, when d increases, the test data are closer to abnormal data. This leads to the decreases of the distance, as shown in Fig. 7. This also leads to the changes of false-positive and detection ratios, as shown in Figs. 8 and 9.

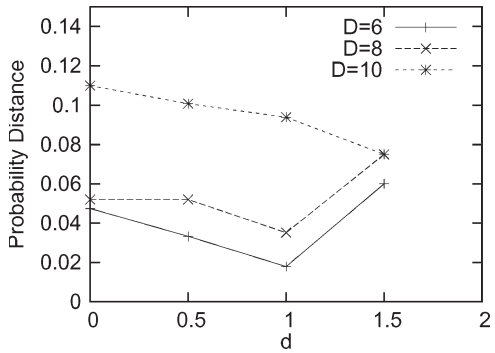

Fig. 7. *A posteriori* probability distance.

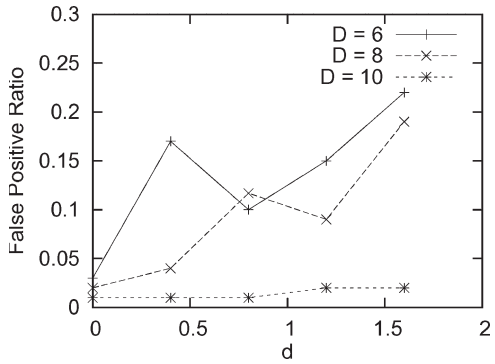

Fig. 8. False-positive ratio at different calling activities—Bayesian decision rule.

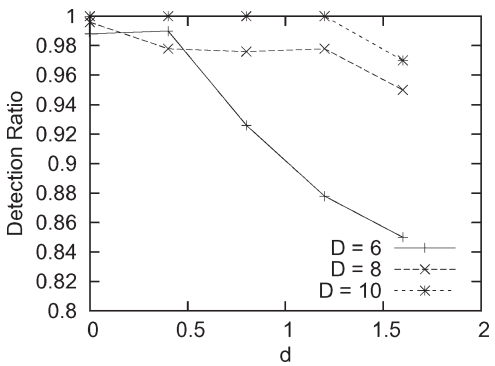

Fig. 9. Detection ratio at different calling activities—Bayesian decision rule.

Second, when D is small and when d increases to some value, the distance increases. This is because we measure the absolute value. When d is large enough, the test data become closer to the abnormal data. Therefore, the distance decreases.

Third, we can also observe that if d stays the same and D increases, the distance increases. This is also what we expect. A larger D means a clearer separation between the normal and abnormal behaviors in the training data. This contributes to the increase of the distance.

2) False-Positive Ratio: Simulation results of the false-positive ratio of the Bayesian detection algorithm are shown in Fig. 8. We have the following observations. First, when the normal and abnormal behaviors have a large distance (for example, $D = 10$), the false-positive ratio is very low, even if the user demonstrates a relatively large variation. This will lead to a very small false-positive ratio.

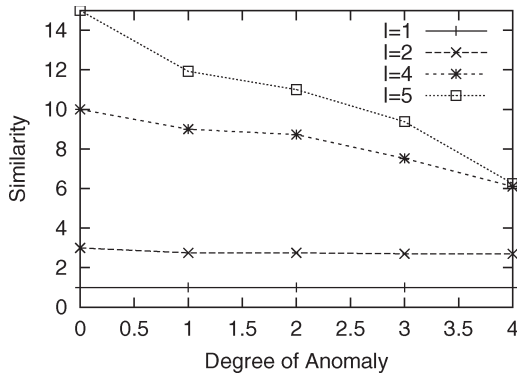

Fig. 10. Similarity value of test string to normal profile.

Second, given the same D, with the increase of d, we have an increasing false-positive ratio. This is what we have expected. With the increasing d, the user's behaviors tend to demonstrate a large deviation. It is very normal that false-positive ratio increases.

Third, given the same d, with the decrease of D, the false-positive ratio increases. This is because, with the decrease of the D, the users' normal and abnormal behaviors tend to have more overlap. This leads to an increase of the false-positive ratio.

3) Detection Ratio: Simulation results of the detection ratio of our algorithm are shown in Fig. 9. We have the following observations. First, when the normal and abnormal behaviors have a large distance (for example, $D = 10$), the detection ratio is very high, even if the user demonstrates a relatively large variation. This leads to a very high detection ratio.

Second, given the same D, with the increase of d, we have a decreasing detection ratio. This is what we have expected. With the increasing d, the users' behavior tends to demonstrate a large deviation. It is very normal that the detection ratio decreases.

Third, given the same d, with the decrease of D, the detection ratio decreases. This is because, with the decrease of the D, the users' normal and abnormal behaviors tend to have more overlap. This leads to a decrease of the detection ratio.

