Spatio-temporal Anomaly Detection for Mobile Devices

Gaurav Tandon
Department of Computer Sciences
Florida Institute of Technology
Melbourne, Florida
gtandon@cs.fit.edu

Philip K. Chan
Department of Computer Sciences
Florida Institute of Technology
Melbourne, Florida
pkc@cs.fit.edu

Abstract

With the increase in popularity of mobile devices, there has been a significant rise in mobile related security problems. The biggest threat for a mobile subscriber is lost or stolen device, which can lead to confidential data leakage, identity theft, misuse, impersonation, and high service charges. A significant amount of time may elapse between losing a device and disabling it through the service provider, during which an unauthorized malicious user may gain access and incur severe damage. We propose a probabilistic approach to spatio-temporal anomaly detection and evaluate smoothing techniques for sparse data. Our approach outperforms Markov Chain in experiments with a mobile phone dataset comprising over 500,000 hours of real data. Results indicate that our approach can effectively and efficiently detect device abnormalities for location, time, or both.

1. Introduction

Mobile devices like cellular phones, smartphones and Pocket PCs are rapidly gaining popularity worldwide, featuring increased storage capacity, greater application support, and reduced price. Smartphones and PDAs are increasingly used by employees to connect to corporate networks, retrieve and store important company data. Data stored on a device is often not encrypted and no security mechanism is adopted by the user. Losing a phone or having it stolen is currently the biggest risk that a mobile consumer faces [8, 16]. Over 55 million mobile phones were estimated lost worldwide in 2006 alone and projections for the total number over the subsequent five years exceed 500 million handsets [16]. A lost device may contain personal and confidential company data that can be accessed by an unauthorized user. The device can be misused, leading to identity theft, data leakage (e.g. social security numbers of employees or customers), impersonation, and high service charges to the subscriber. Although one can notify the loss to the network operator and have the device disabled or data wiped out [1], a significant amount of time may have elapsed. The unauthorized user may have already succeeded in his malicious intent, incurring significant losses to the individual, the enterprise and potentially numerous other people whose personal infomation was compromised.

A mobile phone is a wearable sensor that people generally carry with them all the time. With this assumption, we propose a spatial temporal learning based anomaly detection approach to tackle the problem of lost or stolen mobile device. Our approach uses the location of a subscriber at different time intervals over a period of time and generates a probabilistic user (subscriber) model. We assume that the device is used by a single user, and will interchangeably use the terms *user* and *device*. In this paper, we aim at determing spatial temporal patterns for a given user. The subscriber would generally be at the same location at a given time. For example, an employee would typically be in his office from 9 a.m. to 6 p.m. during the weekdays, and at home most of the remaining time. Even the weekends could generate potential patterns like weekly kids' soccer games, and grocery shopping. A learning algorithm can be used to learn these contexts and then make predictions to determine any anomalies. A lost or stolen device could result in contextual anomalies for the given model, either in terms of location, or time, or both. The authors, to their best knowledge, are unaware of any prior work to tackle the problem of lost and stolen mobile phones in automated manner.

We present the problem as one dealing with sparse data, which has been researched in the natural language processing domain and applied to language modeling, text compression and information retrieval problems. We evaluate some smoothing methods on data set comprising over 500,000 hours of real data collected from 90 users. We present STAD (Spatio Temporal Anomaly Detection) and compare it with Markov Chains using multiple evaluation criteria and parameter settings. Results show that our proposed schemes are effective in capturing spatial temporal

anomalies introduced by an unauthorized user, and incur minimal computational overhead for online usage. We also demonstrate the effectiveness of feedback for model update to take into account concept drift and restrict the generation of false alarms.

This paper is organized as follows: Section 2 presents existing literature from relevant domains. In Section 3 describes our proposed technique and discusses the smoothing methods used. We present evaluation and comparison of the various techniques on a real data set in Section 4. Parameter study and time requirements are also presented. Key findings are compiled in Section 5.

2. Related Work

The relevant work is discussed under three categories: context-aware mobile computing; anomaly detection systems; and smoothing techniques for sparse data.

Context-aware mobile computing has been an active area of research in the last decade. Overview and earlier work is reviewed in [9, 5]. Active badge system [35] used infra-red based badges to determine the location of an employee and forward his call to the nearest telephone. A radio frequency system was proposed in [4] to detect user location within a building. Signal strengths from different base stations were triangulated to obtain that information. Signal strength probability distributions and location clustering for infrastructure LANs was presented in [38], whereas a framework for plan recognition in an indoor RF-based wireless network was proposed in [37]. Unsupervised clustering and classification of contexts obtained from multiple sensors has been studied in [21, 12, 20], and fuzzy low level contextual information was segmented to obtain high level contexts in [17]. Most of context-aware research has stressed on context extraction, clustering and learning. But the discerned context has not been applied to securing the devices themselves. This paper demonstrates use of contextual information for *fingerprinting* mobile devices.

