Enhancing Security Using Mobility-Based Anomaly Detection in Cellular Mobile Networks

Bo Sun, *Member, IEEE*, Fei Yu, *Member, IEEE*, Kui Wu, *Member, IEEE*, Yang Xiao, *Senior Member, IEEE*, and Victor C. M. Leung, *Fellow, IEEE*

***Abstract*—Location information is an important feature in users' profiles in cellular mobile networks. In this paper, by exploiting the location history traversed by a mobile user, two domain-independent online anomaly detection schemes are designed, namely the Lempel–Ziv (LZ)-based and Markov-based detection schemes. The authors focus on the identification of a group of especially harmful internal attackers—masqueraders. For both schemes, cell IDs traversed by each mobile user are extracted as the feature value. Specifically, the mobility pattern of each user is characterized by a high-order Markov model. The LZ-based detection scheme from the well-developed data compression techniques is derived. Moreover, the technique of exponentially weighted moving average is used to modify a user's normal profile dynamically. The user profile can characterize the normal behavior of each user accurately and is sensitive to abnormal changes. For the Markov-based detection scheme, a fixed-order Markov model is used to characterize the normal behavior. Based on the constructed probability transition matrix, the probability of the user's current activity is calculated. A threshold policy is then used in both schemes to determine whether a mobile device is potentially compromised or not. Simulation results are presented to show the effectiveness of the proposed schemes. Moreover, our results show that the LZ-based detection scheme performs better than the Markov-based detection scheme, especially for low-speed mobile users.**

***Index Terms*—Anomaly detection, cellular mobile networks, mobility.**

I. Introduction

IN RECENT years, the rapid development of cellular mobile data services has made people increasingly rely on cellular phones in their daily lives for important and sensitive tasks such as E-shopping and E-banking. The booming new services, while bringing great convenience, have caused serious security concerns. Although there are many authentication protocols in cellular mobile networks, designing a highly secure cellular mobile network is still a very challenging issue due to the open radio transmission environment and the physical vulnerability of mobile devices.

Manuscript received May 30, 2005; revised August 26, 2005 and September 26, 2005. The review of this paper was coordinated by Prof. Y.-B. Lin.

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

F. Yu and V. C. M. Leung are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: feiy@ece.ubc.ca; vleung@ece.ubc.ca).

K. Wu is with the Department of Computer Science, University of Victoria, Victoria, BC V8W 3P6, Canada (e-mail: wkui@cs.uvic.ca).

Y. Xiao is with the Department of Computer Science, University of Memphis, Memphis, TN 38152 USA (e-mail: yangxiao@ieee.org).

Digital Object Identifier 10.1109/TVT.2006.874579

In general, two complementary classes of approaches exist to protect a system, namely 1) prevention-based approaches and 2) detection-based approaches. Prevention-based techniques, such as authentication and encryption, can effectively reduce attacks by keeping illegitimate users from entering the system. They are usually based on some symmetric or asymmetric mechanisms to ensure that users conform to predefined security policies. However, the experience in security of wired networks indicates the necessity of multilayer and multilevel protections because there are always some weak points in the system that are hard to predict. This is especially true for mobile networks, given the low physical security of mobile devices. The attackers could utilize various techniques to crack the secrets embedded in mobile devices. Currently, tamper-resistant hardware and software are still expensive or unrealistic for mobile devices. Therefore, if a device is compromised, all the secrets associated with the device become open to the attackers, rendering all prevention-based techniques helpless and resulting in great damage to the whole system. To solve this problem, intrusion detection systems (IDSs), which serve as the second wall of protection, could effectively help identify malicious activities.

Generally, there are two intrusion detection techniques, namely 1) misuse-based detection and 2) anomaly-based detection [2]. A misuse-based detection technique encodes the known attack signatures and system vulnerabilities and store them in a database. The system monitors the current subject activities. If it finds a match between the current activity and the signature, an alarm is generated. Misuse detection techniques are not effective to detect novel attacks because of the lack of corresponding signatures. An anomaly-based detection technique creates normal profiles of system states or user behaviors and compares them with the current activities. If a significant deviation is observed, the system raises an alarm. Anomaly detection can detect unknown attacks. However, the normal profiles are usually very difficult to build for cellular mobile networks due to the mobility of end users. Therefore, establishing normal profiles of mobile users is crucial in designing an efficient intrusion detection scheme in cellular mobile networks.

Lin *et al.* [1] proposed an excellent study to detect the potential fraudulent usage of cloned phones in cellular mobile networks. In this paper, we propose two different approaches to establish normal profiles of mobile users, from which efficient intrusion detection schemes are designed. Our work is based on such a common observation: 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/she could not follow the movement pattern of the authentic owner. That is, the masquerader tends to have different routines. By establishing an accurate normal profile that can reflect the normal movement pattern and comparing it with the current mobility pattern, we could thus effectively identify misbehaviors.

Both of our approaches are based on recent advances in mobility prediction techniques in cellular mobile networks [3]–[6]. It is shown that mobility prediction can significantly improve the performance of mobility management [3], QoS provisioning [4], and resource management [5] in cellular mobile networks. However, to the best of our knowledge, the design of efficient schemes to detect misbehaviors based on mobility prediction techniques has not been addressed in previous work, which is the gap that we want to fill in this paper.

We use the cell list traversed by a mobile user during his/her call as the feature value in intrusion detection. Our first intrusion detection scheme is based on Lempel–Ziv (LZ) data compression techniques [7], which are both theoretically optimal and good in practice. It has been demonstrated that data compression is synonymous with prediction [8]. A mobility trie, or a multiway tree, is constructed online to record the cells efficiently. We then use the mobility trie to derive a probabilistic model of the user's mobility pattern. The second proposed scheme is based on Markov prediction techniques. We apply a fixed-order Markov model to the extracted cell string and compute the transition probability of the next location given the previous locations. A threshold policy is applied to both schemes to decide whether the current activities are abnormal or not. Simulation results are presented to show the effectiveness of the proposed schemes.

II. Related Work

This paper has two important aspects of related work, namely 1) intrusion detection and 2) location prediction.

Furthermore, there are two important intrusion detection techniques, namely 1) misuse detection and 2) anomaly detection. A good taxonomy of existing technologies is presented in [2]. The research of intrusion detection began with Denning's seminal paper [14]. Since then, many research efforts have been devoted to different detection techniques. For example, expert system [15], colored petri nets [16], and state transition analysis [17] have been used to construct misuse-based detection techniques. Different statistical approaches [15] and neural networks [18] have been used to construct anomaly-based detection techniques. All existing approaches take into consideration domain specific knowledge to build suitable detection systems.

Compared to the research on intrusion detection for wired networks, relatively few research efforts have been devoted to intrusion detection research of wireless networks. In [19], Samfat and Molva proposed the intrusion detection architecture for mobile networks (IDAMN), which includes two algorithms to model the behavior of users in terms of both telephony activity and migration patterns. Lin *et al.* [1] proposed an excellent study to detect the potential fraudulent usage of cloned phones in cellular mobile networks. In [20], Büschkes *et al.* applied the Bayes decision rule to user's mobility patterns to increase the security in mobile networks.

Location prediction schemes have been proposed in many other research areas of wireless cellular networks. It could be used to improve the performance of mobility management [3] and call admission control [4]. In [13], Song *et al.* used extensive real Wi-Fi data to evaluate LZ-based and Markov-based location predictors. A recent survey of location predictors is presented in [6].

