

# Tracking User Mobility to Detect Suspicious Behavior

Gaurav Tandon\*

Philip K. Chan†

## Abstract

Popularity of mobile devices is accompanied by widespread security problems, such as MAC address spoofing in wireless networks. We propose a probabilistic approach to temporal anomaly detection using smoothing technique for sparse data. Our technique builds up on the Markov chain, and clustering is presented for reduced storage requirements. Wireless networks suffer from oscillations between locations, which result in weaker statistical models. Our technique identifies such oscillations, resulting in higher accuracy. Experimental results on publicly available wireless network data sets indicate that our technique is more effective than Markov chain to detect anomalies for location, time, or both.

## 1 Introduction

Wireless networks enable users to connect to a network without any physical connection. 802.11 wireless local area networks (WLANs) allow mobile hosts to join a network without being inside a building, hence bypassing traditional physical security. Publicly available tools like NetStumbler [21] can locate available WLANs while driving around in the neighborhood, called war driving [9]. Security experts have identified many WLAN attacks; they include ARP poisoning, MAC/IP spoofing, man-in-the-middle, session hijacking, and replaying [22, 24]. These attacks generally try to redirect traffic and masquerade as legitimate users with stolen identity.

Predicting user location has been extensively researched to determine paths traversed, primarily for mobility management to study effective handoff between cells with lower power requirements, and better load balancing. Mobility traces suffer from oscillations between locations, which can have an adverse effect on accuracy of the location predictors. To the best of our knowledge, there is no automated technique to identify oscillations.

This paper focuses on security issues for mobile hosts, such as MAC spoofing in WLAN. Anomaly detection is a one class problem that learns model from "normal" class labels, and significant deviations from

the learned model could be a result of potential misuse. For example, an employee is typically at work from 9 a.m. to 6 p.m. during weekdays, and at home most of the remaining time. A machine learning algorithm could model these contexts and make predictions to determine any anomalies. An unauthorized WLAN user, may result in contextual anomalies for the given model, either in terms of location, or time, or both.

**1.1 Problem Definition** A user mobility trace consists of a temporal sequence of locations, represented by the access point (or cell id) in WLANs. Given a training user trace  $\{(t_1, l_1), (t_2, l_2) \dots\}$ , where  $l_1$  and  $l_2$  are locations at time  $t_1$  and  $t_2$  respectively, an anomaly detection system creates a model for "normal" class. Test traces are compared against the learned model and significant deviations are flagged. The goal is an automated system that can generalize beyond seen data and detect suspicious behavior. Two additional desirable properties of such a system are identification of oscillations in mobility traces, and small size models.

**1.2 Approach** Considering the mobile nature of the problem, we propose *TLAD* (Temporal Location Anomaly Detection) that monitors user location at different time intervals and learns a probabilistic model. We assume that the device is used by a single user, and will interchangeably use the terms *user* and *device*. Our algorithm learns location probability distributions for a user, and uses a modified Markov based approach to anomaly detection. Oscillations are identified using statistical correlation tests between successive locations. In addition, a Kullback-Leibler divergence based agglomerative hierarchical clustering technique merges day profiles for concise modeling.

**1.3 Contributions** In this paper, we make three contributions:

1. Automated oscillation detection: Oscillations in mobility patterns are identified automatically. Oscillating locations are grouped together into locales, resulting in more accurate models.
2. Temporal location anomaly detection (*TLAD*): Location probability distributions are proposed to aid

\*Florida Institute of Technology and Nuance Communications. Email:gtandon@fit.edu

†Florida Institute of Technology. Email:pkc@cs.fit.edu

detection of MAC spoofing in WLANs. A modified Markov chain based algorithm called *TLAD* is presented.

3. Reduced model size: We propose clustering user day profiles for a compressed model. An agglomerative hierarchical algorithm is presented to reduce model storage requirements.

**1.4 Organization** Section 2 presents existing literature from location prediction in mobile computing and anomaly detection. A general Markov based approach to anomaly detection for mobile devices is described in Section 3. In Section 4, we propose modification to the Markov based anomaly detector, identification of oscillations in mobile data and clustering location profiles for reduced storage. Results and analysis of experiments on real data sets is presented in Section 5. Computational and storage overheads are also presented. Key findings are compiled in Section 6.

## 2 Related Work

Context-aware mobile computing has been an active area of research in the last decade. A radio frequency system was proposed in [1] to determine user location based on signal strength triangulation. Signal strength probability distributions and clustering for infrastructure LANs was studied in [27], and [26] proposed a framework for plan recognition in an indoor RF-based wireless network. Context-aware research has stressed on context extraction and learning [11, 8, 14], but the discerned context has not been applied to securing the devices themselves. The same is applicable to mobility modeling studies for WLANs, which have largely been pursued for mobility management, such as low power handoff between cells and load balancing [2, 3]. This paper demonstrates use of contextual information from mobility traces for *fingerprinting* mobile devices.

Research on machine learning methods for anomaly detection has been pursued to complement signature-based intrusion detection systems (e.g. anti-virus). These studies mostly focused on wired networks [23, 15], not wireless networks. A technique to detect mobile phone *cloning* fraud was proposed in [7], where patterns of fraud were learned and adapted to user call behavior. Our technique differs in that only normal patterns are learned and anomalies are flagged, and also applicable to *cloning* fraud detection. Suspicious large moving objects, such as ships, have been detected as anomalies [18]. Though it involves route modeling, it deals with additional attributes like speed and direction information generally not available on laptops and mobile phones. Even if a GPS (global positioning system) was

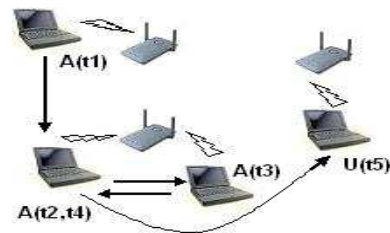


Figure 1: Mobility trace ( $t1 - t4$ ) for authorized user  $A$ , and misuse by unauthorized user  $U$  at time  $t5$ .

available on these devices, an intruder will likely disable it to evade such systems. This paper proposes temporal and location based device profiling for anomaly detection.

## 3 Approach

This section introduces the framework for detecting abnormalities attributed to unauthorized users. We describe how probabilities are learned and present a Markov based approach to anomaly detection. The scenario is represented in Fig. 1. For an authorized user  $A$  in a WLAN, his laptop associates with an access point (or re-associates with a new access point) and that is representative of its location. The information is sent to a centralized server, where a model is learned for the device location over different time intervals. The device may communicate with multiple access points, but all contextual information is routed to the server for model learning. Though the data lacks the physical topology and proximity of the actual locations, it is easy to extract and gives a reasonable location estimate. In Fig. 1, the model is created for locations at time instances  $t1 - t3$ . The learned model is then used to maintain conformity for device (time  $t4 - t5$  in the figure). Now consider the possibility of an unauthorized user  $U$  attempting to spoof the MAC address of user  $A$ 's laptop. Any subsequent usage ( $t5$  in Fig. 1) by  $U$  would most likely be inconsistent with the model in terms of location and time, raising an alarm. The wireless device could then be denied access to the network, or verify authenticity via a challenge-response mechanism.

