DRFAT DO NOT DISTRIBUTE
SenSec: Mobile Application Security through Passive Sensing

Jiang Zhu, Sean Wang, Pang Wu, Joy Zhang
Carnegie Mellon University, Moffet Field, CA, USA
{*jiang.zhu, sean.wang, pang.wu, joy.zhang*}*@sv.cmu.edu*

Emmanuel Owusu, Adrian Perrig
Carnegie Mellon University, Pittsburgh, PA, USA
{*eowusu, adrian*}*@andrew.cmu.edu*

Abstract

The commoditization of sensor technologies coupled with advances in modeling user behavior offer us new opportunities for simplifying and strengthening authentication. We envision a new mobile system framework, *SenSec*, which uses passive sensory data to ensure application security: *SenSec* is constantly collecting sensory data from accelerometer, gyroscope, GPS, WiFi, microphone or even camera. Through analyzing the sensory data, it constructs the context under which the mobile device is used including locations, movements and usage patterns, etc. From the context, the system can calculate the certainty that the system is at risk. We then compare the certainty with the sensitivity levels for different mobile applications. For those applications of which sensitivity passes the certainty threshold, authentication mechanism would be employed before the application is invoked to ensure security policy for that application. In this paper, we investigate and evaluate a number of inexpensive and easily acquired passive factors, rigorously examining how well these factors differentiate between people including for example motion, location, running applications, etc. We developed methods for fusing these passive factors and model people's behavior in a manner that is effective, robust, and reliable. We built a functional SenSec prototype based on our MobiSens framework and evaluated it the effectiveness to detect behavior anomalies using the data collected by the system from 50 users for one months. The results show that our approach can archive 70∼80% accuracy with only a few days of sensory data.

1 Introduction

Mobile applications and devices are becoming ubiquitous and will increasingly interact with different sensors, services and other mobile users. It is crucial for mobile users to privately and securely interact with their environment and data and for mobile services to trust the identity of the user. While mobile devices such as smartphones make our lives convenient in ways that were unimaginable before, applications such as email, web browsing, social network, shopping and online banking know too much about our private lives. Mobility introduces additional security and privacy challenges in being able to provide services in a way that neither compromises the environment of users nor their data. Protecting a user's privacy and ensuring the accountability of mobile applications in a seamless and non-intrusive way poses great challenges to next generation mobile computing platforms.

Recently, a new survey [1] has revealed that 36 percent of consumers in the United States have either lost their mobile phone or had it stolen. Another survey [2] has also revealed that 329 organizations polled had collectively lost more than 86,000 devices with average cost of *lost data* at $49,246 per device, worth $2.1 billion or $6.4 million per organization. Given the high loss rate and high cost associated with these losses, accountable schemes are needed to protect the data on the mobile devices.

Reliable authentication is an essential requirement for a mobile device and its applications. Today, passwords are the most common form of authentication. This results in two potential problems. First, passwords are also a major source of security vulnerabilities, as they are often easy to guess, re-used, often forgotten, often shared with others, and are susceptible to social engineering attacks. Secondly, to secure the data and applications on a mobile device, the mobile system would prompt user for authentication quite often and this results in series usabil-

[1] Strategy One survey conducted among a U.S. sample of 3,017 adults age 18 years or older on September 21-28, 2010, with an oversample in the top 20 cities (based on population)

[2] "The Billion Dollar Lost-Laptop Study" conducted by Intel Corporation and the Ponemon Institute, analyzed the scope and circumstances of missing laptop PCs

ity issues. We also observe that different applications on a mobile device may have different sensitivities towards the aforementioned threats and data loss. For example, the Angry Bird game on an android is less sensitive than Contact List or Phone Album should the device is operated by unauthorized user. One-thing-for-all approach in authentication schemes may be either too loose for some applications, which expose them to risks, or too tight for others, which cause usability problems.