Anomaly detection research has been pursued to complement signature-based intrusion detection systems (e.g. anti-virus). Anomaly detectors applied to intrusion detection are known to detect novel attacks but generate false alarms. Network-based systems monitor network protocol headers and payload and can be specific to a protocol or application [3, 23, 26]. Host-based systems model system calls [18, 22], and have used representations such as neural networks [14] and finite state automata [33]. Combinations of system features are proposed in [34]. Most of the security-related anomaly detection research is focused on wired networks and hosts, or specific to certain protocols or applications, though a framework has been proposed to detect routing anomalies in ad-hoc wireless networks [40]. Anomaly detection has also been used to detect outliers in spatial data [2], where neighborhood relationships are modeled and outliers are identified. Most recently, suspicious moving objects have been detected as anomalies [24]. Though it involves route modeling, it deals with additional attributes like speed and direction information generally not available on mobile phones. This paper proposes contextual information-based device profiling for anomaly detection.

Smoothing techniques have been applied to problems arising due to data sparseness in speech recognition [29, 19], text compression [28, 36] and information retrieval [31, 39]. They are essentially variance-reduction techniques, where a small probability mass is subtracted from total probability of *seen* events and assigned to novel (*unseen*) events. A review of various smoothing techniques is presented in [6, 27]. Smoothing techniques can be straightforward like additive smoothing, where a constant frequency count is added to each observed event. It can also involve discounting observed frequencies [30], distinct observation count [36], or counting observations of certain frequencies [15, 19, 7]. This paper presents smoothing methods to detect spatial temporal anomalies for mobile devices.

3. Approach

This section introduces the framework for detecting abnormalities attributed to unauthorized users. We describe how probabilities are learned and anomalies are scored. We introduce the problem of data sparseness that motivates the need for smoothing. We also describe a few smoothing techniques in this section.

The scenario is represented in Fig. 1. For an authorized user *A*, cell tower communicates with a mobile device and captures its location. The information is sent to a centralized server, where a model is learned for the device location over different time intervals, as explained in the next section. The device may communicate with multiple cell towers, but all contextual information is routed to the centralized server for model learning. Though fine grained location information is known to the cell tower (for services such as E911), we use cell ID for our experiments as that information is easy to extract. Although multiple cell IDs can correspond to a single physical location, we argue that all possible IDs can potentially be observed and learned over a period of time. But cell ID data lacks the physical topology and proximity of the actual locations. 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 $t4 - t6$ in the figure). Now consider the possibility of device theft by unauthorized user *U*. Any subsequent usage (Fig. 1 time instance *t6*) by *U* would most likely be inconsistent with the model in terms of location and time, hence raising an anomaly flag. The device can then be locked by the network operator using a PIN that can only be unlocked by the

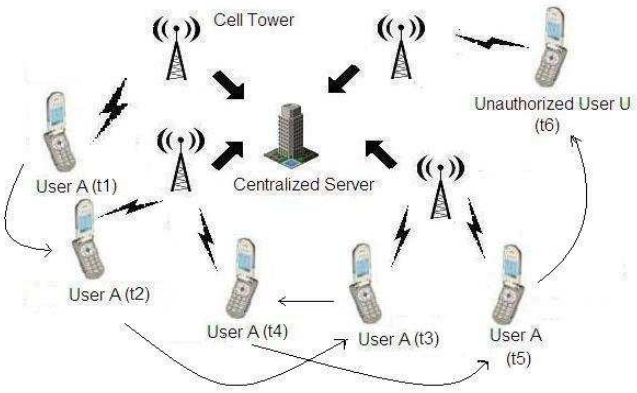

Figure 1. Mobile Anomaly Detection: For authorized user A, mobile data is used for context learning over a period of time ($t1 - t3$). Subsequent locations (at time instance $t4 - t6$) are validated against the model to ensure conformance. An unauthorized user U is likely to be inconsistent with the spatio-temporal model for user A.

operator or the authentic user.

3.1. Naive Approach

We propose tracking the frequency of the mobile phone 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 phone *m* at location *l* during time interval *t* is estimated by:

$$P_m^t(l) = \frac{freq_m^t(l)}{\sum_l freq_m^t(l)} \tag{1}$$

P_m^t is called the **spatial 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 (δ) of 10 minutes results in 144 intervals per day (η). This creates a profile for a single day. Thus, for any day of week *d*, a profile consists of η spatial probability distributions (Eq. 1) denoted formally as

$$Profile_m^d = (P_m^{d,1}, P_m^{d,2}, ..., P_m^{d,\eta}) \tag{2}$$

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

During the monitoring (test) phase, we use the learned profile (*spatial probability distributions*) to estimate the likelihood of each data record in the test set. To include some state information of where the mobile phone was previously, we consider a 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 $P_m^{d,W}(l_{c-w+1}, l_{c-w+2}, ..., l_{c-1}, l_c)$ as $P_m^{d,W}(l_W)$, and approximate it using probability chain rule as:

$$\begin{aligned} P_m^{d,W}(l_W) = \; & P_m^{d,t_c}(l_c|l_{c-1}, ..., l_{c-w+1}) \times \\ & P_m^{d,t_{c-1}}(l_{c-1}|l_{c-2}, ..., l_{c-w+1}) \times \\ & \cdots \times P_m^{d,t_{c-w+2}}(l_{c-w+2}|l_{c-w+1}) \times \\ & P_m^{d,t_{c-w+1}}(l_{c-w+1}) \end{aligned} \tag{3}$$