We propose tracking the frequency of the wireless device, such as a laptop, at various locations within a fixed time interval. The frequency is then normalized across all possible locations and probability approximated at each of those locations during the time period. The probability of a mobile device  $m$  at location  $l$  during time interval  $t$  is estimated by:

$$(3.1) \quad P_m^t(l) = \frac{freq_m^t(l)}{\sum_l freq_m^t(l)}$$

$P_m^t$  is called the **location probability distribution** of  $m$  at time  $t$ . To reduce the data size and complexity and ease computation, we suggest using time intervals. For example, an interval size ( $\delta$ ) of 10 minutes results in 144 intervals per day ( $\eta$ ). This creates a profile for a single day. Thus, for any day of week  $d$ , a profile consists of  $\eta$  location probability distributions (Eq. 3.1) denoted formally as

$$(3.2) \quad Profile_m^d = (P_m^{d,1}, P_m^{d,2}, \dots, P_m^{d,\eta})$$

where  $d \in D = \{Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday\}$ .

During the monitoring (test) phase, we use the learned profile (*location probability distributions*) to estimate the likelihood of each data record in the test set. To include some state information of where the device was previously, we consider a non-overlapping time window  $W$  previous to the current time instance to estimate the probability of current location  $l_c$  at current time instant  $t_c$ . Time window  $W$  is measured by number of minutes in this paper and is a parameter to our algorithm. Let  $w$  be the number of time instances (data records) in  $W$  (minutes). Let  $t_{c-w+1}$  to  $t_c$  be the  $w$  instances in time window  $W$ . We denote  $P_m^{d,W}(l_{c-w+1}, l_{c-w+2}, \dots, l_c)$  as  $P_m^{d,W}(l_W)$ , and approximate it using probability chain rule as:

$$(3.3) \quad P_m^{d,W}(l_W) = P_m^{d,t_c}(l_c|l_{c-1}, \dots, l_{c-w+1}) \times P_m^{d,t_{c-1}}(l_{c-1}|l_{c-2}, \dots, l_{c-w+1}) \times \dots \times P_m^{d,t_{c-w+1}}(l_{c-w+1})$$

The probability of a sequence of states is thus denoted as the product of probabilities of a state conditioned upon the previous states in the sequence. Storing all such probability values imposes an overhead and also increases the computational complexity.

**3.1 Markov based approach to detect location anomalies** For simplicity and because the independence assumption of the Naive Bayes classifier generally seems to work well [6], we assume independence between subsequent locations, resulting in  $P_m^{d,t_c}(l_c|l_{c-1}, \dots, l_{c-w+1}) = P_m^{d,t_c}(l_c)$ . The likelihood of mobile device  $m$  over the time window  $W$  is thus approximated as the product of the marginal probabilities:

$$(3.4) \quad P_m^{d,W}(l_W) = \prod_{i=c-w+1}^c P_m^{d,t_i}(l_i)$$

To avoid the underflow in multiplication, we use log likelihood instead:

$$(3.5) \quad \log(P_m^{d,W}(l_W)) = \sum_{i=c-w+1}^c \log(P_m^{d,t_i}(l_i))$$

For anomaly detection systems, an anomaly score denotes the degree of abnormality for the test data instance. An anomaly score can be calculated for  $m$  and location  $l_c$  using the negative log likelihood of aggregated location probability distribution over a window  $W$ :

$$(3.6) \quad \begin{aligned} AnomalyScore_m^{d,W}(l_W) &= -\log(P_m^{d,W}(l_W)) \\ &= -\sum_{i=c-w+1}^c \log(P_m^{d,t_i}(l_i)) \end{aligned}$$

The lower the likelihood of a location given the current context, the higher is the anomaly score.

The independence assumption of the Naive approach is not always valid, since to get to a specific location one typically traverses a fixed set of locations. The Naive approach can be considered as zero-order Markov Chain. The assumption is relaxed with Markov Chains of first order, described next.

In first order Markov Chain, the current state depends only on the previous state. This technique involves a probability transition matrix comprising of single step transition probabilities for all observed states. The location probability distribution of Eq. 3.1 is modified as

$$(3.7) \quad P_m^t(l_j|l_k) = \frac{P_m^t(l_j, l_k)}{P_m^t(l_k)} = \frac{freq_m^t(l_j, l_k)}{\sum_{k'} freq_m^t(l_k, l_{k'})}$$

For the Markov Chain,  $P_m^{d,t_c}(l_c|l_{c-1}, \dots, l_{c-w+1}) = P_m^{d,t_c}(l_c|l_{c-1})$ . The probability estimate for the sequence of traversed states of Eq. 3.3 is now revised as

$$(3.8) \quad P_m^{d,W}(l_W) = P_m^{d,t_c}(l_c|l_{c-1}) \times P_m^{d,t_{c-1}}(l_{c-1}|l_{c-2}) \times \dots \times P_m^{d,t_{c-w+1}}(l_{c-w+1})$$

Log likelihood is used to prevent underflow and the modified anomaly score is the negative log likelihood of aggregated location probability distribution:

$$(3.9) \quad \begin{aligned} AnomalyScore_m^{d,W}(l_W) &= -\log(P_m^{d,W}(l_W)) \\ &= -\log P_m^{d,t_{c-w+1}}(l_{c-w+1}) - \sum_{i=c-w+2}^c \log(P_m^{d,t_i}(l_i|l_{i-1})) \end{aligned}$$

We limit ourselves to lower order Markov Chains for ease of computation. Higher order Markov Chains can also be considered with higher time and space complexity.

#### 4 Temporal Location Anomaly Detection (TLAD)

In this section we present limitations of mobile data that maps location to cell id or access point, and the

anomaly detector presented in Sec. 3.1. We motivate and present *TLAD*, which comprises of a modified Markov based anomaly detector (Sec. 4.1), automated oscillation identification for better modeling (Sec. 4.2), and clustering day profiles for reduced model storage requirements (Sec. 4.3).

**4.1 Modified Markov anomaly detector for inaccurate frequency estimates** A 802.11 based laptop may be in the vicinity of multiple access points. The signal strength may vary due to distance and orientation, and may undergo attenuation due to obstructions such as walls. The device may connect to the access point with the highest signal strength from the current orientation, or the optimal one for load balancing. Thus, a physical location may correspond to multiple access points. Though we expect to aggregate information for connectivity with all access points corresponding to a single location over a period of time, the frequency information may not always be an accurate estimate. The Markov chain based approaches discussed in Sec. 3.1 flag valid but low frequency events as anomalous, resulting in higher false alarms. Next, we present an approach called *TLAD* (Temporal Location Anomaly Detection) to alleviate false alarms due to low probability events. *TLAD* only considers novel events in calculating the anomaly score. That is, no matter how frequent/likely an event has been observed, it is considered normal and has no contribution to the anomaly score. For example, a student may sometimes visit a professor during his office hours before the class. A low occurrence frequency may still flag the event as anomalous using naive and Markov Chain approaches, resulting in a false positive. But it would be deemed normal in *TLAD*. Thus, low frequency access points would also correspond to valid locations.