D. IBL-Based Detection Algorithms

1) Similarity: Given a test string X, we compute its similarity value to the normal profile based on (6). This gives us ideas about how the similarity value changes at a different degree of anomaly. The result is shown in Fig. 10.

First, when l is large, with the increase of the degree of anomaly, the similarity value decreases. This is because when l is large, it is easier to have a character mismatch when the degree of anomaly becomes larger.

Second, when l is small, we do not observe much difference of similarity values. This demonstrates the impact of l on the similarity value of the test string. We can also see that, if we fix the degree of anomaly, with the increase of l, the similarity value increases.

2) False-Positive and Detection Ratios: Simulation results of the false-positive and detection ratios of the IBL-based detection algorithms are shown in Figs. 11 and 12, respectively. Let l represent the length of consecutive edges.

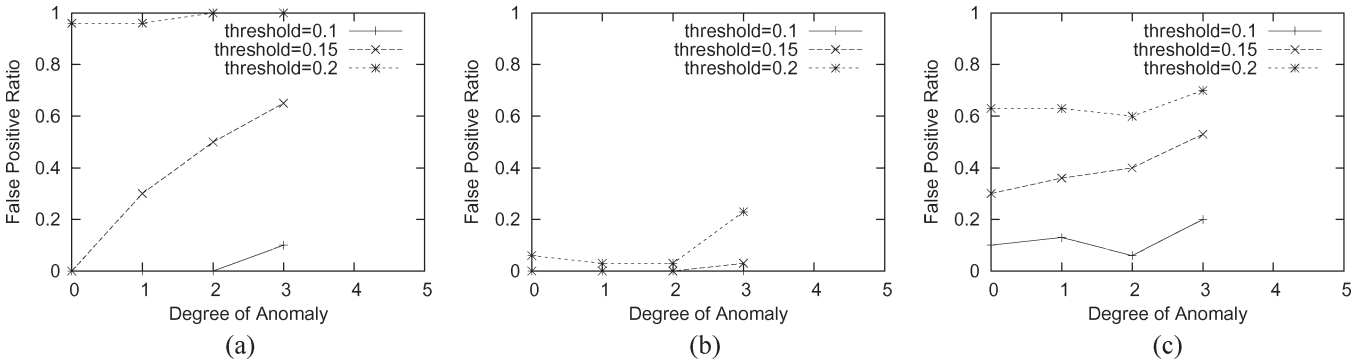

Fig. 11. False-positive ratio at different mobility activities—IBL. (a) $1 = 1$. (b) $1 = 4$. (c) $1 = 6$.

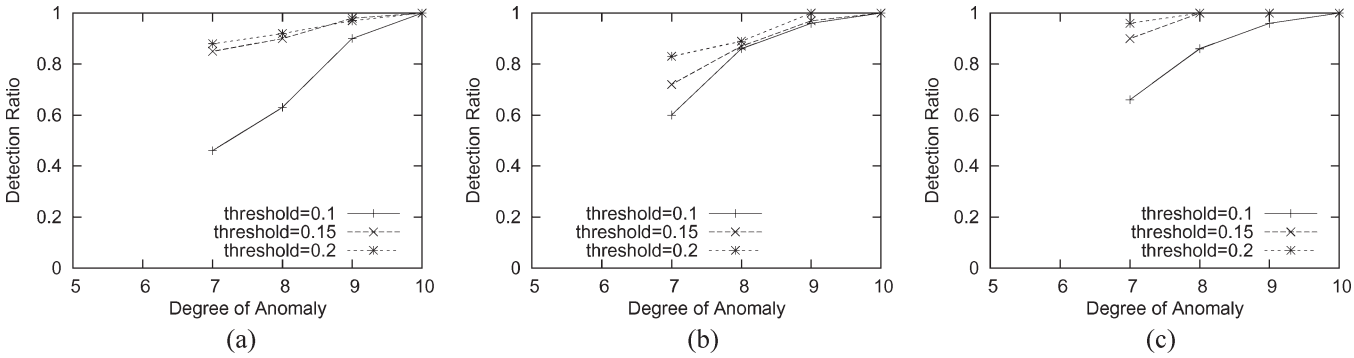

Fig. 12. Detection ratio at different mobility activities—IBL. (a) $1 = 1$. (b) $1 = 4$. (c) $1 = 6$.

First, if l is too small (for example, $l = 1$), considering a reasonable false-positive ratio, detection ratio will be too low. A small l does not take into consideration the sequential feature of users' movement patterns. This will lead to a low detection ratio.

Second, if l is too large (for example, $l = 6$), considering a reasonable detection ratio, false-positive ratio will be too high. This is because the path probability will be small given a very large path length l, thus decreasing the similarity of the test path. This leads to the increase of the false-positive ratio.