In the equation above, the probability of a sequence of states is 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. For simplicity and because the independence assumption of the Naive Bayes classifier generally seems to work well [10], we assume independence between subsequent locations, resulting in $P_m^{d,t_c}(l_c|l_{c-1}, ..., l_{c-w+1}) = P_m^{d,t_c}(l_c)$. The likelihood of mobile phone m over the time window W is thus approximated as the product of the marginal probabilities:

$$P_m^{d,W}(l_W) = \prod_{i=c-w+1}^{c} P_m^{d,t_i}(l_i) \tag{4}$$

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

$$log(P_m^{d,W}(l_W)) = \sum_{i=c-w+1}^{c} log(P_m^{d,t_i}(l_i)) \tag{5}$$

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 spatial probability distribution over a window W:

$$\begin{aligned} & AnomalyScore_m^{d,W}(l_W) = -log(P_m^{d,W}(l_W)) \\ & = -\sum_{i=c-w+1}^{c} log(P_m^{d,t_i}(l_i)) \end{aligned} \tag{6}$$

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 usually not valid, since to get to a specific location one typically traverses a fixed set of locations. The assumption is relaxed with Markov Chains, described next.

3.2. Markov Chain

In 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 spatial probability distribution of Eq. 1 is modified as

$$P_m^t(l_j|l_k) = \frac{P_m^t(l_j,l_k)}{P_m^t(l_k)} = \frac{freq_m^t(l_j,l_k)}{freq_m^t(l_k)} \tag{7}$$

For the Markov Chain, $P_m^{d,t_c}(l_c|l_{c-1},...,l_{c-w+1}) = P_m^{d,t_c}(l_c|l_{c-1})$. The probability estimate for the sequence of traversed states of Eq. 3 is now revised as

$$\begin{aligned} P_m^{d,W}(l_W) = \quad & P_m^{d,t_c}(l_c|l_{c-1}) \times P_m^{d,t_{c-1}}(l_{c-1}|l_{c-2}) \times \\ & \cdots \times P_m^{d,t_{c-w+2}}(l_{c-w+2}|l_{c-w+1}) \times \\ & P_m^{d,t_{c-w+1}}(l_{c-w+1}) \end{aligned} \tag{8}$$

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

$$\begin{aligned} & AnomalyScore_m^{d,W}(l_W) = -log(P_m^{d,W}(l_W)) \\ & = -logP_m^{d,t_{c-w+1}}(l_{c-w+1}) - \sum_{i=c-w+2}^{c} log(P_m^{d,t_i}(l_i|l_{i-1})) \end{aligned} \tag{9}$$

The Naive approach in Section 3.1 can be considered as zero-order Markov Chain, whereas this section describes first-order Markov Chain. 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.

Both the approaches discussed so far can flag valid but low frequency events as anomalous, resulting in higher false alarms. Next, we present STAD to alleviate false alarms due to low probabilities.

3.3. STAD

In the naive and Markov Chain based approaches of Sections 3.1 and 3.2, probabilities of observed and novel events contribute to the overall probability and anomaly score. To reduce false alarms and increase detection rates, we reason that observed events should have no contribution to the final anomaly score. Hence, we introduce STAD (Spatial Temporal Anomaly Detection) that only considers novel events in calculating the anomaly score. STAD ignores frequency for observed locations to reduce false alarms. 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, an employee heading to work uses an alternate route when there is traffic congestion on the regular route. 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 STAD.

For STADn (or n^{th} order STAD), only smoothed probability for novel event $P_m^{d,W}(l_i|l_{i-1}...l_{i-n})$ is estimated. Smoothing methods and probability estimation formulae are discussed in Section 3.4. We investigate zero and first order STAD in this paper, called STAD0 and STAD1 respectively. STAD0 maintains the subsequent location independence assumption of Naive approach, whereas the assumption is relaxed in STAD1 and the current state depends on the previous state. In addition to reducing false alarms, STAD ignores observed frequency counts to reduce the size of the stored model. Since observed events have no contribution to the anomaly score, frequency counts (or probabilities) of observed events need not be stored. Further, it improves the computational efficiency by avoiding additional calculations of anomaly score involving observed location probabilities. STAD assigns negative logarithm of the novel probability estimate as the anomaly score for the current test instance. For STAD0, it is

$$Score0_i = \begin{cases} -log(P_m^{d,t_i}(l_i)), & \text{if } freq_m^{d,t_i}(l_i) = 0 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

The anomaly score for l_c is aggregated over a time window W (l_W) of w instances to ignore spurious anomalies. For STAD0, it is computed as:

$$AnomalyScore_m^{d,W}(l_W) = \sum_{i=c-w+1}^{c} Score0_i \tag{11}$$

For STAD1, which assumes that the current state is dependent on the previous, the anomaly score for the current test instance is calculated as:

$$Score1_i = \begin{cases} -log(P_m^{d,t_i}(l_i|l_{i-1})), & \text{if } freq_m^{d,t_i}(l_i,l_{i-1}) = 0 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

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

$$AnomalyScore_m^{d,W}(l_W) = Score0_{c-w+1} + \sum_{i=c-w+2}^{c} Score1_i \tag{13}$$

An alarm is generated on exceeding a threshold.