For *TLAD<sub>n</sub>* (or  $n^{th}$  order *TLAD*), only smoothed probability for novel event  $P_m^{d,W}(l_i|l_{i-1} \dots l_{i-n})$  is estimated. We investigate zero and first order *TLAD* in this paper, called *TLAD0* and *TLAD1* respectively. *TLAD0* maintains the subsequent location independence assumption of Naive approach, whereas the assumption is relaxed in *TLAD1* and the current state depends on the previous state. *TLAD* assigns negative logarithm of the novel probability estimate as the anomaly score for the current test instance. For *TLAD0*, it is

$$(4.10) \text{Score}0_i = \begin{cases} -\log(P_m^{d,t_i}(l_i)), & \text{if } \text{freq}_m^{d,t_i}(l_i) = 0 \\ 0, & \text{otherwise} \end{cases}$$

The anomaly score for  $l_c$  is aggregated over a time window  $W$  ( $l_W$ ) of  $w$  instances to ignore spurious

anomalies. For *TLAD0*, it is computed as:

$$(4.11) \text{AnomalyScore}_m^{d,W}(l_W) = \sum_{i=c-w+1}^c \text{Score}0_i$$

For *TLAD1*, which assumes that the current state is dependent on the previous, the anomaly score for the current test instance is calculated as:

$$(4.12) \text{Score}1_i = \begin{cases} -\log(P_m^{d,t_i}(l_i|l_{i-1})), & \text{if } \text{freq}_m^{d,t_i}(l_i, l_{i-1}) = 0 \\ 0, & \text{otherwise} \end{cases}$$

The anomaly score for  $l_c$  over window  $W$  ( $l_W$ ) is calculated as:

$$(4.13) \text{AnomalyScore}_m^{d,W}(l_W) = \text{Score}0_{c-w+1} + \sum_{i=c-w+2}^c \text{Score}1_i$$

An alarm is generated on exceeding a threshold.

**4.1.1 Smoothing probability values** Our techniques use probability estimate of novel events to score anomalies. Variance reduction techniques are required to compute the non-zero unobserved event probability estimate. In the event of a novel location, location probability distribution underestimates the probability of the new value by assigning it a value 0, resulting in an undefined anomaly score (Eqs. 3.6, 3.9, 4.11, 4.13). This problem of data sparseness is similar to the one in maximum likelihood estimator of a language model in natural language processing [5, 19].

Let  $s$  be the number of times a device was present at a specific location  $l$  in a given time interval. Thus,  $s = \text{freq}(l)$ . We denote  $n$  as the total frequency count  $= \sum_l \text{freq}(l)$ ; and  $r$  as the number of distinct locations for the device. Furthermore, let  $f_k$  be the number of distinct locations with frequency count equal to  $k$  at a given context. It can be observed that  $\sum_k f_k = r$  and  $\sum_k k f_k = n$ .

Witten and Bell [25] studied different schemes to deal with the *zero* frequency problem in adaptive statistical coding applied to text compression. They found the following estimate (due to [20]) to give the best results:

$$(4.14) P(l) = \begin{cases} \frac{s}{n+r}, & \text{if } l \text{ is observed} \\ \frac{r}{n+r}, & \text{otherwise} \end{cases}$$

This is referred to as *Method C* in the original paper. The rationale is to increase the probability of novel events with the number of distinct observations. For example, given two 10-integer sequences  $S_1 = \langle 1, 0, 0, 0, 0, 1, 1, 0, 0, 0 \rangle$  and  $S_2 =$

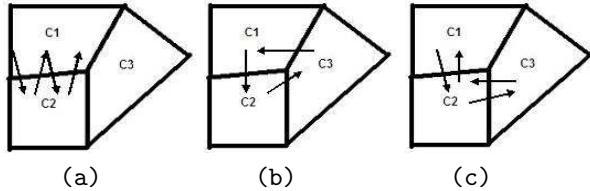


Figure 2: Example oscillations between adjacent cells.

$\langle 1, 2, 1, 0, 5, 7, -8, 12, 0, 9 \rangle$ ,  $S_2$  is more likely to encounter a novel subsequent value than  $S_1$ , since there is more randomness and variability in  $S_2$  than  $S_1$ . In Eq. 4.14, note that the sum of all smoothed probability values (including novel events) is unity.

## 4.2 Identifying oscillations in device locations

Given the nature of the data, there is one-to-many correspondence between the geographic co-ordinate and the location as captured by access point. A device can be within range to multiple access points at any instant. A wireless device, such as laptop, associates with an access point to gain access to the network. There may be frequent re-associations between multiple access points in a short amount of time. This may occur due to the strength signal variations based on the orientation of the device, or to balance the load across the multiple access points. Figs. 2a-c present some oscillating variations among adjacent cells  $C_1, C_2, C_3$ . In Fig. 2a, the user locations switches back and forth between  $C_1$  and  $C_2$ . Fig. 2b involves a cyclic transition through the three cells  $C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_1$ . In Fig. 2c, the transitions are across the three cells and then reversed in order -  $C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_2 \rightarrow C_1$ . Identification of oscillations is a hard problem as it entails making assumption about user intent. Consider two scenarios with syntactically similar transitions as in Fig.2a. Both scenarios represent the same sequence of re-associations across access points. But semantically they could be different. In one case, the user could be visiting the location, spending over an hour. On the other hand, it might be an oscillation as the time spent is very small, e.g. 5 seconds. Thus, it is imperative to consider the duration of time spent at the locations in addition to the transitions.

Due to the reason cited above, limited research has been pursued for the identification of oscillations. Hard coded rules have been suggested to identify oscillations [17], such as ones depicted in Figs. 2. There are two drawbacks to their approach. First, finding all applicable hard coded rules is a labor intensive and expensive process. In their case, the authors only used a couple of rules and ignored the remaining possibilities. Second, time spent with an access point is ignored.

As argued above, the time duration at an access point may be key to determining oscillations. More recently, detection of transition points between cell ids has been presented [4]. The algorithm requires "in transit" and "stationary" class labels provided by the user. This approach is hard to implement in a practical setting, as users cannot be expected to continuously provide labels on their own mobility trace. In this section we present an automated way to identify oscillations.

We represent device mobility as a directed graph, where the locations form the nodes and transitions between successive locations form the edges of the graph. Each edge is assigned a weight signifying the likelihood of an oscillation. We identify oscillations between each pair of connected nodes by estimating the correlation between the two locations occurring in succession based on the time duration at each node. The correlation is identified using three criteria, described next.

**4.2.1 Correlation test** Contingency tables are used to test the correlation between two variables. The cells in the table correspond of the joint frequency counts of their respective values. In our directed graph, we consider every node pair connected by an edge. Each pair of such locations has a contingency table. For two successive nodes (locations)  $A$  and  $B$  in the graph and the duration in each location  $\in \{short, long\}$ , we get a  $2 \times 2$  contingency table.  $A_{short}$  represents a short duration (e.g.  $\leq 10$  seconds) at location  $A$ , whereas  $B_{long}$  signifies long duration ( $> 10$  seconds) at location  $B$ .