III. Motivations

For most mobile users in cellular networks, movement patterns can be captured and modeled. We can learn the pattern from the mobility history of an authentic user and identify intruders by comparing the current mobility information with the normal movement pattern. Nevertheless, there are a certain number of users such as taxi drivers who do not exhibit regular movement patterns. It will be very hard, if not impossible, to model those users' movement patterns. In addition, it is normal for people to change their normal routines occasionally. For example, people on vacation may exhibit significant deviation from their normal movement patterns, and different vacation routines will lead to very rare events. All these factors may result in the inaccuracy of the established normal profile. Therefore, we should not expect that our detection based on mobility patterns is accurate for all users in all situations.

Based on these considerations, our research is not motivated to build a system to accurately detect all intrusions. Instead, we are aimed at providing an optional service to end users as well as a useful administration tool to service providers. The attacker can cause a huge loss for the authentic owner if the compromised cellular phone is not identified in time. Because of this reason, the real owner might need some warning information via other channels (e.g., email, phone call to home) if the system observes some abnormal behavior. Such warning could be something like, "We observe that you are having a significant change of movement patterns. Is your handheld still safe?" We believe that such an optional service will be popular given the increasing number of security related incidents of wireless networks. For the service provider, the system can build a "gray list" to include the users who exhibit dramatic changes of movement patterns. The traffic patterns and behaviors of the users in the "gray list" need to be monitored with more cautions. As long as they try to issue some dangerous commands to the network, immediate response is required to avoid potential financial loss. The "gray list" should be updated dynamically. For instance, a person who leaves on holidays may be added into the "gray list" but will be removed when he resumes his normal routines. The similar strategy has been used in credit card companies. For example, a customer will be called if the abnormal usage of his/her credit card is detected, such as the card being used at another country that is not the owner's residence and the owner frequently visits.

Our proposed approach requires the tracking of people's locations. It is a location tracking service that is based on the system tracking users' locations, which are, in our case, the cell list traversed by each user. This will give rise to user's location

privacy issues. Therefore, our system provides the user with an option to turn off this service or not. 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, for example, [9]. Therefore, it is promising to integrate our proposed service with other existing location privacy protection schemes.

IV. Assumptions

We have the following assumptions in designing mobility-based anomaly detection schemes.

First, we assume that there is a mobility database for each mobile user that describes his normal activities. This is a reasonable assumption in cellular mobile networks because this mobility database could be constructed by location tracking and prediction services. This mobility database could be stored together with the mobile user's personal information, such as billing information, in the home location register (HLR). Note that in realistic networks, the locations of mobile users are actually tracked for the purpose of service provision and smooth handoff, even though the end users may be unaware of such monitoring. We assume that HLR is secure and the movement information is accurate. In most cases, because of its importance, HLR is protected with highly secure measures, and thus, it is extremely hard to be compromised. Furthermore, the update and registration of the location is usually based on the device's current serving cell and the hardware registration such as the serial number of SIM card. Therefore, it will be hard for the attacker to hide or fabricate his location even if he has compromised all the secrets of the mobile device. Even if an attacker finds some magical way to fabricate his location, he still has no idea about the normal movement profile of the real device owner.

Second, we assume that mobile devices can be compromised and 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, because all prevention techniques will be rendered helpless once the mobile device is captured and compromised. Actually, if we could assume the tamper resistance of hardware/software, the whole security research could become much easier.

Third, we assume that most users have favorite or regular itineraries. This makes it viable for us to establish each user's normal profile. This assumption is reasonable given most users' regular daily lives. Actually, all research on intrusion detection is based on two assumptions, namely 1) the subject activity is observable via some system auditing mechanisms and 2) the normal and malicious activities should demonstrate distinct behavior. Therefore, it is possible to reason about the evidence in the data to determine whether the system is currently under attack. If a user has totally random behaviors, for example, the movement of a taxi driver, it will be very difficult, if not impossible, to create his normal movement profile. Our mobility-based detection algorithm alone is not suitable for such kind of users. Nevertheless, our method is automatically user selective, because the optional warning service mentioned in the previous section will tend to give many false warning messages to this type of users and force them to unsubscribe such a service.

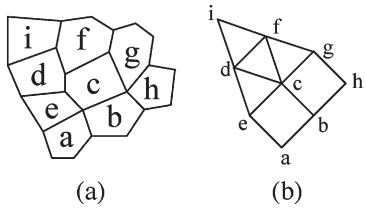

Fig. 1. Example cellular network and its graph model. (a) Example of cellular network with cells. (b) Graph model of the cellular network.

V. Model Description

A. Threat Model

The complex cellular mobile network system could incur software errors and design errors. This could make many attacks possible. One example is cell phone cloning [1]: The mobile phone card of an authenticate user A is cloned by some attacker B, which enables B to use the cloned phone card to make fraudulent telephone calls. If cell phone cloning happens, the bills for the calls will go to the legitimate subscriber. In addition, the masquerader could fake the international mobile equipment identifier (IMEI) and the subscriber identity module (SIM) card to get the service illegally. Subscription fraud [1] could also enable the intruder to subscribe the service using the authentic user's name.

All these enable the necessity of a fraud detection system that can complement existing intrusion prevention systems for cellular mobile networks. By comparing the different behaviors demonstrated by the authentic user and the attacker, the system can detect the potential misbehavior.

B. Network Model

Most of the previous work on wireless cellular networks uses structured graph network topology models, such as hexagonal or square cell configurations. However, these models may not accurately represent a cellular network in practice, where the cell shape and size may vary depending on the antenna radiation pattern and propagation environment. In wireless cellular networks, each cell usually has a base station to serve it. Therefore, in our system, the wireless cellular network is modeled as a generalized graph $G = (V, E)$. The vertex set V represents all the base stations. If two cells are adjacent to each other, there is an edge between their two vertices.

An example of the model is illustrated in Fig. 1(a) and (b). In this example, the vertex set is $V = \{a, b, c, d, e, f, g, h, i\}$, and the edge set is $E = \{(a, b), (b, c), \ldots (f, i)\}$.

C. Mobility Model

The random walk model has been widely used in the literature, in which a mobile user will move to any one of the neighboring cells with equal probability after leaving a cell. This may not be realistic in practice, because mobile users normally travel with a destination in mind. Therefore, we use the

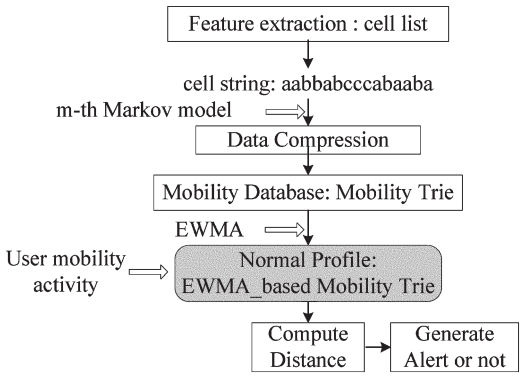

Fig. 2. LZ-based anomaly detection.

mth-order Markov model in this paper. In such a model, the mobility of a user can be represented by a sequence of characters, $C_1, C_2, C_3, \ldots, C_i, \ldots$, where C_i denotes the identity of the cell visited by the mobile. Because the future locations of the mobile are likely to be correlated with its movement history, the sequence of characters $C_1, C_2, C_3, \ldots, C_i, \ldots$ is assumed to be generated by a mth-order Markov source, where the states correspond to the context of the previous m characters. The probability that the user moves to a particular cell depends on the location of the current cell and a list of cells recently visited.