STAD uses the probability estimate of novel events to score anomalies. Variance reduction techniques are required to compute the non-zero unobserved event probability estimate, described next.

3.4. Smoothing Probability Estimates

In the event of an anomaly (i.e. a novel location), spatial probability distribution underestimates the probability of the new value by assigning it a value 0, resulting in an undefined anomaly score (Eqs. 6, 9, 11, 13). This problem of data sparseness is similar to the one arising in maximum likelihood estimator of a language model in natural language processing. Various smoothing techniques are used to adjust the probability values such that no value has probability zero or one. We experimented with four smoothing techniques that are simple to understand but inherently different in nature and rationale, and have been successfully applied to a variety of problems. But our approach can be easily extended to other smoothing methods as well.

3.4.1 Additional Notations

Let s be the number of times a device was present at a specific location l in a given time interval. Thus, $s = freq(l)$. We denote n as the total frequency count $= \sum_l freq(l)$; 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 $\sum_k k f_k = n$.

3.4.2 Good-Turing Estimate

The Good-Turing method [15] is a popular and widely accepted variance reduction and probability estimation technique. It also forms a basis for other methods, such as Katz [19] and Church-Gale [7] smoothing. A simplified Good-Turing approach is presented in [13]. The underlying idea is that of curve fitting through *seen* value pairs (k, f_k) using some function F, and subsequently using $F(k)$ as a smoothed value. The total probability mass associated with novel values is $\frac{f_1}{n}$.

For our study, we used the following approximation to the Good-Turing estimate:

$$P(l) = \begin{cases} (1 - \frac{f_1}{n})\frac{s}{n}, & \text{if } l \text{ is seen} \\ \\ \frac{f_1}{n}, & \text{otherwise} \end{cases} \quad (14)$$

Since we do not know apriori the total number of *unseen* values, we assign the entire mass to the current novel location. For the case when $f_1 = n$, we used the *fall-back* scheme presented in [36].

3.4.3 Witten-Bell Smoothing

Witten and Bell studied different schemes to deal with the *zero* frequency problem in adaptive statistical coding, where token probabilities are estimated in given context and used for compression [36]. They found the following estimate (due to [28]) to give the best results:

$$P(l) = \begin{cases} \frac{s}{n+r}, & \text{if } l \text{ is seen} \\ \\ \frac{r}{n+r}, & \text{otherwise} \end{cases} \quad (15)$$

This is referred to as *Method C* in the original paper [36]. The rationale behind the above equation 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 $S_2 = \langle 1, 2, 1, 0, 5, 7, -8, 12, 0, 9 \rangle$, 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.

3.4.4 Absolute Discounting

Absolute discounting [30] involves deleted estimation, i.e. reduction of frequency count for all *seen* events by a constant, and assigning the aggregated mass uniformly to *novel* events. Let the discount constant be denoted as $D \in [0, 1]$. We approximated the smoothed probability values as:

$$P(l) = \begin{cases} \frac{s-D}{n}, & \text{if } l \text{ is seen} \\ \\ \frac{Dr}{n}, & \text{otherwise} \end{cases} \quad (16)$$

The value for the discount parameter D is suggested as $\frac{f_1}{f_1+2f_2}$ in [30]. It is interesting to note that Eq. 16 is a further smoothed version of *Method B* [$P(seen) = \frac{s-1}{n}$; $P(novel) = \frac{r}{n}$] in [36], where D = 1.

3.4.5 Dirichlet Priors

Given a distribution $P = \{P_1, P_2, \ldots, P_N\}$ such that $\sum_i P_i = 1$, a Dirichlet distribution for P is $Dir(\alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i P_i^{\alpha_i - 1}$, where $\Gamma(x)$ denotes the gamma function and α_i is the hyper-parameter, i.e. a parameter of prior. The model is a multinomial distribution and the appropriate value for α_i (for a unigram model) is suggested as $\mu P_C(l)$, where $P_C(l)$ is the probability of location l in a collection C of contexts [25]. The Dirichlet smoothed model is thus represented as

$$P(l) = \frac{s + \mu P_C(l)}{n + \mu}. \quad (17)$$

The Laplace method is a special case with $\mu = 1$.

For each of the equations above (Eqs. 14- 17), note that the sum of all smoothed probability values (including novel events) is unity.

Table 1. Confusion matrix in the context of anomaly detection for mobile devices

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

4 Empirical Evaluation

4.1 Experimental Data and Procedures

To evaluate and compare the smoothing methods, we used over 500,000 hours of context data formerly studied to better understand individual behavior and group patterns [11]. The data was logged using the Context Phone framework [32] and comprises of varied contexts including location, time, and device proximity. Approximately 90 different users, ranging from freshmen to graduate students to faculty members at a university, were the subjects in the experiments.