We are interested in estimating if oscillation is more prevalent than non-oscillation between two successive locations. That is, if the event  $(A_{short}, B_{short})$  is more correlated than the remaining three cases between the same pair of locations. As the first criterion, we use chi-square ( $\chi^2$ ) test to check if the two variables  $A$  and  $B$  are correlated. The null hypothesis is that the two variables are independent and rejecting the null hypothesis means that the two variables are related and can predict the other. The expected counts for the variables are computed as joint probabilities of their respective values with the assumption of independence between the two variables. The test also uses a critical value, which dictates the confidence threshold for accepting or rejecting the hypothesis, i.e. whether it is an oscillation or not. We use a critical value of 95%. The null hypothesis of the chi-square test is that  $A$  and  $B$  are independent. If it is rejected (at 95% significance level),  $A$  and  $B$  are dependent with statistical significance.

Chi-square test suggests there is significant correlation between two variables, but it doesn't tell how the four pairs of values are correlated. Consider the follow-

ing notation:

$$(4.15) \quad I(X, Y) = P(X, Y) / P(X)P(Y)$$

Our second criterion for correlation between variables  $X$  and  $Y$  uses the following relation:

$$(4.16) \quad I(X, Y) > 1,$$

Eq. 4.16 stresses on correlation between  $X$  and  $Y$  since the joint probability  $(X, Y)$  is more likely than the joint probability calculated by assuming  $X$  and  $Y$  are independent. Thus,  $(A_{short}, B_{short})$  are correlated if

$$(4.17) \quad I(A_{short}, B_{short}) > 1$$

Eq. 4.17 only considers one of the four cells in the contingency table. However, correlation could be attributed to the other cells as well. We desire  $(A_{short}, B_{short})$  to have the largest contribution. The third criterion for correlation between successive locations  $A$  and  $B$  is that  $I(A_{short}, B_{short})$  should be greater than each of the remaining  $\{I(A_{short}, B_{long}), I(A_{long}, B_{short}), I(A_{long}, B_{long})\}$ .

Thus, all the three criteria need to be satisfied for two successive locations to be correlated. Together, these criteria mean that the two successive locations are correlated, and the maximum contribution comes from the short durations at these locations. For efficiency, the second criterion can be checked first, then the third, followed by the chi-square test.

Once the oscillating locations are identified, we group them together into locales. The locale comprises of only one location if there is no evidence of oscillation. Once formed, we use the locales as features in place of locations with the anomaly detection algorithm described in Sec. 4.1.

**4.3 Clustering day profiles** Our anomaly detectors create a model for each day of the week, as represented in Eq. 3.2. Typically, a high volume of data is tracked by such a system in real time. Thus, small models are desired. Merging similar day profiles together reduces the storage overhead. For example, an executive assistant with a relatively same schedule from Monday to Friday may result in a succinct model for all weekdays. Similarly, a professor teaching same classes on Tuesday and Thursday can have a similar model for those two days. Trajectory clustering has been proposed in [16], though the temporal information is not used.

Each individual day profile is an object during clustering, and each day has the same representation as the day profile in Eq. 3.2. That is, for  $\eta$  intervals per day and  $N$  locations per interval, each object has  $\eta \times N \times N$  location probability distributions. The distance

between a pair of location distributions is estimated using Kullback Leibler divergence, which is a measure of the relative entropy between two distributions. Kullback Leibler divergence  $KL$  between location probability distributions  $p_1$  and  $p_2$  is calculated as:

$$(4.18) \quad KL^t(p_1 || p_2) = \sum_i p_1^t(i) \log \frac{p_1^t(i)}{p_2^t(i)}$$

Since Kullback Leibler divergence is not symmetric [ $KL^t(p_1 || p_2) \neq KL^t(p_2 || p_1)$ ], it cannot be used as-is to measure distance for clustering similar location distributions. Thus, we define distance  $\Delta^t(p_1, p_2)$  between  $p_1$  and  $p_2$  as:

$$(4.19) \quad \Delta^t(p_1, p_2) = KL^t(p_1 || p_2) + KL^t(p_2 || p_1)$$

The  $\Delta$  value is normalized in the computation above. It can be noted that  $\Delta$  is symmetric for a given distribution pair  $p_1$  and  $p_2$ . The total distance between a pair of profiles (objects)  $Profile_m^{d_1}$  and  $Profile_m^{d_2}$  for days  $d_1$  and  $d_2$  respectively is computed using Euclidean distance:

$$(4.20) \quad \mathcal{D}(Profile_m^{d_1}, Profile_m^{d_2}) = \sqrt{\sum_{i=1}^{\eta} \Delta^i(p_i^{d_1}, p_i^{d_2})^2}$$

---

**Algorithm 1** CLUSTERING DAY PROFILES

---

**Require:**  $\bigcup_{d \in D} Profile_m^d$ , as defined in Eq. 3.2

**Ensure:**  $\bigcup_{d' \subseteq D} Profile_m^{d'}$

- 1: Compute pair-wise cluster distances using Eq. 4.20
  - 2: **while** (stopping criterion not met) **do**
  - 3:   Identify clusters with minimum pairwise distance
  - 4:   Merge minimum distance profiles into  $Profile_m^{d'}$
  - 5:   Recompute pair-wise cluster distances with merged cluster
  - 6: **end while**
- 

Agglomerative hierarchical clustering is a bottom-up approach to clustering [12]. It initially starts with multiple clusters at the bottom, and repeatedly merges the most similar clusters together until a terminating condition is met. We present an agglomerative hierarchical clustering technique to create succinct learned models. In our problem setting, all day profiles are disjoint clusters input to the clustering algorithm, as shown in Algorithm 1. The distance function in Eq. 4.20 is used to compute distance between all cluster pairs in Step 1. Steps 2–6 represent the merging iterations for the bottom-up approach. During each iteration, two most similar clusters (profiles) are merged (Step 3). Cluster similarity is measured using the distance function in

Eq. 4.20. Distance between exactly same profiles is zero. Merging entails combining the location probability distributions for the profiles with the minimum distance (Step 4). Individual location probability distributions are then replaced by the merged profile. Associated cluster distances are recomputed in Step 5. The stopping criterion for the algorithm (in Step 2) can be the number of desired clusters  $k$  or a distance threshold  $\theta$  for merging most similar clusters.

**4.3.1 Space complexity analysis of learned model** Considering a potentially large number of mobile users, a small user model is desired for anomaly detection. Our clustering approach merges similar location probability distributions and creates a succinct learned model. For *TLAD0*, the space complexity of the stored model is  $O(\eta CN)$ , where  $\eta$  is the number of time intervals per day,  $C$  is the number of clusters, and  $N$  is the number of locales per interval. For *TLAD1*, the space complexity is  $O(\eta CN^2)$ . But the probability matrix is generally sparse and can be implemented more efficiently. In the worst case,  $C = C_{max}$ , which is the same as no clustering. In our case,  $C_{max} = 7$  (number of days per week).