VI. Mobility-Based Anomaly Detection Schemes

In this section, we present two mobility-based anomaly detection schemes called LZ-based scheme and Markov-based scheme, both relying on extraction of the cell list traversed by each user as the feature value.

In the LZ-based scheme, based on users' regular itineraries, a mobility trie is constructed from the accumulative history of users' movement patterns. The recent normal profile of a user is built by applying the exponentially weighted moving average (EWMA) [11] technique to the mobility trie. In other words, this modified mobility trie serves as the normal profile of the user in the recent past and reflects the stationary part of the user's regular mobility pattern. Based on this, we use a blending scheme, which is introduced in Section VI-B2, to calculate the probability of each user's activity. Fig. 2 illustrates the LZ-based scheme.

Note that similar approaches have been used in different applications such as in [3] to solve the location management problem, in [4] to solve the call admission control and bandwidth reservation problem, and in [13] to evaluate location predictors with extensive real Wi-Fi mobility data. However, it is the first time that this approach is used for intrusion detection. Furthermore, those constructed mobility databases in [3], [4], and [13] cannot be used directly and effectively because recent activities of users are not taken into consideration. In this paper, recent activities of users are taken into consideration and a new similarity measure, which is introduced in Section VI-B3b, is derived to provide criteria to evaluate the normalcy of the user's itineraries.

The second scheme, which is the Markov-based scheme, is based on order-o Markov predictors. The Markov-based approach is one of the most commonly used approaches in making predictions. We adopt it mainly to compare our LZ-based approach. For the Markov-based approach, given an order o, the probability of being the next cell given the previous o cells is constructed. In other words, the probability of the future activity can be calculated.

Both the LZ-based and the Markov-based schemes are online predictors, which means that they examine the history so far, extract the current context, and predict the next cell location. Once the next location is known, the history is appended with one character (standing for one cell), and the predictor updates its history to prepare for the next prediction.

In this paper, we focus on the detection of misbehaviors utilizing users' mobility patterns. It is possible that after an intruder successfully "fakes" the authentic user's mobile phone, he may keep static or semistatic when he makes the phone calls. Several existing work has been proposed to utilize other features, such as call residence time [20], to detect this kind of potential misbehaviors. In this paper, we do not consider this case, but in further work, we will extend our intrusion detection system to handle this case so that more intruders may be identified.

In our LZ-based scheme, we adopt LZ algorithms [7], [21]. In the rest of the paper, when we discuss these algorithms, we use the word character. When we apply them to cellular mobile networks, we use the word cell. These two words have the same meaning in their respective contexts. Similarly, the word string is used in discussing LZ algorithms, while cell list is used in cellular mobile networks.

A. Feature Extraction

The first step in intrusion detection is to extract effective features. Features are security-related measures that could be used to construct suitable detection algorithms. Effective features must be selected to reflect the subject activities. In our environment, we build the normal profiles of mobile users with regular movement patterns in cellular mobile networks. Under the assumption that each user will have his own favorite itineraries, the cell list traversed by each user is an ideal candidate feature for our usage. It is relatively stable, and the resulting alphabet, i.e., different cells, is small. To be specific, we denote each cell as a character. Therefore, a string could represent the path taken by a user. This string (or cell list) will feed into our model to construct the mobility trie or the fixed-order Markov model.

B. LZ-Based Intrusion Detection

1) Optimal Data Compression: Data compression is the encoding of data to minimize its representation. Some of the most common lossless compression algorithms used in practice are dictionary-based schemes, where a dictionary $D = (M, C)$ is a finite set of phrases M and a function C that maps M onto a set of codes. In practice, when no *a priori* knowledge of the source characteristics is available, the problem of data compression becomes considerably complicated. Therefore, we often resort to universal coding schemes whereby the coding process is interlaced with a learning process for the varying source characteristics.

```
Input: string S to be encoded
Output: the parsed string stored in the dictionary

BEGIN
    Initialize the dictionary D:= EMPTY
    Initialize the current prefix P := EMPTY
    LOOP
    C := next character in S
    IF string (P + C) exist in the dictionary D
    THEN P := P + C
    ELSE
        Add string (P + C) to the dictionary D
        P := EMPTY
    IF no more characters exist in S
    THEN break
    FOREVER
END
```

Fig. 3. LZ78 algorithm (+ means concatenation).

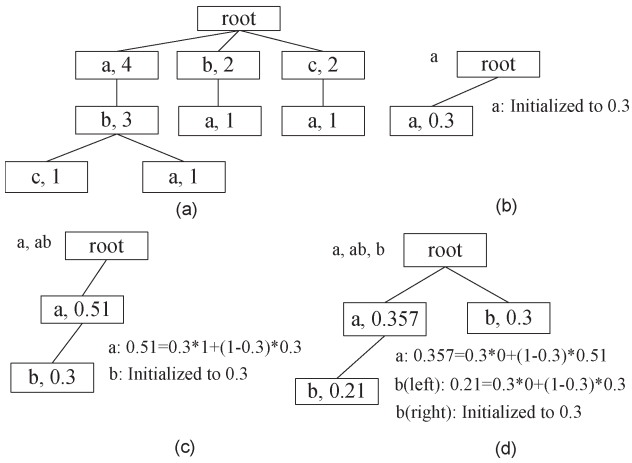

Fig. 4. Example of mobility trie and example of building mobility trie. (a) Example mobility trie. (b) When (a) is parsed. (c) When $(a)(ab)$ is parsed. (d) When $(a)(ab)(b)$ is parsed.

The family of LZ algorithms belongs to dictionary-based text compression and encode techniques [8]. They are based on a popular incremental parsing algorithm by Ziv and Lempel [7], [21] and have been widely used in data compression. Since its invention, many variations have been developed, and LZ78 is the most popular one.

The original LZ78 [7] is a word-based data compression algorithm. It parses the input string S of size n in a greedy manner into distinct substrings $x_1, x_2, \ldots, x_m$ with the following property: For $j > 1$, there exists a number $i < j$ that makes x_j equal to x_i concatenated by c, where c is one character in the alphabet. This is the so-called prefix property [8]. In the parsing process, if a phrase is the longest matching phrase seen previously concatenated by one character, the phrase, which is called a new phrase, is added to the dictionary. Substring x_j is encoded by the value i, using $\lceil \lg(j-1) \rceil$ bits, which is followed by the ASCII encoding of the last character of x_j, using $\lceil \lg \alpha \rceil$ bits, where α is the size of the input string's alphabet. Here, the base of the logarithm is 2. Word-based LZ78 algorithm is illustrated in Fig. 3.

The Ziv–Lempel algorithm can be converted from a word-based method to a character-based algorithm by building a probabilistic model that feeds probability information to an arithmetic coder [12], which encodes a sequence of probability of p using $\lg(1/p) = -\lg p$ bits.

LZ78 is both theoretically optimal and good in practice. When the input text is generated by a stationary and ergodic source, LZ78 algorithms enjoy the property of being asymptotically optimal as the input size increases. That is, it codes an indefinitely long string in the minimum size dictated by the entropy of the source. See Section VI-B4 for a proof. Being good in practice means that searching of LZ78 can be implemented efficiently by inserting each phrase in a trie data structure.

A trie is suitable to store the parsed phrases and is a multiway tree with any path from the root to a unique node forming a string. In a trie, only the unique prefix of each string is stored because the suffix can be determined by searching the string. A longest match is found by following down the tree until no match is found, or the path ends at a leaf.