For our experiments, we extracted the location and time data - location corresponds to the cell id (for cellular networks), and timestamp was broken into *day of the week* and *time of the day* features. Though multiple cell ids can correspond to a single location, we assert that all of them can be observed and learned by our model over time. Our anomaly detector can also be used with location information of varied granularity, such as GPS-based coordinates that boast of an accuracy within 50 feet. A sampling rate of 10 seconds/instance was used in our experiments.

To quantify the efficacy of the the smoothing algorithms, we used location data for all devices. Disjoint training and test sets were created for each user. Since we do not have explicit labels for bad behavior, for each mobile device, we pretend that the behavior of unauthorized users is similar to the other mobile devices. That is, given a trained model for a mobile device, test data from the remaining devices were used to approximate behavior of unauthorized users. The confusion matrix is presented in Table 1. A test data sample from device B is validated against all models. Any alarm against model A is a true positive and against model B is a false alarm. No alarm against model B is a true negative, but not flagging an anomaly against model A is considered an undetected malicious attack on the device.

We compared the four smoothing methods using the Naive approach. The best smoothing strategy was then used to evaluate and compare Naive, Markov Chain, STAD0, and STAD1. The discount parameter D in absolute discounting (Eq. 16) is assigned the value $\frac{f_1}{f_1+2f_2}$, as suggested in [30]. For Dirichlet smoothing, parameter μ in Eq. 17 is set to 0.5 in the spirit of the widely accepted Jeffreys-Perk's law (or Expected Likelihood Estimation).

4.2. Evaluation Criteria

Computer security techniques are typically evaluated using a Receiver Operator Characteristic (**ROC**) curve. The ROC curve plots the rate of correct anomalies detected (i.e. different device) alongwith 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 number of false alarms. Biometric systems use another evaluation criterion called the equal error rate (**EER**), which is the point at which the true negative rate equals the true positive rate (i.e. the point where the ROC curve intersects the line $y = 1 - x$). A lower EER value implies better performance, since it reflects how well a system curtails false alarms as well missed detections. We compute the equal error rates to compare the various smoothing methods applied for anomaly detection. Apart from model accuracy, it is also imperative for a security mechanism to be computationally efficient. We compare the time requirements for model creation and validation.

Table 2. Area under ROC curve (AUC) for varied δ values. Larger AUC value is better.

Smoothing Method	AUC			
	δ=10 min.	*δ=20 min.*	*δ=30 min.*	*δ=60 min.*
Good-Turing	0.766	**0.786**	0.798	0.814
Witten-Bell	0.680	0.725	0.743	0.767
Absolute-discounting	**0.770**	0.781	0.788	0.801
Dirichlet	0.769	0.784	**0.801**	**0.823**

Table 3. Equal Error Rate (EER) for varied δ values. Smaller EER value is better.

Smoothing Method	EER			
	δ=10 min.	*δ=20 min.*	*δ=30 min.*	*δ=60 min.*
Good-Turing	0.289	0.268	0.258	0.241
Witten-Bell	0.373	0.333	0.313	0.289
Absolute-discounting	0.258	0.246	0.239	0.225
Dirichlet	**0.247**	**0.236**	**0.217**	**0.203**

4.3. Parameter Settings and Comparison of Smoothing Methods

Time interval δ is a parameter (Section 3.1) common to all smoothing methods for model creation, as explained in Section 3. Coarse-grained values for δ reduce sparseness, but tend to include multiple contexts. On the other hand, fine-grained values for δ are focused on specific context but exacerbate the issue of data sparseness. We experimented with 4 different values for δ: 10, 20, 30 and 60 minutes. Tables 2 and 3 list the AUC and EER values respectively for the smoothing methods using different δ values. Results show that performance improved with increase in the interval size, irrespective of the technique. This suggests that larger time intervals alleviate the problem data sparseness, since more data is available to model. Inclusion of more samples also improves the spatial probability distribution for observed events. A small δ includes less data, resulting in inaccurate models and higher number of anomalies, thereby increasing the number of false alarms as well. Also, larger interval sizes accomodate spurious anomalies better than small intervals. For example, an employee is usually at work around 8:30 in the morning but could be late by 30 minutes due to a traffic jam. A small δ, say 10 or 20 minutes, will cause a false alarm but a larger interval size might still accept it as a normal event. Thus, large δ values perform better.

We also studied the effect of the time window parameter W (Eq. 4) on the accuracy of smoothing methods. Large W value would take longer to capture an anomaly but help restrain false alarm generation. Small W value, on the other hand, is expected to identify the spatio-temporal anomaly at an early stage thereby minimizing loss, but have higher number of false positives. AUC and EER results for W values ranging from 10-60 minutes are compiled in Table 4. Results show that though AUC increases and EER decreases with increase in W, the increase in performance was not significant. Thus, small window size can be used without loss in accuracy and emphasizes on the suitability of the smoothing methods for online usage.