## 5 Results

This section reports some empirical evaluation of our proposed algorithm.

**5.1 Experimental Data and Procedures** To evaluate and compare the spatio-temporal anomaly detection techniques, we used publicly available real WLAN data set. It comprises of syslog records collected at Dartmouth College campus spanning over three years between April 1, 2001 and June 30, 2004 [13]. The data set contains the following information: MAC address, access point name, and associated timestamp. The data is gathered from 163 different buildings belonging to six different categories – ACA (academic), ADM (administrative), ATH (athletic), LIB (library), RES (residential) and SOC (social). The building where a user spends the most time is deemed as his "home" building. We maintain a distinction between *same* and *different* "home" building – users from *same* "home" building might be harder to distinguish from self as they generally frequent the same building; whereas users from *different* "home" building might more dissimilar from self as they are expected to visit other buildings more often than their "home". A sampling rate of 1 second/instance was used in our experiments.

From the entire data sets, we selected users with at least one week of training data. This resulted in 6,688 WLAN users from the 163 buildings. Data distribution

Table 1: Data sets.

Data set:	ACA	ADM	ATH	LIB	RES	SOC
No. bldgs:	31	25	9	5	86	7
No. users:	1321	332	162	456	4252	162

Table 2: Confusion matrix.

Actual	Prediction	
	Unauthorized user	Authorized user
Unauthorized user	True Positive	False Negative
Authorized user	False Positive	True Negative

is presented in Table 1. Disjoint training and test sets were created for each user. Ideally, we would like to train on authorized user  $A$  and test against data for an unauthorized user  $B$  trying to impersonate  $A$ . However, we are not aware of any such publicly available data. In the absence of explicit labels for bad behavior, we adopt the standard approach of simulating unauthorized usage by replaying the mobility trend of a randomly picked user on the authorized user's device. That is, given a trained model for a mobile device, test data from a randomly selected user was used to approximate behavior of unauthorized user. The confusion matrix is presented in Table 2. Data sample from device  $B$  is tested against the learned model for  $A$ . Any alarm against model  $A$  is a true positive (successful detection) and against model  $B$  is a false alarm. Thus, we tested each user model on their own test data set and one *simulated* unauthorized user. For the Dartmouth WLAN data, we also distinguished between *same* and *different* "home" building, as mentioned above. Hence we conducted the experiment twice for every WLAN user, picking the unauthorized user from the *same* and *different* "home" buildings respectively.

Time interval  $\delta$  (Sec. 3) is a parameter to our techniques. Coarse-grained values reduce sparseness, but tend to include multiple contexts. On the other hand, fine-grained values are focused on specific context but exacerbate the issue of data sparseness, thereby increasing the number of false alarms. Larger interval sizes also accommodate spurious anomalies better than smaller intervals. We present results with  $\delta=60$  minutes, though values from 15-60 minutes yielded similar results. The time window  $W$  (Eq. 3.4) is another parameter for our techniques. Small  $W$  value flags anomalies at an early stage thereby minimizing loss, hence we chose  $W=10$  minutes. For oscillation identification (Sec. 4.2), any duration over 60 seconds is considered long. A conservative distance threshold  $\theta=0.5$  (Sec. 4.3) was used for clustering.

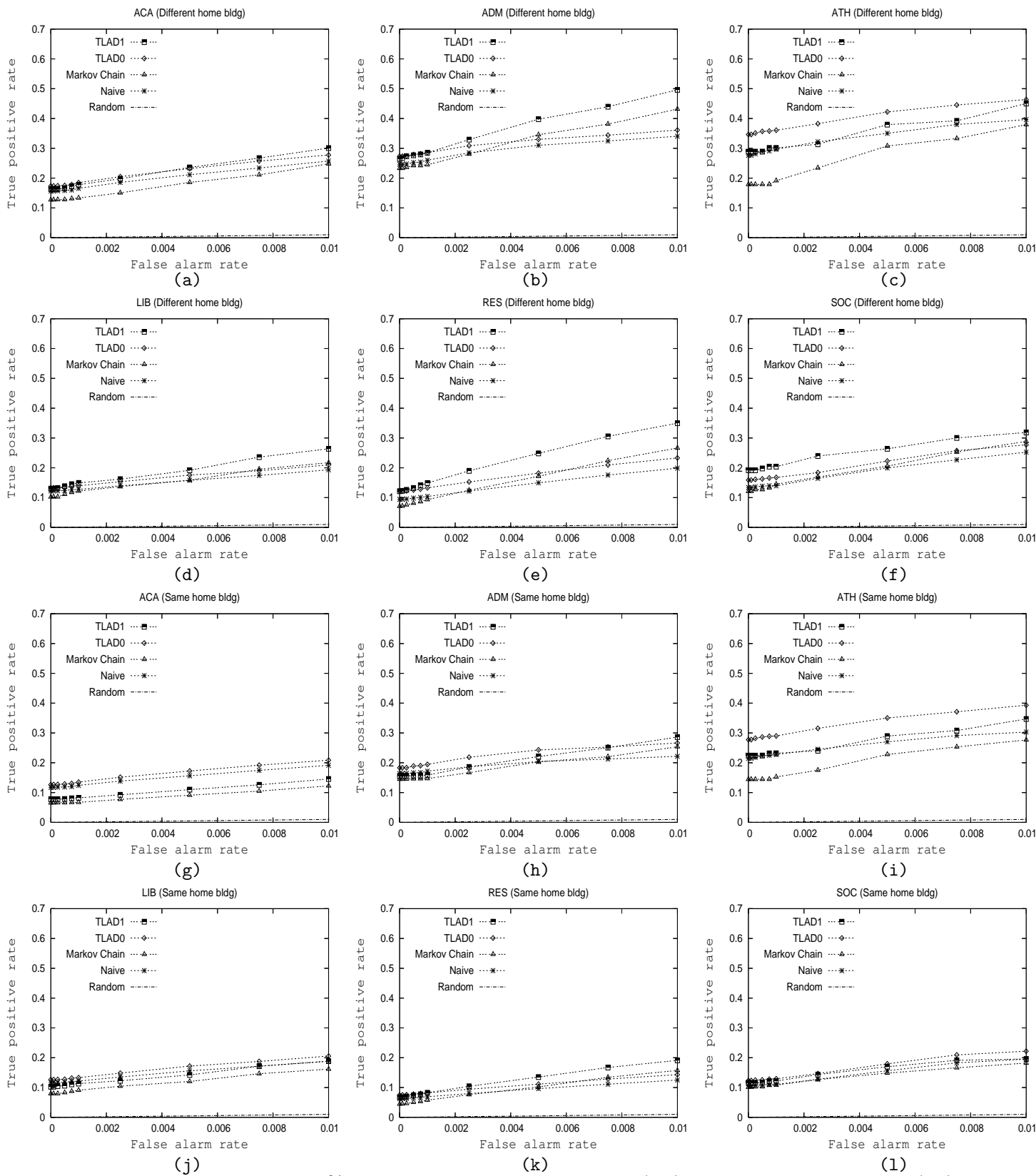


Figure 3: ROC curves up to 1% FAR for "different" home building (a-f) and "same" home building (g-l).