The commoditization of sensor technologies coupled with advances in modeling user behavior offer us new opportunities for simplifying and strengthening authentication. We envision a new mobile system framework, *SenSec*, which uses passive sensory data to ensure application security: *SenSec* is constantly collecting sensory data from accelerometer, gyroscope, GPS, WiFi, microphone or even camera. Through analyzing the sensory data, it constructs the context under which the mobile device is used. This includes locations, movements and usage patterns, etc. From the context, the system can calculate the certainty that the system is at risk. We then compare the certainty with the sensitivity levels for different mobile applications. For those applications of which sensitivity passes the certainty threshold, authentication mechanism would be employed before the application is invoked to ensure security policy for that application.

In this paper, we investigate and evaluate a number of inexpensive and easily acquired passive factors, rigorously examining how well these factors differentiate between people including for example motion, location, running applications, etc. We developed methods for fusing these passive factors and model people's behavior in a manner that is effective, robust, and reliable. We built a functional SenSec prototype based on our MobiSens framework [7] and evaluated it the effectiveness to detect behavior anomalies using the data collected by the system from 50 users for one months. The results show that our approach can archive 70 80% accuracy with only a few days of sensory data.

2 Related Work

Mobile applications and devices are becoming ubiquitous and will increasingly interact with different sensors, services and other mobile users. It is crucial for mobile users to privately and securely interact with their environment and data and for mobile services to trust the identity of the user. General authentication mechanisms explored in past work mainly include biometric authentication, passive authentication and context-aware authentication.

Biometric authentication verifies user identity by leveraging the uniqueness of behavioral characteristics and/or physical trait. Over the years, a number of biometric techniques have been introduced. Traditional schemes may include fingerprint scanners, iris recognition, voice recognition, face recognition and so forth. Since last decade, novel approaches have sprung out.

Orr and Abowd [6] designed a system which can identify users based on their footstep force profiles. Extensive footstep data is used for modeling and testing. In the work by Peacock et al. [8], authors concentrated on keystroke patterns and the underlying typing characteristics to perform user identification. They provided a crystal overview by surveying most published literature. Drawbacks found in previous work were addressed to bring the scheme more practical implication. Jakobsson et al. [5] put forward the notion of implicit authentication. Authentication strategy is carried out based on what applications and features on a mobile phone are being used.

Other biometrics including blinking pattern, writing style, etc. are explored in literature. These metrics tend to focus on behaviors of individuals. The implicit authentication proposed by Jakobsson et al. [5] partially captures our idea of application security enforcement through passive sensing. We both consider to leverage user behavior patterns to perform authentication scheme. In their work, the behavioral features considered are time-lapse since the user last checked email and the GPS location. During the training phase, distribution of feature values is estimated for each time of day value t. In the testing time, the feature distribution function at time t assigns the likelihood score of the observed feature value. The two feature scores are combined through a weighted linear function to calculate an overall "authentication score", which is then compared with a pre-defined threshold to authenticate the user. This work is motivated to solve similar problems as our project. The main difference is that our scheme provides the framework of a generic solution to the contextual security problem. And we take an elastic approach which take into accounts both the security score of system and the sensitivity of applications that are supposed to be protected from unauthenticated access. Also, we do not intend to use passive/implicit authentication to replace active authentication methods such as emails. Rather, SenSec security scheme constantly evaluates the certainty that mobile devices or application are at risks and prompts the user for active authentication when the risk level is deemed high.

As sensing and computing capability become standard on smartphones, researchers have begun to collect sensory data on device and to use the on-device computing resource to infer user behavior and the environment they interact with. In the work of Zheng et al. [9], the GPS reading has been used to detect whether a person is walking, running, in a car, or on a bus. Fogarty et al. [2] have

employed microphones and keyboard activity to classify whether a person is interruptible or not. Keng-hao et al. [4] have proposed a method to identify different household members. They detect the unique pattern people wield the television remote control using the accelerometers embedded in. This technique can help enable personalized TV watching. In the paper by Davrondzhon et al. [1], the proposed authentication scheme is enforced by recognizing user's gait. They use accelerometer to record acceleration of lower leg for the gait recognition. Their approach achieves sound accuracy.

Our work differs from the aforementioned efforts in three folds:

1. We fully utilized the sensing capability of the mobile device and collected multi-dimensional sensory data for analysis and modeling.

2. We converted heterogenous sensory data into behaviorial text and adopt n-gram model to efficiently construct sufficient statistics of user behavior, capturing both the steady-state (uni-gram) and transient behavior (bi-gram and n-gram) simply through counting.