Amongst the four smoothing methods, Dirichlet smoothing generally performs the best to detect anomalies, as seen from Tables 2 and 3. But why did one smoothing method perform better than the others? The anomaly score is high if there is a novel event. The more the anomalous events are aggregated, the higher the anomaly score. A statistical language model is considered better if it assigns a high probability to words than another technique. But anomaly detection is inherently different from language modeling as aberrations with low probability need to be flagged. The lower the probability, the more severe the alarm would be. The best anomaly detector scores novel event the highest, thus assigning it the lowest probability, as compared to an observed event. Probability comparison of smoothing methods is presented in Table 5. For the test set, the table lists the average *seen* and *unseen* event probability values for all smoothing techniques. The performance can largely be attributed to the way the probability mass is allocated for novel events (Eqs. 14- 17). Dirichlet smoothing uses hierarchical modeling - in addition to the current context, it also utilizes the knowledge from a collection of contexts. This enables two pronged learning (from individual context and context collection) and results in more accurate modeling. Since Dirichlet smoothing assigns *unseen* events with very low probability, it performs the best and suggests more suitability for anomaly detection. Thus, we used it to compare the four anomaly detectors for detecting spatio-temporal anomalies for mobile devices, discussed next.

Table 4. AUC and EER range for different window size (δ = 60 min.)

Smoothing Method	AUC range	EER range
	***W*=10–60 min.**	***W*=10–60 min.**
Good-Turing	0.801 – 0.814	0.252 – 0.241
Witten-Bell	0.754 – 0.767	0.299 – 0.289
Absolute discounting	0.792 – 0.801	0.233 – 0.225
Dirichlet	0.814 – 0.823	0.211 – 0.203

Table 5. Probability comparison for smoothing methods for δ=60 min.

Smoothing Method	Average Probability	
	Observed	Novel
Good-Turing	0.141	0.025
Witten-Bell	0.138	0.047
Absolute discounting	0.144	0.013
Dirichlet	0.142	0.009

4.4. Comparison of anomaly detection techniques

We evaluated and compared the four anomaly detectors – Naive, Markov Chain, STAD0 and STAD1 as zero and first order STAD respectively. Dirichlet smoothing was used with parameters δ = 60 and W = 10 minutes. Accuracy was measured in terms of *AUC* and *EER*, and time requirements for training and testing were noted.

Accuracy Anomaly detection systems are prone to high false alarm rates (FAR). Dealing with higher number of false alarms can be overwhelming and frustrating for the user. It is thus imperative for an anomaly detector to minimize them. Thus, in addition to computing the AUC of the

Table 6. Area under ROC curve (AUC) upto various false alarm rates (FAR).

Technique	AUC			
	$\times 10^{-5}$	$\times 10^{-3}$	$\times 10^{-1}$	
	FAR=0.001	*FAR=0.01*	*FAR=0.1*	*FAR=1*
Random	0.050	0.050	0.050	0.500
Naive	0.150	**0.101**	0.080	0.814
Markov Chain	**0.200**	0.046	0.230	0.857
STAD0	0.150	**0.101**	0.080	0.808
STAD1	**0.200**	0.052	**0.280**	**0.862**

Table 7. Equal Error Rate (EER) comparison.

Technique	EER
Random Detector	0.500
Naive	0.211
Markov Chain	0.197
STAD0	0.209
STAD1	**0.184**

entire ROC curve (i.e. 100% FAR), we also compute the area upto lower FAR - 0.1%, 1% and 10%.

Table 6 lists the AUC values for the anomaly detectors upto various FAR, with the second row listing the constant factor multiplied to AUC at the respective FAR. The EER results are presented in Table 7. The random detector has the same false alarm rate and true positive rate for any threshold (x=y line for ROC). The AUC was highest for STAD1 for all FAR except 1%, where Naive and STAD0 performed better. STAD1 also had the lowest EER, which indicates fastest convergence. Both these tables suggest that STAD1 anomaly detection technique generally performs the best to detect the spatio-temporal anomalies. Comparing the same order techniques against each other, Markov Chain followed STAD1 closely but never outperforming it. Naive approach and STAD0 had same AUC upto 0.1 FAR, though Naive was slightly better on the complete ROC curve. But at the same time, STAD0 had lower EER than Naive technique.

Time Requirements for Training and Testing For an anomaly detection system to be effective, it should be able to detect misuse in real-time. Training can be performed offline, but testing needs to be performed online to minimize loss. During training, STAD does not increment the frequency counts, and only novel locations contribute to the anomaly score during testing. So we expected STAD to be faster as compared to the other techniques. For completeness, we computed time requirements for model creation (training) as well as model validation (testing) for the four anomaly detection techniques. Experiments were performed on a SUN Ultra 60 workstation with 450 MHz clock speed and 512 MB RAM. The results are compiled in Table 8. As expected, STAD0 and STAD1 time requirements were slightly better than Naive and Markov Chain approaches respectively. Compared to STAD1, STAD0 was 38% faster in training and 33% faster in testing, as only marginal probabilities are involved. But all four approaches incur reasonable computational overhead for an online system.