Table 3: Area under ROC curve (in %) up to 0.1% and 1% FAR. Random detector has area = 0.05% (at 0.1% FAR), 0.5% (at 1% FAR). Values for *TLAD* greater than both Naive and Markov are emphasized.

"Different" home building								
Data set	0.1% FAR				1% FAR			
	<i>Naive</i>	<i>Markov</i>	<i>TLAD0</i>	<i>TLAD1</i>	<i>Naive</i>	<i>Markov</i>	<i>TLAD0</i>	<i>TLAD1</i>
ACA	15.92	12.92	<b>17.67</b>	<b>16.81</b>	20.90	18.32	<b>22.95</b>	<b>23.32</b>
ADM	25.16	23.94	<b>27.86</b>	<b>27.60</b>	30.29	33.40	32.47	<b>38.61</b>
ATH	28.71	18.04	<b>35.46</b>	<b>29.35</b>	34.74	28.68	<b>41.39</b>	<b>36.38</b>
LIB	12.46	11.07	<b>13.54</b>	<b>13.87</b>	15.72	16.31	<b>17.23</b>	<b>19.73</b>
RES	9.84	8.12	<b>12.59</b>	<b>13.36</b>	14.82	17.22	<b>18.01</b>	<b>24.47</b>
SOC	13.75	12.96	<b>16.32</b>	<b>19.78</b>	19.53	20.82	<b>22.00</b>	<b>26.38</b>
"Same" home building								
Data set	0.1% FAR				1% FAR			
	<i>Naive</i>	<i>Markov</i>	<i>TLAD0</i>	<i>TLAD1</i>	<i>Naive</i>	<i>Markov</i>	<i>TLAD0</i>	<i>TLAD1</i>
ACA	11.95	6.62	<b>12.95</b>	7.93	15.57	9.14	<b>17.05</b>	10.98
ADM	16.64	14.58	<b>18.78</b>	15.66	19.86	19.66	<b>23.43</b>	<b>21.84</b>
ATH	22.18	14.54	<b>28.50</b>	<b>28.09</b>	26.65	21.58	<b>34.28</b>	<b>34.28</b>
LIB	11.84	8.37	<b>12.90</b>	10.74	15.34	12.30	<b>16.81</b>	14.58
RES	6.56	5.02	<b>7.79</b>	<b>7.37</b>	9.52	10.34	<b>11.07</b>	<b>13.38</b>
SOC	10.85	10.43	<b>12.61</b>	<b>11.81</b>	15.38	14.52	<b>17.66</b>	<b>16.44</b>

**5.2 Evaluation Criteria** Anomaly detectors with known attack labels consider detections associated with a timestamp or any alarm in the entire data sequence. Since we simulate unauthorized behavior, we used a strict evaluation criterion that **all** event windows from an unauthorized user account for the algorithm accuracy. Correctly detecting malicious usage for a single event window does not count as success for all other intervals from the same user. Thus, each event interval is evaluated independently of the others, thereby making it harder for the anomaly detectors. Computer security techniques are typically evaluated using a Receiver Operator Characteristic (**ROC**) curve that plots the rate of correct anomalies detected (i.e. different device) alongside the false alarm (i.e. same device) rate. The area under the ROC curve (**AUC**) is also calculated. A larger AUC value is desired as it is representative of the total percentage of true positives detected at the cost of varied false alarm rates (**FAR**). In addition to model accuracy, we compared the time requirements and storage overhead.

**5.3 Comparison of Accuracy** We present an algorithm called Temporal Location Anomaly Detector (*TLAD*) to detect abnormalities in mobility traces. It is a modified Markov chain based algorithm which scores anomalies for novel locations (Sec. 4.1), removes oscillations based on correlation tests (Sec. 4.2), and clusters location probability distributions together (Sec. 4.3).

We evaluated and compared the four anomaly

detection techniques – Naive, Markov, *TLAD0*, and *TLAD1* – on the 802.11 WLAN data sets. The drawback of anomaly detection is the generation of false alarms, since not all abnormalities are due to misuse. Thus, we focus on small false alarm rates (*FAR*). ROC curves up to 1% FAR are presented in Fig. 3. Figs. 3a-f shows results for *different* "home" building, and Figs. 3g-l for *same* "home" building. The random detector has the same FAR and true positive rate for any threshold (represented by the diagonal  $x = y$  in the ROC curve). Accuracy, measured in terms of respective areas under the ROC curve (*AUC*), is listed in Table 3 for  $FAR \in \{0.001, 0.01, 1\}$ . The ROC curves suggest that *TLAD* generally had higher accuracy than Naive and Markov chain, though there was no single technique that outperformed the others. Table 3 shows that *TLAD0* had *AUC* greater than Naive and Markov chain in all cases except ADM at 1% FAR, whereas *TLAD1* had higher *AUC* for "different" home users but only in some cases for "same" home. In terms of number of times greater *AUC* than traditional approaches in the table, *TLAD0* is the best technique. As expected, it was harder to detect anomalies from the "same" than "different" home building.

We applied the paired *t*-test to check if the improvements in accuracy were statistically significant. We paired *TLAD0* with Naive and *TLAD1* with Markov chain to obtain the confidence intervals. We consider accuracy improvement with confidence level lower than 90% as not statistically significant. For the "differ-

Table 4: Accuracy comparison with (W/) and without (W/O) oscillation detection (OD). Higher AUC value in each pair is highlighted.