Third, given a reasonable l (for example, $l = 4$) and a suitable threshold, we can achieve a reasonable tradeoff between the false-positive and detection ratios. With the increase of the threshold, we observe an increase of the false-positive and detection ratios. This is a general trend we also observe when l is one and six.

VIII. Conclusion and Future Work

In this paper, motivated by the observations that most users demonstrate certain regularities in their activities, we aim at constructing an end user's profile for anomaly detection in wireless networks. To utilize users' calling activities, we formulate the intrusion detection problem as a multifeature two-class pattern-classification problem and apply the Bayesian decision rule to the collected data. To exploit movement patterns demonstrated by mobile users, we first propose a realistic network model integrating geographic road-level granularities. We then apply an IBL technique to construct mobile users' movement patterns. We also perform simulations to evaluate the effectiveness of these two algorithms. Simulation results demonstrate that under certain properly tuned parameters, both algorithms can achieve desirable performance.

It is obvious that the end user's behavior is very complex and changing all the time. We plan to consider more features in the future to make our system more general and robust. We also plan to further revise our algorithms to address the problem incurred by complex users' behaviors. One more important work is to test the proposed algorithms using real-world data.

References

[1] B. Sun, F. Yu, K. Wu, Y. Xiao, and V. C. M. Leung, "Enhancing security using mobility-based anomaly detection in cellular mobile networks," *IEEE Trans. Veh. Technol.*, vol. 55, no. 4, pp. 1385–1396, Jul. 2006.

[2] Y.-B. Lin, M. Chen, and H. Rao, "Potential fraudulent usage in mobile telecommunications networks," *IEEE Trans. Mobile Comput.*, vol. 1, no. 2, pp. 123–131, Apr.–Jun. 2002.

[3] M. Zhang and Y. Fang, "Security analysis and enhancements of 3GPP authentication and key agreement protocol," *IEEE Trans. Wireless Commun.*, vol. 4, no. 2, pp. 734–742, Mar. 2005.

[4] H. Debar, M. Dacier, and A. Wespi, "A revised taxonomy for intrusion-detection systems," *Ann. Telecommun.*, vol. 55, no. 7/8, pp. 361–378, 2000.

[5] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 7, pp. 222–232, Feb. 1987.

[6] B. Sun, F. Yu, K. Wu, and V. C. M. Leung, "Mobility-based anomaly detection in cellular mobile networks," in *Proc. ACM WiSe Conjunction ACM Mobicom*, Philadelphia, PA, 2004, pp. 61–69.

[7] U. Lindqvist and P. A. Porras, "Detecting computer and network misuse through the production-based expert system toolset (P-BEST)," in *Proc. IEEE Symp. Security Privacy*, Oakland, CA, May 1999, pp. 146–161.

[8] S. Kumar and E. Spafford, "A pattern matching model for misuse intrusion detection," in *Proc. 17th Nat. Comput. Security Conf.*, Oct. 1994, pp. 11–21.

[9] K. Ilgun, "USTAT: A real-time intrusion detection system for unix," in *Proc. IEEE Symp. Res. Security Privacy*, Oakland, CA, May 1993, pp. 16–28.