Table 8. Average training and testing rates (microseconds/instance).

Technique	Training Rate	Testing Rate
Naive	0.282	0.279
Markov Chain	0.469	0.415
STAD0	0.280	0.279
STAD1	0.450	0.414

Table 9. Comparison of no-update vs. model update on false alarm. For random detector, AUC = EER = 0.5.

Technique	AUC		EER	
	No-update	Update	No-update	Update
Naive	0.814	0.924	0.211	0.156
Markov Chain	0.857	0.912	0.209	0.165
STAD0	0.808	**0.946**	0.197	**0.092**
STAD1	0.862	0.918	0.184	0.115

Model update using feedback The limitation of anomaly detection systems is the generation of false alarms. In our problem setting, it represents flagging a new location visited by an authorized user. For example, a researcher attending a conference in a new city will constantly violate the learned model, flooding the device with false alarms and overwhelming the user. To suppress false alarms, the new location information is fed back to the centralized server to update the trained model for each erroneous alarm. We repeated the experiments and computed the AUC and EER for ROC curves obtained for all techniques with model update upon false alarm. Table 9 lists and compares the results of the no-update techniques (Section 4.4) with the ones being updated. Results show a significant improvement in the number of detections for model update in all four anomaly detectors, indicating its effectiveness in tackling concept drift and curtailing false alarms. Amongst the four techniques, STAD0 performed the best with the highest AUC and lowest EER for model update. STAD0 also had the maximum improvement, with 17% AUC increase and EER reduced by 53%.

4.5. Privacy Issues

As per our architecture, the models are created and stored on a centralized server. It is thus imperative to discuss the privacy implications. The continuous monitoring of user information may be deemed as unethical and breach of privacy which every individual is entitled to. Currently, not all mobile devices have high storage capabilities to store data instances for training. But technology is progressing fast with time and high storage devices do not seem a distant future. Our experiments were aimed at demonstrating that device usage patterns can be modeled and detect unauthorized usage and also understand the smoothing variables responsible for detecting anomalies. This knowledge can be applied as appropriate with the advances in technology. With sufficient device storage capacity, the data instances and the spatial probability distribution models can be stored locally on the device. Alternatively, incremental models can be learned instead of batch training. The device can run the anomaly detector without any information being sent off it. On detecting any spatial temporal anomaly, it can activate a *self-lock* mode that can only be unlocked by the authorized user or the carrier.

5. Conclusions

We presented an automated technique to detect spatial temporal anomalies for mobile devices to alleviate the problem of lost and stolen devices. Our technique creates stochastic user models and aggregates model violations to flag alarms. We used probability smoothing methods that have been successfully applied in language modeling for speech recognition, data compression and information retrieval. We performed experiments on a real data set comprising over 500,000 hours of mobile data. We evaluated four smoothing methods - Good-Turing, Witten-Bell, absolute discounting and Dirichlet smoothing. Parameter study suggested using larger time interval size (δ = 60 min.) for better performance due to more accurate modeling (Section 4.3). Low W values (= 10 min.) are effective in detecting spatio-temporal anomalies without significant loss in accuracy than higher values. Amongst the four smoothing methods Dirichlet priors performed the best, with highest area under ROC curve (AUC) and lowest equal error rate (EER) for varied δ values (Section 4.3). The reason for the best performance of Dirichlet smoothing lies in the fact that it assigns a lower probability to novel events as compared to other methods.

We presented STAD in Section 3.3. We evaluated and compared the zero and first order STAD (called STAD0 and STAD1 respectively) with Naive approach and Markov Chain (Section 4.4). Results indicate that our technique was effective in capturing contextual abnormalities due to an unauthorized user, with STAD1 attaining the highest AUC of 86.2%. STAD1 also demonstrated fastest convergence with the lowest EER of 0.184. STAD reduces the storage requirements of the trained models by ignoring observation frequencies. This also enabled it to reduce the number of false alarms. STAD has low time requirements for both training (0.28-0.45 μsec./instance) and testing (0.28-0.41 μsec./instance), suggesting suitability for online usage. We also demonstrated the effectiveness of feedback for model update to take into account concept drift, thereby reducing the generation of false alarms. AUC for STAD0 increased to 94.6% and EER was 0.092.

The anomaly detector captures deviations from the spatial probability distribution model. Since all aberrations do not imply misuse, false alarms are generated. Similarly, it is possible for misuse to go unnoticed. This is even more relevant to our data set as the subjects are university students and faculty, many of whom may go to same building and demonstrate contextual similarity. Finer-grained location data can be used (e.g. GPS) to tackle this issue. Alternately, our anomaly detector can be used in a conglomerate of security mechanisms for a mobile device, including signature-based systems like anti-virus and multi-modal biometric systems such as fingerprint and voice recognition.

References

[1] http://www.ericsson.com/solutions/operators/news/2005/q2/20050411_mobile_device_management.shtml.

[2] N. R. Adam, V. P. Janeja, and V. Atluri. Neighborhood based detection of anomalies in high dimensional spatio-temporal sensor datasets. In *ACM SAC*, 2004.