Here is an example of how to parse a string using LZ78 algorithm and construct a trie. Suppose the alphabet A is (a, b, c), and one possible string S over this alphabet is $aabbabcccabaaba\ldots$. Each element of the alphabet A could be one possible cell that the user visits. S could be one possible cell list traversed by this user. Each substring in the parse is encoded as a pointer followed by an ASCII character. Based on the greedy parsing manner, which is shown in Fig. 3, this string will be parsed into phrases listed as follows: $(a)(ab)(b)(abc)(c)(ca)(ba)(aba)\ldots$. The match ab of the eighth substring aba is encoded using $\lceil \lg 7 \rceil$ bits with a value of 2, because the match ab is the second substring, and the last character a is encoded using $\lceil \lg 3 \rceil$ bits, because the alphabet size is 3.

In the character-based version of the Ziv–Lempel encoder, a trie is built when the previous substring ends. A trie at the start of the ninth substring is shown in Fig. 4. The number associated with each node indicates the frequency in terms of the number of times that this node has been parsed in the construction of the mobility trie.

This trie characterizes the probability model of the string $aabbabcccabaaba\ldots$. There are four previous substrings beginning with an a, two beginning with a b, and two beginning with a c. Therefore, the probability of a at the root is $4/8 = 1/2$. Similarly, the probability of b at the root is $2/8 = 1/4$, and the probability of c at the root is $2/8 = 1/4$. Of the four substrings that begin with an a, three begins with b. Therefore, the probability of b from a is $3/4$.

2) Probability Calculation: The probability calculation is based on the prediction by partial matching (PPM) [10] scheme. Here, we use a context model to predict the next character based on the previous consecutive characters. Specifically, we use a mth-order Markov model to model the sequence. That is, we use the consecutive previous m characters to predict the next character and calculate its probability. Here, m is the order of the Markov model. For a first-order $(m = 1)$ Markov model, it assumes that the next event only depends on the last event in the past. A high-order ($m > 1$ order) Markov model assumes that the next event depends on multiple (m) events in the past.

A tradeoff exists here. If the order m is too small, the prediction will be poor in the long run because little audit data will be available to make a decision. However, if the order is

too large, most contexts will seldom happen, and initially, the probability estimation will have to solely rely on the resolve of zero-frequency problems [8]. Based on these considerations, we take a blending approach, where the predications of several contexts of different lengths are combined into a single overall probability. It uses a number of models with different orders to compute the probabilities, respectively, assign a weight to each model, and calculate the weighted sum of the probabilities.

Let us denote the maximum order as m. The next character, denoted by α, is predicted on the basis of previous i characters. For each character α, let $p_i(\alpha)$ be the probability assigned to α by the finite-context model of order i. Note that when i is zero, the probability of each character is estimated independently of other characters. If the weight given to the model of order i is w_i and the blending weight vector is $[w_0, w_1, \ldots, w_m]$, the blended probability $p(\alpha)$ is computed as $p(\alpha) = \sum_{i=0}^{m} w_i * p_i(\alpha)$, where the sum of weights is normalized to 1. The larger the order, the larger the weight assigned to it, because context models with larger orders tend to be more accurate and should weigh more in the current normal profile. The maximum order m and the weight w_i are design parameters and will be discussed in Section VII.

3) Anomaly Detection Algorithm: We adopt the character-based LZ78 to deal with the anomaly detection problem, and a classifier is trained with known "normal" data to distinguish normal behaviors from anomalous ones.

a) Integration of EWMA into the mobility trie: In anomaly detection, each subject (i.e., user in this application) has a normal profile. For an individual subject, its activity may change over time. Therefore, it is necessary for the normal profile to be updated to reflect the recent activities. In our situation, the normal profile of the user activity should be dynamic. Generally, activities in the recent past should weigh more than activities long time ago. Adaptively modifying the normal profile correspondingly is a suitable mechanism.

Based on these considerations, we integrate EWMA [11] to the mobility trie. The mobility trie is modified when a new phrase is formed during the string parsing. When a new phrase is inserted, we say an event happens. Note that this event corresponds to a sequence of characters. The insertion of the new phrase needs to modify the existing frequency of the mobility trie. We will call the modified frequency EWMA-based frequency hereafter. EWMA-based frequency measures how often the corresponding node appears in the recent past. Note that we do not need to do an extra trie search to modify the frequency. Instead, it is done at the same time with the update of the mobility trie to improve efficiency.

The EWMA-based frequency of each node in the mobility trie is updated as follows:

$$F(i) = \lambda * 1 + (1 - \lambda) * F(i) \tag{1}$$

where node i is one item of the corresponding events, and

$$F(i) = \lambda * 0 + (1 - \lambda) * F(i) \tag{2}$$

where node i is not one item of the corresponding events.

Here, $F(i)$ is the EWMA-based frequency value stored in node i after a new phrase is inserted. For example, in Fig. 4(c),

```
initialize mobility database := null

LOOP
    wait for a sequence s
    IF (the mobility trie of the mobile exists)
        IF (a path p corresponding to s is found)
            add s to the mobility trie
            using EWMA to modify the frequencies of nodes
        ELSE
            create new nodes, and initialize their frequencies to λ
    ELSE
            1) create a mobility trie := single sequence s
            2) initialize the frequencies for every node in sequence
            s to λ
FOREVER
```

Fig. 5. Integrating EWMA into the mobility trie construction.

the EWMA-based frequency associated with a is 0.51. The EWMA-based frequency associated with b is 0.3. Here, λ is a smoothing constant that determines the decay rate. If a node i is not observed for continuous k events (one event happens when a new phrase is inserted), the EWMA-based frequency of node i will be decayed to $(1 - \lambda)^k$. In this way, the EWMA-based frequency of each node measures the intensity of this node over the recent past.

Continuing the example illustrated in Fig. 4(a), we illustrate how to integrate EWMA into the construction of the mobility trie. In this example, we let λ be 0.3. When the first character a is parsed, the corresponding mobility trie is illustrated in Fig. 4(c). When ab is parsed, the corresponding mobility trie is illustrated in Fig. 4(d). When b is parsed, the corresponding mobility trie is illustrated in Fig. 4(d). As we can see, the EWMA-based frequency value associated with each node is exponentially faded. The EWMA-based mobility trie construction is summarized in Fig. 5.

b) Similarity measure: EWMA-based mobility trie maintains the stationary part of each user's recent activities. Based on this, we could accurately predict whether the future activities are normal or not.

Let the sample space be all the possible cells traversed by a user. Because a user has his favorite routine of activity, this could lead to a small set of sample space. Let $S = (X_1, X_2, \ldots, X_n)$ denote the observed activities of the user, where X_i denotes a cell number. We want to identify whether or not it is normal based on our constructed mobility trie. We use a high-order Markov model to compute its blending transition probabilities.

Given an order o of the Markov model, we define the oth-order probability of S as follows:

$$P_o = \sum_{i=1}^{n-o} P(X_{I+o} | X_i, X_{i+1}, \ldots, X_{i+o-1}). \tag{3}$$

When it is an order-0 model $(o = 0)$, the probability of S is calculated as $P_o = P_0 = \sum_{i=1}^{n} P(X_i)$.

To calculate the probability of the transition $(X_i, X_{i+1}, \ldots, X_{i+o-1}) \longrightarrow X_{i+o}$ in (3), we need to search $(X_i, X_{i+1}, \ldots, X_{i+o-1})$ from the root. Let $F(X_{i+o})$ denote the EWMA-based

frequency of node X_{i+o}. If $(X_i, X_{i+1}, \ldots, X_{i+o-1})$ is found, the probability $P(X_{i+o}|X_i, X_{i+1}, \ldots, X_{i+o-1})$ is defined as follows:

$$P(X_{i+o}|X_i, X_{i+1}, \ldots, X_{i+o-1}) = \frac{F(X_{i+o})}{F(X_{x+o-1})}. \quad (4)$$

If $(X_i, X_{i+1}, \ldots, X_{i+o-1})$ is not found, its probability is assigned 0.