3. We also built a Risk-Analysis-Tree model and carefully designed a set of experiments to evaluate the risk of the context attributed by various factors and discovered which factors are more influential than others.

3 System Architecture

In this section, we provide a 20000-feet picture of SenSec framework. Functional components will be explained to help readers understand the way SenSec works. Figure 1 illustrates the framework of the proposed *SenSec* mobile application security approach.

SenSec constantly collects sensory data from the mobile device. From *hard sensors* such as accelerometers, gyroscope, compass, GPS receivers and *soft sensors* such as calendar and active Apps, we infer the contextual factors including network, personal, behavioral and application factors of the mobile device. SenSec collects sensory data through MobiSens App.

Raw data from sensors will first be preprocessed through quantization, clustering or partitioning to be converted into symbolic representations for the convenience of modeling. We will describe in details the data preparation phase in later section.

Various sensor information is then fused together to infer meta-factor information such as the activity of the user. SenSec trains module for each user which will be pushed back to device through MobiSens. Contextual information is collected in real-time by on-device sensors and then fed to the module to detect any anomaly. The risk of mobile device and applications is evaluated periodically.

The system combines relevant contextual factors to assess the risk level "certainty score" of the mobile device which indicates the certainty that the system is at risk. The score will be fed to a comparator to determine which level of security scheme to be performed.

On the other hand, different Apps on the device have various sensitivities. Normally, game Apps like Angry Birds are less sensitive than contact list or photo album. Each App is automatically or manually assigned a sensitivity score. When the user clicks on one App icon, the sensitivity score is also fed to the comparator. The comparator takes into accounts both the sensitivity score of that app and the current risk score of system to determine the security scheme to be performed. Specifically in our project, the SenSec module may choose to prompt the user to enter password. For example, running less-sensitive applications like posting on Facebook at locations such as work or home that a person spends a lot of time at, choose easy authentication methods. Similarly, for places that she rarely or never goes to, or when the walking gait is very different from her usual pattern, make authentication highly reliable such as requesting the user to pass face recognition if she is about to open the Email client.

4 Contextual Security Certainty Model

4.1 Contextual Factors

Modern mobile devices come with various mobile sensors such as accelerometers, gyroscope, magnetometer, microphone, camera, GPS receiver, WiFi and Bluetooth receivers, etc. Combined, these sensors are able to sense the context of the mobile device such as the outdoor and indoor location, wireless environment, user's motion and gesture.

In our study, we are focusing on the following factors

- Surrounding environment: GPS coordinates and WiFi Access Point (WAP) SSID and signal strength. Most of the time, a mobile device is likely to be in just a few environments such as office and home. Enumerating those highly-trusted network environment is quite straightforward and computationally cheap.

- User activities: motion sensors (accelerometers, gyroscope, compass). This factor describes physical aspects of individuals. Many of the characteristics we will explore here are highly related to biometrics such as how a user holds her phone and how she types on the keyboard. and application in use.