[3] D. Anderson, T. Lunt, H. Javitz, A. Tamaru, and A. Valdes. Detecting unusual program behavior using the statistical component of the next generation intrusion detection expert system (nides). Technical Report SRI-CSL-95-06, Computer Science Laboratory SRI, 1995.

[4] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *IEEE Infocom*, 2000.

[5] G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical Report TR2000-381, Dartmouth College, 2000.

[6] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University, Cambridge, Massachusetts, 1998.

[7] K. W. Church and W. A. Gale. A comparison of the enhanced good-turing and deleted estimation methods for estimating probabilities of english bigrams. *Computer, Speech and Language*, 5:19–54, 1991.

[8] D. Dedo. Windows mobile-based devices and security: Protecting sensitive business information. Technical report, Microsoft Corporation, April 2004.

[9] A. K. Dey and G. D. Abowd. Towards a better understanding of context and context-awareness. Technical Report

GIT-GVU-99-22, Georgia Institute of Technology, Atlanta, Georgia, 1999.

[10] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[11] N. Eagle and A. Pentland. Reality mining: Sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, May 2006.

[12] J. A. Flanagan, J. Mäntyjarvi, and J. Himberg. Unsupervised clustering of symbol strings and context recognition. In *ICDM*, 2002.

[13] W. A. Gale and G. Sampson. Good-turing frequency estimation without tears. *Journal of Quantitative Linguistics*, 2(3):217–237, 1995.

[14] A. Ghosh and A. Schwartzbard. A study in using neural networks for anomaly and misuse detection. In *USENIX Security Symposium*, 1999.

[15] I. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3,4):237–264, 1953.

[16] A. Goode. Mobile data security - access, content, identity and threat management, 2006-2011. Technical report, Juniper Research, September 2006.

[17] J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmäki, and H. T. Toivonen. Time series segmentation for context recognition in mobile devices. In *ICDM*, 2001.

[18] S. A. Hofmeyr, S. Forrest, and A. Somayaji. Intrusion detection using sequence of system calls. *Journal of Computer Security*, 6:151–180, 1998.

[19] S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-35:400–401, 1987.

[20] P. Korpipää, M. Koskinen, J. Peltola, S.-M. Mäkelä, and T. Seppänen. Bayesian approach to sensor-based context awareness. *Personal and Ubiquitous Computing*, 7(2):113–124, July 2003.

[21] K. V. Laerhoven and O. Cakmakci. What shall we teach our pants? In *International Symp. Wearable Computers*, 2000.

[22] T. Lane and C. E. Brodley. Temporal sequence learning and data reduction for anomaly detection. *ACM Trans. Information and System Security*, 2(3):295–331, 1999.

[23] A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava, and V. Kumar. A comparative study of anomaly detection schemes in network intrusion detection. In *SDM*, 2003.

[24] X. Li, J. Han, S. Kim, and H. Gonzalez. Roam: Rule- and motif-based anomaly detection in massive moving object data sets. In *SDM*, 2007.

[25] D. J. MacKay and L. C. Peto. Speech recognition using hidden markov models. *The Lincoln Laboratory Journal*, 3:41–62, 1990.

[26] M. V. Mahoney and P. K. Chan. Learning rules for anomaly detection of hostile network traffic. In *ICDM*, 2003.

[27] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[28] A. Moffat. A note on the ppm data compression algorithm. Technical Report 88/7, University of Melbourne, Victoria, Australia, 1988.

[29] A. Nadas. Estimation of probabilities in the language model of the ibm speech recognition system. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-32(4):859–861, August 1984.

[30] H. Ney, U. Essen, and R. Kneser. On the estimation of 'small' probabilities by leaving-one-out. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(12):1202–1212, December 1995.

[31] F. Peng, D. Schuurmans, and S. Wang. Augmenting naïve bayes classifiers with statistical language models. *Information Retrieval*, 7(3):317–345, 2003.

[32] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen. Contextphone - a prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing*, 4(2):51–59, 2005.

[33] R. Sekar, M. Bendre, D. Dhurjati, and P. Bollineni. A fast automaton-based method for detecting anomalous program behaviors. In *IEEE Symp. Security and Privacy*, 2001.

[34] J. Shavlik and M. Shavlik. Selection, combination, and evaluation of effective software sensors for detecting abnormal computer usage. In *KDD*, 2004.

[35] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Trans. Information Systems*, 10(1):91–102, 1992.

[36] 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):1085–1094, 1991.

[37] J. Yin, X. Chai, and Q. Yang. High level goal recognition in wireless lan. In *AAAI*, 2004.

[38] M. A. Youssef, A. Agrawala, and A. U. Shankar. Wlan location determination via clustering and probability distributions. In *IEEE PerCom*, 2003.

[39] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Information Systems*, 22(2):179–214, 2004.

[40] Y. Zhang, W. Lee, and Y. Huang. Intrusion detection techniques for mobile wireless networks. *ACM/Kluwer Wireless Networks Journal*, 9(5), 2003.