To calculate $P(X_i)$, we need to compute the sum of the EWMA-based frequency of the root's children. $P(X_i)$ is then defined as $F(X_i)/\sum F(X_{\text{root's children}})$.

If X_i is not a child of the root, $P(X_i)$ is 0. That is, we only search from the root to decide the probability of each X_i.

Take the trie illustrated in Fig. 4(d) as an example; $P(a) = 0.357/(0.357 + 0.3) = 0.5434$, $P(b) = 0.3/(0.357 + 0.3) = 0.4566$, and $P(b|a) = 0.21/0.357 = 0.5882$.

Suppose that the blending weight vector is $[w_0, w_1, \ldots, w_m]$, where w_i is the weight value associated with the ith-order Markov model. $\sum_{i=0}^{m} w_i = 1$ and $w_i \geq 0 \quad \forall i$. The probabilities of string S is defined as $P = \sum_{i=0}^{m} w_i * P_i$.

Intuitively, P increases with the increase of S's length because more transitions will be considered when S is longer. Therefore, P is not a good metric. We propose to use the following metric as our similarity measure similarity$(S) = P/\text{Length}(S)$, where Length(S) is the length of string S.

Based on our definition, the similarity measure could be normalized by the length of the string and provides good criteria to evaluate its normalcy. Intuitively, similarity indicates how good a mobile user follows its routines.

For the input string S, we calculate its similarity(S). When a user follows one of its favorite itineraries, because this path is integrated into the mobility trie to construct the normal profile, many of its transitions, which are illustrated in (4) at different orders o, will be found in the mobility trie, i.e., normal profile. Based on our definition, similarity(S) will be a relatively large value. However, when the mobile is stolen and the intruder takes an infrequent path, the similarity of this string tends to be a very small value, because many transitions cannot be found in the mobility trie.

We introduce a threshold P_{thr}, which is a design parameter. When similarity$(S) \geq P_{\text{thr}}$, string S is evaluated as normal, otherwise, string S is identified as anomalous.

Because our mobility trie records the most frequently used path of a user, it is very sensitive to anomalous paths, even if they are very short strings. This enables our detection algorithm to detect the abnormal very quickly—an important quality for reducing potential damage by a malicious user. At the same time, our detection algorithm has a very high detection rate. Furthermore, when a frequently used path is taken, our detection algorithm can tolerate slight variations from the path and thus has a small false-positive rate.

4) Theoretical Analysis of LZ-Based Intrusion Detection Scheme: Because our intrusion detection scheme is derived from LZ data compression algorithm, we first analyze the optimality of the word-based LZ algorithm and show that the character-based LZ algorithm is at least as good as the word-based scheme. Then, we will show that our intrusion detection scheme inherits the optimality of these data compression algorithms.

Given a sequence x^n of length n over an alphabet A of α letters, where x^n is the sequence, and n is its length, word-based LZ data compressor parses it into different phrases, $x_1, x_2, \ldots, x_t$. Let $t(x^n)$ denote the maximal possible number of distinct phrases. For an information lossless (IL) data compressor C (a class of data compression algorithms that allows the original data to be reconstructed exactly from the compressed data), the compression ratio $\rho_C(x^n)$ can be calculated as follows:

$$\rho_C(x^n) = |C(x^n)| / n \lg(\alpha) \quad (5)$$

where the base of the logarithm is 2.

Here, $|C(x^n)|$ denotes the length (in bits) of the output that C produces on x^n. $n \lg(\alpha)$ is the entropy of x^n. Because, ideally, the length of a message after it is encoded should be equal to its entropy, (5) represents the compression ratio of the compressor C.

Let $\rho_\sigma(x^n)$ denote the best compression ratio attainable for x^n by any IL compressor of σ states. Sequence x^n is parsed into different phases: $x^n = x_1, x_2, \ldots, x_t$. The maximum possible number of distince phrases is $t(x^n)$. Define $q(x^n) = (t(x^n) \lg(t(x^n)))/(n \lg(\alpha))$.

It is shown that [7] $\rho_\sigma(x^n) \geq q(x^n) - \delta(\sigma, n)$ with $\lim_{n\to\infty} \delta(\sigma, n) = 0$. The quantity $q(x) = \lim_{n\to\infty} \sup t(x^n) \lg(t(x^n))/n \lg(\alpha)$, which is measurable on x by parsing it into distinct phrases, establishes a low bound on the best compression. The word-based LZ algorithm achieves a compression ratio that is (asymptotically) equal to $q(x)$. That is, its compression ratio will continually approach $q(x)$ but never actually reach it. Thus, the algorithm is universal and asymptotically optimal.

The coding length obtained in the character-based LZ algorithm is shown in [8] to be at least as good as that obtained using the word-based approach. Therefore, the character-based LZ algorithm is also universal and asymptotically optimal.

If the false alarm rate is defined as the ratio of the total number of event false alarms that our scheme incurs over the total number of alarms, and the expected false alarm rate is defined as the best possible false alarm rate achievable by any intrusion detection algorithm that makes its predication based only on the past history, the following theorem holds.

Theorem 1: If the source is a stationary mth-order Markov source, the expected value of the false alarm rate of the intrusion detection scheme derived from the LZ algorithm is within an additive factor of $O(1/\sqrt{n})$ from the expected false alarm rate of the source, where n is the length of the source sequence.

For the proof of Theorem 1, please refer to [12]. The same is true for detection rate. This theorem shows that our intrusion detection algorithm inherits the asymptotic optimality of the LZ algorithm after it converges.

5) Implementation Issues: In practice, an important issue is how to store the mobility information in a trie. A trie is actually a multiway tree with a path from the root to a unique node for each string represented in the tree. The fastest approach for processing is to create an array of pointers for each node in the trie with a pointer for each character of the input alphabet.

Although this approach is easy for processing, it wastes memory space. Another approach is to use a linked list at each node, with one item for each possible branch. This method uses memory economically, but the processing is intensive. A trie can also be implemented as a single hash table with an entry for each node. For further details, the reader can consult books on algorithms and data structures.

C. Markov-Based Anomaly Detection

Markov predictors are a very popular family of predictors. They have been widely used and studied in the literature [13]. Let X_t be the cell visited by the user or the state of the user's activity at time t. The order-o Markov predictor assumes that the location can be predicted from the current context, which is the sequence of the previous o most recent characters in the location history $(X_{t-o+1}, X_{t-o}, \ldots, X_t)$. Under this Markov model, the transitions represent the possible cell locations that follow the context.

A Markov chain with order o of only one-step event transitions is a stochastic process with the following assumptions:

$$\begin{aligned}
&P(X_{t+1}=i_{t+1}|X_t=i_t, X_{t-1}=i_{t-1}, \ldots, X_0=i_0)\\
&=P(X_{t+1}=i_{t+1}|X_t=i_t, X_{t-1}=i_{t-1}, \ldots, X_{t-o+1}=i_{t-o+1})\\
&P(X_{t+1}=i_{t+1}|X_t=i_t, X_{t-1}=i_{t-1}, \ldots, X_{t-o+1}=i_{t-o+1})\\
&=P(X_{t+1}=j|X_t=i_o, X_{t-1}=i_{o-1}, \ldots, X_{t-o+1}=i_1)\\
&\equiv p_{\{i_1,\ldots,i_{o-1},i_o\}\to j}.
\end{aligned}$$

It describes the two important properties of the Markov Chain [11], which are stated as follows:

1) Equation (6) states that the probability distribution of the user at time $t+1$ depends on the state at time $t, t-1, \ldots, t-o+1$ and does not depend on the previous states leading to the states at $t, t-1, \ldots, t-o+1$.
2) Equation (6) states that the state transitions from time $t, t-1, \ldots, t-o+1$ to $t+1$ is independent of time.