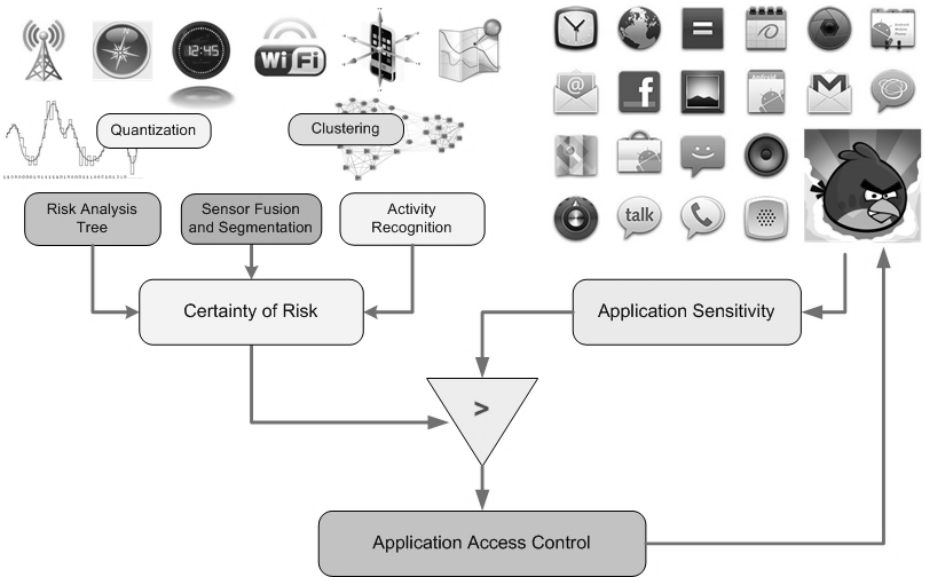

Figure 1: SenSec Framework. Contextual factors collected or inferred from mobile sensors are used to assess the certainty of the mobile device. The system adjusts the security settings based on the running App's sensitivity level accordingly. If the certainty level is low and the app is highly sensitive, strong authentication is required to assure the accountability of the application.

- Communications: application specific network traffic.

4.1.1 Modeling User's Behaviors by Text Representation

Mobile devices are equipped with various types of sensor which could constantly monitor certain aspects of a user's behavior. However, these sensors are heterogeneous in data sampling rate, data format and meaning of sensor readings. Other approaches such as [3] build classifiers on each sensor readings and combine the classification results to model user's behavior.

Based on the principle of "language as action", natural language and human behavior share common characteristics. They are both "meditational means" or tools by which we achieve our ends, their meanings both depend on the structure of the observed sequence of the composing elements, and they both have grammars that help to explain the underlying "syntactic structure" of an observed sequence. By representing the sensor readings as symbols through quantization or clustering and applying statistical natural language processing algorithms on these symbols sequences, we can then picturing mobile sensors constantly "writing" a lifelogger in text about our lives. Applying statistical natural language processing algorithms on this *lifelog* will allow us to build behavior models for a user.

In our study, we modeled motion trace using the n-gram model. Similar to natural language, we assume that the sequence of user's behavior can also be predicted by her motions in the past. We consider a pseudo behavior label as a "word" in the language and train an n-gram language model on the motion trace data as text by converting each motion feature vector into a symbolic label.

The model estimates the next behavior label l_i given the previous $n-1$ pseudo locations from the WiFi trace as $P(l_i|l_{i-n+1}, l_{i-n+2}, \ldots, l_{i-1})$ or $P(l_i|l_{i-n+1}^{i-1})$ in short.

By feeding the sequence of the behavioral labels a sequence of behavioral labels $L = l_1, l_2, \ldots, l_N$ in a stepping window of size N to the learned n-gram model, we can also estimate the probability that this sequence is generated by the given n-gram model as

$$P(L) = P(l_1, l_2, \ldots, l_N) = \prod_{i=1}^{N} P(l_i|l_{i-n+1}^{i-1}) \qquad (1)$$

or average log probability as

$$\frac{1}{N}\sum_{i=1}^{N} \log P(l_i|l_{i-n+1}^{i-1}) \qquad (2)$$

The model probabilities $P(l_i|l_{i-n+1}^{i-1})$ can be estimated through the Maximum Likelihood Estimation (MLE) from the training data by counting the occurrences of behavioral text labels: $P_{\text{MLE}}(l_i|l_{i-n+1}^{i-1}) = \frac{C(l_{i-n+1},\ldots,l_{i-1},l_i)}{C(l_{i-n+1},\ldots,l_{i-1})}$

4.1.2 Partition Feature Space through Risk Analysis Tree

Effective modeling the context requires the system to consider different factors together. In pervasive computing, this line of research is referred to as heterogeneous sensor fusion. Intuitively, fusing information from different types of sensors should improve the discriminative power of the passive authentication classifier. For example, we can obtain the user's outdoor location trace and her traveling speed with just a GPS sensor. However, if her mobile phone is stolen by an attacker that is familiar with the user's commuting pattern and walks along the same route as the user while to hack into the system, then the geo-trace based on GPS information will not be able to catch this attack. In this case, fusing the location information with other sensor information such as gait and phone tilting can help to identify the anomaly. The most straightforward way of sensor fusion is to concatenate several sensor readings into one value and train the classifier on the concatenated input. For example, after fusing the location, acceleration and time information, the input may look like "location=37.378535 N 122.086585 W; acceleration-x=0.017g; acceleration-y=0.002g; acceleration-z=-0.014g; time=2:05pm".