"Different" home building at 0.1% FAR								
	<i>Naive</i>		<i>Markov</i>		<i>TLAD0</i>		<i>TLAD1</i>	
	<i>W/O OD</i>	<i>W/ OD</i>	<i>W/O OD</i>	<i>W/ OD</i>	<i>W/O OD</i>	<i>W/ OD</i>	<i>W/O OD</i>	<i>W/ OD</i>
ACA	15.92	<b>17.46</b>	12.92	<b>16.74</b>	16.20	<b>17.67</b>	12.92	<b>16.81</b>
ADM	25.16	<b>27.46</b>	23.94	<b>27.40</b>	25.56	<b>27.86</b>	23.98	<b>27.60</b>
ATH	28.71	<b>35.41</b>	18.04	<b>29.35</b>	28.76	<b>35.46</b>	18.04	<b>29.35</b>
LIB	12.46	<b>13.05</b>	11.07	<b>13.87</b>	12.79	<b>13.54</b>	11.07	<b>13.87</b>
RES	9.84	<b>12.55</b>	8.12	<b>13.31</b>	10.77	<b>12.59</b>	8.13	<b>13.36</b>
SOC	13.75	<b>16.17</b>	12.96	<b>19.78</b>	13.91	<b>16.32</b>	12.96	<b>19.78</b>
"Different" home building at 1% FAR								
	<i>Naive</i>		<i>Markov</i>		<i>TLAD0</i>		<i>TLAD1</i>	
	<i>W/O OD</i>	<i>W/ OD</i>	<i>W/O OD</i>	<i>W/ OD</i>	<i>W/O OD</i>	<i>W/ OD</i>	<i>W/O OD</i>	<i>W/ OD</i>
ACA	20.90	<b>22.73</b>	18.32	<b>23.30</b>	21.12	<b>22.95</b>	18.34	<b>23.32</b>
ADM	30.29	<b>32.25</b>	33.40	<b>38.37</b>	30.61	<b>32.47</b>	33.40	<b>38.61</b>
ATH	34.74	<b>41.26</b>	28.68	<b>36.38</b>	34.87	<b>41.39</b>	24.68	<b>36.38</b>
LIB	15.72	<b>16.75</b>	16.31	<b>19.73</b>	16.08	<b>17.23</b>	16.31	<b>19.73</b>
RES	14.82	<b>17.93</b>	17.22	<b>24.43</b>	15.94	<b>18.01</b>	17.24	<b>24.47</b>
SOC	19.53	<b>21.96</b>	20.82	<b>26.37</b>	19.72	<b>22.00</b>	20.83	<b>26.38</b>
"Same" home building at 0.1% FAR								
	<i>Naive</i>		<i>Markov</i>		<i>TLAD0</i>		<i>TLAD1</i>	
	<i>W/O OD</i>	<i>W/ OD</i>	<i>W/O OD</i>	<i>W/ OD</i>	<i>W/O OD</i>	<i>W/ OD</i>	<i>W/O OD</i>	<i>W/ OD</i>
ACA	11.95	<b>12.81</b>	6.62	<b>7.86</b>	11.99	<b>12.95</b>	6.61	<b>7.93</b>
ADM	16.64	<b>17.25</b>	14.58	<b>15.38</b>	17.50	<b>18.78</b>	14.59	<b>15.66</b>
ATH	22.18	<b>26.46</b>	14.54	<b>21.97</b>	22.92	<b>28.50</b>	15.06	<b>22.69</b>
LIB	11.84	<b>12.68</b>	8.37	<b>10.74</b>	12.01	<b>12.90</b>	8.37	<b>14.58</b>
RES	6.56	<b>7.22</b>	5.02	<b>7.30</b>	6.82	<b>7.79</b>	5.03	<b>7.37</b>
SOC	10.85	<b>11.93</b>	10.43	<b>11.82</b>	10.91	<b>12.61</b>	10.43	<b>11.81</b>
"Same" home building at 1% FAR								
	<i>Naive</i>		<i>Markov</i>		<i>TLAD0</i>		<i>TLAD1</i>	
	<i>W/O OD</i>	<i>W/ OD</i>	<i>W/O OD</i>	<i>W/ OD</i>	<i>W/O OD</i>	<i>W/ OD</i>	<i>W/O OD</i>	<i>W/ OD</i>
ACA	15.57	<b>16.33</b>	9.14	<b>10.79</b>	15.78	<b>17.05</b>	9.17	<b>10.98</b>
ADM	19.86	<b>20.30</b>	19.66	<b>20.93</b>	21.54	<b>23.43</b>	20.10	<b>21.84</b>
ATH	26.65	<b>30.92</b>	21.58	<b>26.93</b>	28.18	<b>34.28</b>	22.58	<b>28.09</b>
LIB	15.34	<b>16.30</b>	12.30	<b>14.55</b>	15.78	<b>16.81</b>	12.33	<b>14.58</b>
RES	9.52	<b>10.07</b>	10.34	<b>13.04</b>	9.99	<b>11.07</b>	10.42	<b>13.38</b>
SOC	15.38	<b>16.50</b>	14.52	<b>16.12</b>	15.72	<b>17.66</b>	14.55	<b>16.44</b>

Table 5: Fraction of users with  $AUC \geq 90\%$  at 1% FAR.

"Different" home building				
	<i>Naive</i>	<i>TLAD0</i>	<i>Markov</i>	<i>TLAD1</i>
ACA	8.77	<b>11.28</b>	13.24	<b>17.56</b>
ADM	17.17	<b>20.30</b>	24.10	<b>27.88</b>
ATH	19.14	<b>25.93</b>	19.14	<b>30.25</b>
LIB	6.80	<b>8.11</b>	11.84	<b>14.04</b>
RES	2.75	<b>6.89</b>	8.91	<b>14.06</b>
SOC	7.88	<b>10.30</b>	14.55	<b>20.61</b>
"Same" home building				
	<i>Naive</i>	<i>TLAD0</i>	<i>Markov</i>	<i>TLAD1</i>
ACA	4.40	<b>5.31</b>	3.49	<b>4.55</b>
ADM	6.67	<b>10.00</b>	6.97	<b>8.79</b>
ATH	6.17	<b>14.82</b>	8.03	<b>11.11</b>
LIB	5.48	<b>6.14</b>	7.02	<b>8.77</b>
RES	2.00	<b>2.19</b>	2.62	<b>4.03</b>
SOC	1.82	<b>4.85</b>	4.85	<b>4.85</b>

ent" home data sets, results show that improvement in accuracy for  $FAR=\{0.1\%, 1\%\}$  for both *TLAD0* and *TLAD1* is statistically significant at the 99% level. For "same" home users, there is a statistically significant increase in accuracy for *TLAD0* at the 95% level for  $FAR=\{0.1\%, 1\%\}$ . For *TLAD1*, improvement is not significant at  $FAR=0.1\%$ , but it is significant at 90% level for  $FAR=1\%$ .

Different users behave differently. Some are more predictable than others. We studied the trend in the fraction of users having  $AUC \geq 90\%$  at 1% FAR. Results, listed in Table 5, show that *TLAD* increases the percentage of users with a high accuracy. The maximum absolute increase was in the case of ATH ("different" home building), where *TLAD1* had over 30% users with  $AUC \geq 90\%$ , compared to approx. 19% users for Markov chain. An interesting observation from this table is that the percentage of users with highest accuracy is more for *TLAD1* and Markov chain than their zero-order counterparts. This table is also consistent with our expectation that "same" home building users are harder to detect than "different" home building users.

**Effect of oscillation detection on accuracy** In Sec. 4.2 we presented a statistical approach to identify oscillations in mobility traces. We evaluated the efficacy of the approach for detecting misuse on the various data sets. Results are presented in Table 4, with each anomaly detector evaluated with and without oscillation detection. The table illustrates the benefit of oscillation detection – it results in higher accuracy in all cases. Oscillation detection even boosts the accuracy for Naive and Markov chain, making them comparable to

Table 6: Average number of clusters/user without (W/O) and with (W/) oscillation detection (OD).