If the system has a finite number of states $1, 2, \ldots, s$, these probabilities could be represented in a transition probability matrix, where each element in the matrix is $p_{\{i_1,\ldots,i_{o-1},i_o\}\to j}$, as follows:

$$\begin{bmatrix}
p_{\{1,1,\ldots,1\}\to 1} & p_{\{1,1,\ldots,1\}\to 2} & \cdots & p_{\{1,1,\ldots,1\}\to s}\\
p_{\{1,1,\ldots,2\}\to 1} & p_{\{1,1,\ldots,2\}\to 2} & \cdots & p_{\{1,1,\ldots,2\}\to s}\\
\vdots & \vdots & \vdots & \vdots\\
p_{\{s,s,\ldots,s\}\to 1} & p_{\{s,s,\ldots,s\}\to 2} & \cdots & p_{\{s,s,\ldots,s\}\to s}
\end{bmatrix}. \quad (6)$$

$p_{\{i_1,\ldots,i_{o-1},i_o\}\to j}$ could be learned from the observations of the user' locations in the past. When $o \geq 1$, $P(X_{t+1} = j|X_t = i_o,\ X_{t-1} = i_{o-1}, \ldots, X_{t-o+1} = i_1) = N(Lj)/N(L)$, where $L = \{i_1, \ldots, i_{o-1}, i_o\}$, $N(Lj)$ denotes the number of observation pairs of L and j. $N(L)$ denotes the number of observations of L.

When o is 0, the formula becomes

$$P(X_{t+1} = j) = \frac{N(j)}{N} \quad (7)$$

where N is the total number of observations (i.e., total number of cells). $N(j)$ is the number of observations of a.

Given this estimation, we can calculate the probability of the next location given the previous o locations for a specific user. The larger the probability, the more likely it is normal. We can then derive a threshold policy and use it to decide whether the current activity is normal or not.

That is, given a fixed-order value o and an observed activity in terms of a cell list $S_{\text{observed}} = (X_1, X_2, \ldots, X_n)$, where each X_i denotes a cell number. For $o \geq 1$, we first calculate its o-order transition probabilities as $P_o = \sum_{i=1}^{n-o} P(X_{i+o} = j|X_i = i, X_{i+1} = i+1, \ldots,\ X_{i+o-1} = i+o-1) = \sum_{i=1}^{n-o} p_{\{i,i+1,\ldots,i+o-1\}\to j}$, where $p_{\{i,i+1,\ldots,i+o-1\}\to j}$ can be retrieved from the probability transition matrix whose element is obtained using (6). If the transition does not exist in the transition matrix, we assign $P(X_{i+o}|X_i, X_{i+1}, \ldots, X_{i+o-1})$ to 0.

For $o = 0$, its probability could be calculated as $P_o = \sum_{i=1}^{n} P(X_i = j)$, where $P(X_i = j)$ can be obtained from (7).

Similar to LZ-based mechanism, P_o increases with an increase in S's length. Therefore, for the Markov-based prediction, we also define the following similarity metric: similarity$(S) = P_o/\text{Length}(S)$, where Length$(S)$ is the length of string S.

For the input string S, we calculate its similarity(S). If most transitions can be found, similarity(S) tends to be large. This indicates that S is more likely to be normal. However, if the mobile is stolen and an infrequent or new path is taken, the similarity of the string should be small.

When the mobile is at low mobility, the user usually travels one or two cells during the call. Given a fixed o, it is highly possible that the length of the transition $(o+1)$ is larger than the length of the cell. The Markov-based prediction cannot make a decision under this situation. Therefore, a high-order Markov-based prediction will become helpless for low mobility data. We make a random guess when this situation happens. For example, with a probability of 1/2, this cell list is identified as normal (abnormal).

For the Markov-based prediction, we introduce a threshold $P_{\text{thr\_markov}}$. If similarity$(S) \geq P_{\text{thr_markov}}$, string S is evaluated as normal. $P_{\text{thr\_markov}}$ should be tuned by taking into consideration both false alarm rate and detection rate.

VII. Simulation Study

We use the C language to simulate mobile users' activities at different mobility levels to watch the performance of our anomaly detection schemes. To the best of our knowledge, there is no other work that intends to construct anomaly detection models over users' mobility data in cellular mobile networks.

A. Data Sets

A generalized graph model is used in our simulation to represent a cellular network of 40 cells, each having six neighbors

on the average. The average distance between two base stations is 1 mi. To avoid the edge effect of the finite network size, wraparound is applied to the edge cells. Because most mobile users have favorite routes in reality, in this experiment, each mobile user has five possible paths in the network. A mobile user will take these five paths with probabilities of 0.6, 0.2, 0.1, 0.05, and 0.05, respectively. The paths are generated as follows: 1) Select two cells in the graph randomly as the original and destination cells. 2) Whenever the mobile user leaves the current cell, it moves to a neighboring cell that is closest to the destination. Call durations are exponentially distributed with a mean value of 3 min. With a fixed call duration, the higher the speed, the longer the cell list. Because mobile users travel with different speeds, we consider five mobility levels. The speeds of mobiles are 20, 30, 40, 50, and 60 mi/h in the five cases, respectively.

Note that the aforementioned mobility data sets are generic enough for most users. However, it may not be suitable for users with totally random movement behaviors such as taxi drivers.

For the LZ-based scheme, $m = 2$. That is, we apply a blended Markov model with order 0, order 1, and order 2 to the data sets. The weights for order 0, order 1, and order 2 are 0.1, 0.2, and 0.7, respectively.

For the Markov-based detection method, four schemes are considered with order 0, order 1, order 2, and order 3, respectively. We set λ to 0.3, a commonly used smoothing constant.

B. Performance Metric

1) False alarm rate: It is measured over normal itineraries. Suppose that j normal itineraries are measured, and i of them are identified as abnormal. The false alarm rate is defined as i/j.
2) Detection rate: It is measured over abnormal itineraries. Suppose that j abnormal itineraries are measured, and i of them are detected. The detection rate is defined as i/j.

Because both LZ-based and Markov-based predictors are online predictors, the number of training sequences has an impact on the performance of the proposed detection schemes. Based on this consideration, we measure false alarm rate and detection rate in the proposed schemes with different numbers of training sequences. Detailed discussions in this respect are presented in Section VII-C3.

C. Simulation Results and Analysis

1) False Alarm Rate: Simulation results of the false alarm rate are illustrated in Fig. 6. First, we observe that the false alarm rate of the LZ-based scheme is lower than those of Markov-based schemes with different orders. This is because the LZ-based prediction uses EWMA to make the current activities weigh more in the probability calculation and utilizes the blending scheme to calculate the overall probability, which takes into account predictions with different orders. On the other hand, fixed-order Markov predictions are used in the Markov-based detection schemes. Therefore, the anomaly detection in the LZ-based scheme is more accurate compared to those of the Markov-based schemes.