[10] H. Debar, M. Becker, and D. Siboni, "A neural network component for an intrusion detection system," in *Proc. IEEE Symp. Res. Security Privacy*, Oakland, CA, May 1992, pp. 240–250.
[11] J. Hall, M. Barbeau, and E. Kranakis, "Anomaly-based intrusion detection using mobility profiles of public transportation users," in *Proc. IEEE Int. Conf. Wireless Mobile Comput., Netw. Commun.*, Montreal, QC, Canada, Aug. 22–24, 2005, pp. 17–24.
[12] D. Samfat and R. Molva, "IDAMN: An intrusion detection architecture for mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 7, pp. 1373–1380, Sep. 1997.
[13] H. A. Karimi and X. Liu, "A predictive location model for location-based services," in *Proc. 11th ACM Int. Symp. Advances Geographic Inf. Syst.*, New Orleans, LA, 2003, pp. 126–133.
[14] Y. Fang, I. Chlamtac, and Y. Lin, "Modeling PCS networks under general call holding time and cell residence time distributions," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 893–906, Dec. 1997.
[15] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ: Wiley, Oct. 2000.
[16] S. Rosset, U. Murad, E. Neumann, Y. Idan, and G. Pinkas, "Discovery of fraud rules for telecommunications challenges and solutions," in *Proc. ACM SIGKDD*, San Diego, CA, 1999, pp. 409–413.
[17] D. W. Abbott, I. P. Matkovsky, and J. F. Elder, "An evaluation of high-end data mining tools for fraud detection," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, San Diego, CA, 1998, pp. 2836–2841.
[18] Y. Moreau, H. Verrelst, and J. Vandewalle, "Detection of mobile phone fraud using supervised neural networks: A first prototype," in *Proc. Int. Conf. Artif. Neural Netw.*, Lausanne, Switzerland, 1997, pp. 1065–1070.
[19] P. Burge and J. Shawe-Taylor, "Detecting cellular fraud using adaptive prototypes," in *Proc. AAAI Workshop AI Approaches Fraud Detection Risk Manage.*, Providence, RI, 1997, pp. 9–13.
[20] R. Büschkes, D. Kesdogan, and P. Reichl, "How to increase security in mobile networks by anomaly detection," in *Proc. ACSAC*, Scottsdale, AZ, 1998, pp. 3–12.
[21] G. Gu, M. Sharif, X. Qin, D. Dagon, W. Lee, and G. Riley, "Worm detection, early warning and response based on local victim information," in *Proc. ACSAC*, Tucson, AZ, Dec. 6–10, 2004, pp. 136–145.
[22] C. Kruegel and G. Vigna, "Anomaly detection of web-based attacks," in *Proc. ACM CCS*, Washington, DC, 2003, pp. 251–261.
[23] Q. He, D. Wu, and P. Khosla, "Quest for personal control over mobile location privacy," *IEEE Commun. Mag.*, vol. 42, no. 5, pp. 130–136, May 2004.
[24] S. Schonfelder, "Some notes on space, location and travel behaviour," in *Proc. Swiss Transp. Res. Conf.*, 2001.
[25] K. Choi and W. Jang, "Development of a transit network from a street map database with spatial analysis and dynamic segmentation," *Transp. Res. Part C Emerg. Technol.*, vol. 8, no. 1, pp. 129–146, Jan. 2000.
[26] T. Lane and C. E. Brodley, "Temporal sequence learning and data reduction for anomaly detection," *ACM Trans. Inf. Syst. Security*, vol. 2, no. 3, pp. 295–331, Aug. 1999.
[27] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

Bo Sun (S'01–M'04) received the Ph.D. degree in computer science from Texas A&M University, College Station, in 2004.

He is now an Assistant Professor with the Department of Computer Science, Lamar University, Beaumont, TX. His research interests include security issues (intrusion detection in particular) of wireless *ad hoc* networks, wireless sensor networks, cellular mobile networks, and other communications systems.

Yang Xiao (SM'04) received the Ph.D. degree in computer science and engineering from Wright State University, Dayton, OH.

He worked in the industry as a Medium Access Control Architect, involved with the IEEE 802.11 standard enhancement work. He joined the Department of Computer Science, University of Memphis, Memphis, TN, in 2002. He was a Voting Member of the IEEE 802.11 Working Group from 2001 to 2004. He is currently with the Department of Computer Science, University of Alabama, Tuscaloosa. He has published more than 200 papers in major journals (more than 50 in various IEEE journals/magazines), refereed conference proceedings, and book chapters. His research has been supported by the US NSF. He currently serves as an Editor-in-Chief of the *International Journal of Security and Networks*, the *International Journal of Sensor Networks*, and the *International Journal of Telemedicine and Applications*. He serves on the editorial boards or is an associate editor of several journals such as the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, etc. His research interests include security, telemedicine, sensor networks, and wireless networks.

Dr. Xiao is a member of the American Telemedicine Association. He serves as a Referee/Reviewer for many funding agencies, a Panelist for the US National Science Foundation (NSF), and a member of the Canada Foundation for Innovation Telecommunications Expert Committee. He serves as the Technical Program Chair of more than 100 conferences such as the IEEE International Conference on Computer Communications, the International Conference on Distributed Computing Systems, the ACM International Symposium on Mobile *Ad Hoc* Networking and Computing, the IEEE International Conference on Communications, the IEEE Global Communications Conference, the IEEE Wireless Communications and Networking Conference, etc.

Ruhai Wang (M'03) received the Ph.D. degree in electrical engineering from New Mexico State University, Las Cruces, in 2001.

He is currently an Assistant Professor with the Department of Electrical Engineering, Lamar University, Beaumont, TX. His research interests include computer networks and communication systems with emphases on wireless communications, wireless and space Internet, network protocols and security, and performance analysis. He has published about 50 papers in international journals and conference proceedings on these topics.

Dr. Wang currently serves as a member of the editorial board of the Wiley *Wireless Communications and Mobile Computing Journal* and a Guest Editor for a few other international journals. He has also served as a Technical Program Committee (TPC) Chair and as a member of many major international conferences in computer networks and communication systems, including TPC Co-chair for the IEEE ICC'07 Wireless Communications Symposium.