	<i>ACA</i>	<i>ADM</i>	<i>ATH</i>	<i>LIB</i>	<i>RES</i>	<i>SOC</i>
W/O OD	6.41	5.65	5.66	6.48	5.83	6.25
W/ OD	6.31	5.53	5.42	6.41	5.74	6.06

*TLAD0* and *TLAD1* respectively. But the benefit of *TLAD* is that it eliminates the need to store individual location frequencies, without any deterioration in accuracy. We also applied the paired *t*-test to check if the improvements in accuracy due to oscillation detection were statistically significant. For the "different" home users, accuracy of zero order techniques (Naive and *TLAD0*) was statistically significant at 95% level at both 0.1% and 1% FAR. For "same" home users, the improvement in accuracy for these techniques was significant at the 90% level for both FAR. For the first order techniques – Markov chain and *TLAD1* – the improvement was statistically significant at 99% level for  $FAR=\{0.1\%, 1\%\}$  for "different" home users. For "same" home users, these techniques fared at 95% and 99% for  $FAR=0.1\%$  and 1% respectively.

## 5.4 Time and space requirements

### 5.4.1 Time requirements for training and testing

For anomaly detection system to be effective, it should be able to detect misuse in real-time. For completeness, we computed time requirements for model creation (training) as well as testing for the anomaly detection techniques. Experiments were performed on a 2GHz Pentium M processor, 1 GB RAM PC running Windows XP. Training rate was less than 1  $\mu$ second/instance, and testing rates were below 0.5  $\mu$ second/instance for all data sets. Our techniques incur low computational overhead, making them reasonable for an online system.

### 5.4.2 Reduced storage for trained model

We proposed context clustering using an agglomerative hierarchical approach with Kullback-Leibler-based distance metric in Sec. 4.3. Clustering collapses similar days into a single cluster, thereby reducing the space requirements. No clustering results in 7 clusters. Average number of clusters per user for the data sets are listed in Table 6. The reduction in model size due to clustering is approximately 7–19% for TLAD without oscillation detection. Since oscillation detection combines locations into locales, we expect the number of clusters to be reduced further. This is supported by the results, which show a model size reduction of 8–23% for

TLAD with oscillation detection.

## 6 Conclusions

To alleviate the problem of MAC spoofing, we presented an automated technique called TLAD (temporal location anomaly detection). TLAD creates smoothed stochastic user models from mobility traces and detects abnormalities, potentially due to misuse. We presented a statistical technique to identify oscillations in mobility traces. Such locations are grouped into locales in TLAD for improved accuracy on real WLAN data sets. TLAD0 had higher *AUC* than Naive and Markov approaches at  $\leq 1\%$  FAR for most data sets, but TLAD1 had higher fraction of users with high *AUC* than TLAD0. Results show that TLAD and other temporal approaches are more effective for users who are more dissimilar in mobility patterns, such as ones from "other" home buildings. We also presented a Kullback-Leibler divergence based agglomerative hierarchical clustering to merge profiles for a smaller model, obtaining a reduction of 8–23% in our experiments. The test time requirement for our technique was less than 0.5  $\mu$ second per instance, making it viable for online usage. We are currently experimenting with our technique on a cellular phone data set. We also intend to study the efficacy of coarser contexts such as beginning/mid/end of month, week and day. Since TLAD demonstrated high accuracy for users with low entropy, it can aid and complement other techniques, such as collision detection [7] and multi-modal biometric systems, to detect spoofing attacks.

## References

- [1] P. Bahl and V. N. Padmanabhan, *RADAR: An In-building RF-based User Location and Tracking System*, IEEE Infocom (2000).
- [2] A. Balachandran, G. M. Voelker, P. Bahl and P. V. Rangan, *Characterizing user behavior and network performance in a public WLAN*, IEEE WMCSA (2002).
- [3] M. Balazinka and P. Castro, *Characterizing mobility and network usage in a corporate wireless local-area network*, MobiSys (2003).
- [4] N. Bila, J. Cao, R. Dinoff, T. K. Ho, R. Hull, B. Kumar, and P. Santos, *Mobile user profile acquisition through network observables and explicit user queries*, Int'l. Conf. Mobile Data Management (2008), pp. 98–107.
- [5] S. F. Chen and J. Goodman, *An Empirical Study of Smoothing Techniques for Language Modeling*, Tech. Report TR-10-98, Harvard University (1998).
- [6] P. Domingos and M. Pazzani, *On the Optimality of the Simple Bayesian Classifier under Zero-One Loss*, Machine Learning, 29 (1997), pp. 103–130.
- [7] T. Fawcett and F. Provost, *Adaptive Fraud Detection*, Data Mining and Knowledge Discovery, 1, Kluwer Academic Publishers (1997), pp. 291–316.
- [8] J. A. Flanagan, J. Mäntyjarvi, and J. Himberg, *Unsupervised clustering of symbol strings and context recognition*, IEEE ICDM (2002).
- [9] R. Flickenger, *Wireless Hacks*, O'Reilly (2003).
- [10] N. Friedman and Y. Singer, *Efficient Bayesian Parameter Estimation in Large Discrete Domains*, NIPS (1998).
- [11] J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmäki, and H. T.T. Toivonen, *Time series segmentation for context recognition in mobile devices*, ICDM (2001).
- [12] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall (1988).
- [13] D. Kotz, T. Henderson, and I. Abyzov, *CRAW-DAD trace set dartmouth/campus/movement (v.2005-03-08)*, Dartmouth College (2005).
- [14] K. Laasonen, M. Raento, and H. Toivonen, *Adaptive on-device location recognition*, Pervasive Computing: Second International Conference (2004), pp. 287–304.
- [15] A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava and V. Kumar. *A comparative study of anomaly detection schemes in network intrusion detection*, SDM (2003).
- [16] J. G. Lee, J. Han and K. Y. Whang, *Trajectory Clustering: A Partition-and-Group Framework*, ACM SIGMOD (2007).
- [17] J. K. Lee and J. C. Hou, *Modeling steady-state and transient behaviors of user mobility: formulation, analysis, and application*, ACM MobiHoc (2006).
- [18] X. Li, J. Han, S. Kim, and H. Gonzalez, *ROAM: Rule- and Motif-Based Anomaly Detection in Massive Moving Object Data Sets*, SDM (2007).
- [19] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press (1999).
- [20] A. Moffat, *A Note on the PPM Data Compression Algorithm*, Tech. Report 88/7, University of Melbourne, Australia (1988).
- [21] *Netstumbler*, www.stumbler.net.
- [22] C. Peikair and S. Fogie, *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*, Addison-Wesley (2003).
- [23] J. Shavlik and M. Shavlik, *Selection, combination, and evaluation of effective software sensors for detecting abnormal computer usage*, SIGKDD (2004).
- [24] C. Welch and S. Lathrop, *A survey of 802.11a wireless security threats and security mechanisms*, Technical Report, United States Military Academy (2003).
- [25] I. H. Witten and T. C. Bell, *The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression*, IEEE Trans. Information Theory, 37:4 (1991), pp. 1085–1094.
- [26] J. Yin, X. Chai, and Q. Yang, *High Level Goal Recognition in Wireless LAN*, AAAI (2004).
- [27] M. A. Youssef, A. Agrawala, and A. U. Shankar, *WLAN Location Determination via Clustering and Probability Distributions*, IEEE PerCom (2003).