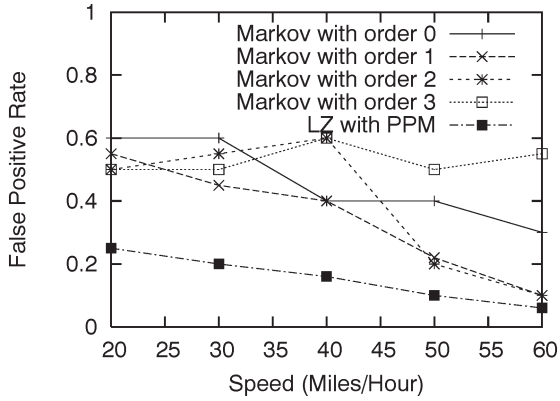

Fig. 6. False alarm rate at different mobility levels.

Second, we observe that, generally, the false alarm rate of all schemes decreases when the mobility level increases. With the increase of mobility, a user tends to traverse more cells during a call, and more mobility information about this user is stored in the database. Therefore, for a normal user with regular movement patterns, his itinerary will demonstrate more resemblance to his regular activities. Consequently, the false positives are reduced.

Third, for Markov-based schemes, their performance in terms of false alarm rate is not desirable, especially when the mobility is low. We can see that Markov-based schemes can have up to 50% false alarm rate when the mobility is low (for example, 20 mi/h). When the mobility is very low, a user will traverse only one or at most two cells during a call in our simulation. This makes it very difficult to identify whether the mobility is normal or not, especially when a higher order Markov-based scheme is used. For a very short cell list, it may not match any context of the Markov model at a given order. Thus, Markov-based schemes become ineffective in making correct decisions under these situations. The random guess mechanism described in Section VI-C could lead to a very high false alarm rate. When the itinerary is relatively long, the situation is better. However, it is still prone to relatively high false-positive rates. This is because the valid paths taken in the training data may happen with very low probability. Therefore, even if the similar path is taken again, it is still possible to be identified as an abnormal path.

2) Detection Rate: Simulation results of the detection rate are illustrated in Fig. 7. First, we observe that the detection rate of the LZ-based scheme is higher than those of Markov-based schemes with different orders. The reason is similar to that in the false alarm rate case. The LZ-based prediction utilizes EWMA and blending schemes to calculate the overall probability with different orders. This makes it more accurate compared to the Markov-based schemes with fixed orders.

Second, we observe that, generally, the detection rate of all schemes increases with the increase of the mobility level. Again, the reason is similar to that in the false alarm rate case. With the increase of mobility, the user tends to traverse more cells during a call. Therefore, for a masquerader, his itinerary tends to deviate significantly from the normal profile. In this way, the detection rate is improved with the increase of mobility.

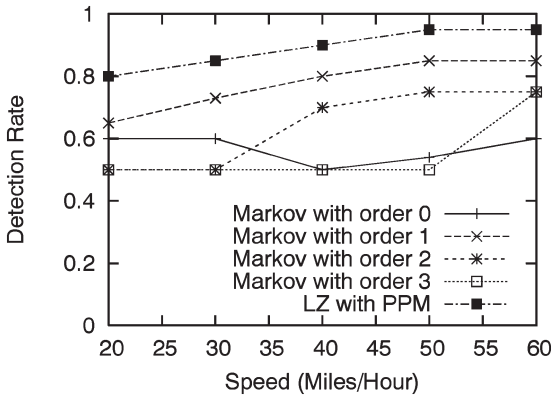

Fig. 7. Detection rate at different mobility levels.

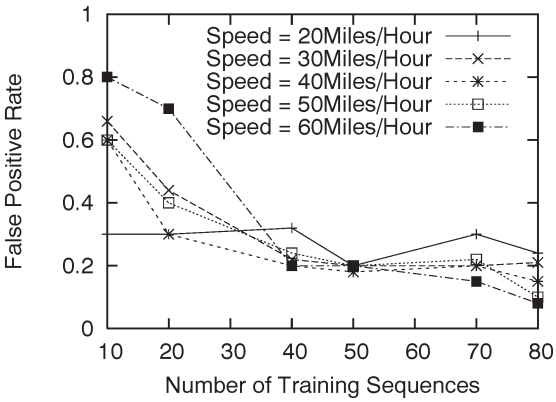

Fig. 8. False alarm rate versus the number of training sequences.

Third, for the Markov-based schemes, their performance in terms of detection rate is not desirable, especially when the mobility is low. Simulation results show that when the mobility is low (for example, 20 mi/h), the Markov-based schemes can only achieve 50% detection rate. For a very short cell list, it may not match any context of a Markov model at a given order. Thus, Markov-based schemes become ineffective in detection under these situations. In the simulation, we cannot achieve 100% detection rate in any case, because when the itinerary is relatively long, it is possible that part of the intruder's path overlaps with some normal paths. Therefore, it is still possible to miss the detection of these kinds of itineraries.

3) Effects of the Number of Training Sequences: We have demonstrated that the LZ-based detection scheme performs better than the Markov-based detection scheme. Therefore, in this section, we use the LZ-based detection scheme to illustrate the impact of the number of training sequences on the detection performance in terms of false alarm rate and detection rate. This could also help us to answer how long the user profile could be applied after online training. That is, when the false alarm rate and the detection rate become relatively flat, the user profile is ready to be used.

Fig. 8 illustrates the false alarm rate using different numbers of training sequences. First, we observe that, at all mobility levels, with the increase of the number of training sequences, the false alarm rate decreases. This demonstrates the impact of the number of training sequences on the detection performance in terms of false alarm rate. With the increase of the number of training sequences, the constructed mobility trie can characterize the normal profile better and predict the user's activity more accurately. Therefore, it could result in a lower false alarm rate. We also observe that after a large number of training sequences has been used to construct the model, the false alarm rate becomes relatively stable, i.e., it does not change much with the increase of the number of training sequences.

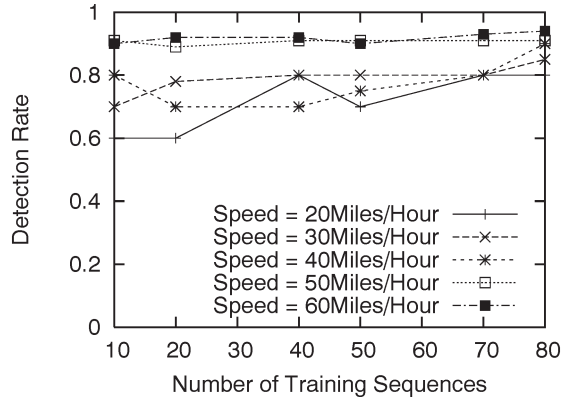

Fig. 9. Detection rate versus the number of training sequences.

Second, it can be observed that the lower the mobility, the slower the false alarm rate decreases with the increase of the number of training sequences. When the mobility is low, the cell list is shorter. Therefore, the portion of "randomness" will play an important role in making the final decisions. Under this situation, the number of training sequences only has little impact. Therefore, for low mobility levels, the false alarm rate decreases slower with the increase of the number of training sequences.

Fig. 9 shows the detection rate using different numbers of training sequences. First, we observe that, at high mobility levels, with the increase of the number of training sequences, the detection rate keeps roughly the same value. This is because for a user activity that is abnormal, it will keep abnormal no matter how many training sequences are utilized to construct the model. Therefore, the number of training sequences has little impact on the detection rate.

Second, we see that, at low mobility levels, the detection rate increases with the increase of the number of training sequences. The reason is similar to the one stated earlier: When the mobility is low, the cell list is shorter. Nevertheless, when more training data are integrated into the system, it becomes more accurate to make final decisions. Therefore, for low mobility levels, its detection rate increases with the increase of the number of training sequences.

VIII. Conclusion and Future Work

This paper presented two approaches to construct an end user's mobility profile for anomaly intrusion detection in wireless cellular networks. In the first method, based on optimal data compression techniques, each user's itinerary was modeled as an mth-order Markov source, and EWMA was applied to make the normal profile up-to-date. An intrusion detection scheme was then developed to detect potential internal

attackers—masqueraders. In the second method, we used a fixed-order Markov-based detection scheme.

We performed a simulation study to compare the performance of different detection schemes. Simulation results demonstrated that the LZ-based detection scheme can achieve more desirable performance in terms of false alarm rate and detection rate. Markov-based detection schemes, especially those with higher orders, are not very effective in anomaly intrusion detection in cellular mobile networks due to relatively lower user mobility in terms of the number of cells traversed compared to other wireless networks, such as the wireless local area network (WLAN). This observation differs significantly from the phenomenon described in [13].

In this paper, we have only considered mobility patterns as feature values, which may not be accurate for some particular types of users such as taxi drivers. In our future work, more features such as call history and activities will be accommodated into the system to make it more suitable to all users.

REFERENCES

[1] 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.

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

[3] A. Bhattacharya and S. K. Das, "LeZi-update: An information-theoretic approach to track mobile users in PCS networks," in *Proc. 5th Annu. ACM/IEEE Int. Conf. MOBICOM*, Seattle, WA, Aug. 1999, pp. 1–12.

[4] F. Yu and V. C. M. Leung, "Mobility-based predictive call admission control and bandwidth reservation in wireless cellular networks," *Elsevier Comput. Netw.*, vol. 38, no. 5, pp. 577–589, Apr. 2002.

[5] I. F. Akyildiz and W. Wang, "A predictive user mobility profile for wireless multimedia networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 6, pp. 1021–1035, Dec. 2004.

[6] C. Cheng, R. Jain, and E. Berg, "Location prediction algorithms for mobile wireless systems," in *Handbook of Wireless Internet*, M. Illyas and B. Furht, Eds. Boca Raton, FL: CRC, 2003.

[7] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 5, pp. 530–536, Sep. 1978.

[8] T. C. Bell, J. G. Cleary, and I. H. Witten, *Text Compression*, ser. Prentice-Hall Advanced Reference Series. Englewood Cliffs, NJ: Prentice-Hall, 1990.

[9] 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.

[10] J. G. Cleary and I. H. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Trans. Commun.*, vol. COM-32, no. 4, pp. 396–402, Apr. 1983.

[11] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*. Upper Saddle River, NJ: Prentice-Hall, 1998.

[12] J. S. Vitter and P. Krishnan, "Optimal prefetching via data compression," *J. ACM*, vol. 43, no. 5, pp. 771–793, Sep. 1996.

[13] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating location predictors with extensive Wi-Fi mobility data," in *Proc. 23rd Annu. Joint Conf. INFOCOM*, Hong Kong, Mar. 2004, pp. 1414–1424.

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

[15] P. Porras and A. Valdes, "Live traffic analysis of TCP/IP gateways," in *Proc. ISOC Symp. NDSS*, San Diego, CA, Mar. 1998.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] R. Büschkes, D. Kesdogan, and P. Reichl, "How to increase security in mobile networks by anomaly detection," in *Proc. Comput. Security Appl. Conf.*, Phoenix, AZ, Dec. 1998, pp. 3–12.

[21] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inf. Theory*, vol. IT-23, no. 3, pp. 337–342, May 1977.

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

Fei Yu (S'00–M'04) received the M.S. degree in computer engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 1998 and the Ph.D. degree in electrical engineering from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2003.

From 1998 to 1999, he was a System Engineer at China Telecom, Beijing, working on the planning, design, and performance analysis of national signaling system 7 and global system for mobile communications networks. From 2002 to 2004, he was a Research and Development Engineer at Ericsson Mobile Platforms, Lund, Sweden, where he worked on dual-mode universal mobile telecommunications system/general packet radio service handsets. He is currently a Research Associate at UBC. His research interests are quality of service, cross-layer design, and mobility management in wireless networks.

Kui Wu (S'98–M'02) received the Ph.D. degree in computing science from the University of Alberta, Edmonton, AB, Canada, in 2002.

He then joined the Department of Computer Science, University of Victoria, Victoria, BC, Canada, where he is currently an Assistant Professor. His research interests include mobile and wireless networks, sensor networks, network performance evaluation, and network security.

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

He was with Micro Linear, San Jose, CA, as a Medium Access Control (MAC) Architect who was involved in the IEEE 802.11 standard enhancement work. In 2002, he joined the Department of Computer Science, University of Memphis, Memphis, TN. He is the Director of W4-Net Laboratory and is with the Center for Information Assurance (CEIA) of the University of Memphis. He will join the Department of Computer Science of the University of Alabama, Tuscaloosa, in August 2006. His research interests include wireless networks, mobile computing, and network security. He has published more than 120 papers in major journals and refereed conference proceedings related to these research areas.

Dr. Xiao was a Voting Member of the IEEE 802.11 Working Group from 2001 to 2004. He currently serves as the Editor-in-Chief of the *International Journal of Security and Networks (IJSN)* and the *International Journal of Sensor Networks (IJSNet)*. He serves as an Associate Editor or on the Editorial Boards of the following refereed journals: *International Journal of Communication Systems, Wireless Communications and Mobile Computing, EURASIP Journal on Wireless Communications and Networking*, and *International Journal of Wireless and Mobile Computing*. He served as the Lead or Sole Guest Editor of five journals from 2004 to 2005. He serves as a Referee/Reviewer of many funding agencies as well as a Panelist of NSF and a Member of the Telecommunications Expert Committee of the Canada Foundation for Innovation (CFI). He serves as TPC for more than 60 conferences such as INFOCOM, ICDCS, ICC, GLOBECOM, and WCNC.

Victor C. M. Leung (S'75–M'89–SM'97–F'03) received the B.A.Sc. (Hons.) degree in electrical engineering from the University of British Columbia (UBC), Vancouver, BC, Canada, in 1977, and was awarded the APEBC Gold Medal as the head of the graduating class in the Faculty of Applied Science. He attended graduate school at UBC on a Natural Sciences and Engineering Research Council Postgraduate Scholarship and received the Ph.D. degree in electrical engineering in 1981.

From 1981 to 1987, he was a Senior Member of Technical Staff at MPR Teltech Ltd., specializing in the planning, design, and analysis of satellite communication systems. He also held a part-time position as Visiting Assistant Professor at Simon Fraser University, Burnaby, BC, in 1986 and 1987. In 1988, he was a Lecturer in the Department of Electronics, Chinese University of Hong Kong. He returned to UBC as a Faculty Member in 1989, where he is currently a Professor and the Associate Head of Graduate Affairs in the Department of Electrical and Computer Engineering and a holder of the TELUS Mobility Industrial Research Chair in Advanced Telecommunications Engineering. His research interests are in the areas of architectural and protocol design and performance analysis for computer and telecommunication networks, with applications in satellite, mobile, personal communications, and high-speed networks.

Dr. Leung is a Voting Member of the ACM. He is an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and an Editor of the *International Journal of Sensor Networks*. He has served on the Technical Program Committee (TPC) of numerous conferences, as the TPC Vice-Chair of IEEE WCNC 2005, and as a General Co-Chair of ACM/IEEE MSWiM 2005.