However, this approach suffers *the curse of dimensionality* problem. When the number of fused sensors increases, the total number of possible combined values increases exponentially. The fused sensor representation requires enormous amount of training data to train a reliable statistical model in order to be general enough to classify unseen data. Even if we preprocess the sensor data by quantizing the sensor readings after which real number values are converted to discrete symbolic labels, the vocabulary's size can still be in the range of hundreds or thousands and the combination of all sensor values can easily grow to millions or billions. Thus fusing by concatenating is only practical to fuse a small number of sensors. Another problem of this approach is over partition. Over partition happens when the classifier partitions the value space into too many classes. For example, if we fused "application usage" information with "location" and "time" information through concatenation, the classifier might partition the "checking email" activity to multiple activities such as "checking email in office at 2:13pm", "checking email in the hallway at 10:30am" and "checking email at home at 11:23pm". This can be overkill as checking email can happen at any time. Knowing the habit of where the user usually checks emails is sufficient for contextual security.

Similar to decision trees, a technique widely used for classification tasks, we can automatically construct a risk analysis tree for each user to determine whether the computer system is at risk or not by querying multiple sensors. Figure 2 shows an example Risk Assessment Tree. Here, RAT first queries the sensors to get the user's outdoor location. If the user is not at home or at work, RAT then queries the GPS to obtain her traveling speed. In case she is most likely driving given the traveling speed, RAT checks which application is running on the mobile device and apply a risk assessment function that fuses the sensor of geo-trace, app-category and time information. Intuitively, RAT partition the full feature spaces into small regions where certain combination of sensors are most effective to asses the risks of the mobile device. To train the RAT model, we take two approaches with slight differences:

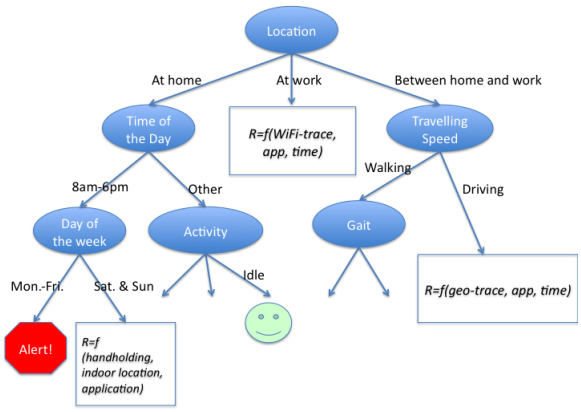

Figure 2: Sensor fusion through Risk Analysis Tree. This tree is conceptually similar to a decision tree, where a system traverses the tree based on responses at each node. RAT partition the full feature spaces into smaller regions and train a sensor fusion function for each region to estimate the current security certainty.

1. Owner Detection - Use the labeled data with simulated attacks. Each leaf can have two outcomes: [0,1], where 1 indicates the data is from the owner, while 0 indicates it is from a unauthorized user.

2. User Identification - Use the data generated by all the users [1...N] and label them with the given user ID. Each leaf can take N outcomes: [0,1,...N]

We then build a decision-tree classifier to partition the feature space into subspaces where a subset of features are responsible for the risk assessment. For each subspace, we will then train a regressor to optimize weights for each sensor feature to best quantify the risk level of the mobile device. Moreover, we can systematically select subsets of the features and build multiple RATs and use a weighted average or voting method to combine the decision from these RATs to quantify the risk level.

5 Data Collection and Feature Constructions

In this section, we explain how raw data is collected through MobiSens framework. From the selected dataset, features of each user can be extracted for training and evaluation. We will crystallize the method used in feature construction.

5.1 Data Collection

Imbedded in the MobiSens framework is a subsystem employed for data collection. In our data collection phase, 50 MobiSens users with Android phones constantly uploaded their sensory data and other context information to the MobiSens server. Average collection period is as long as 30 days.

The data we collected from each user fall into 7 types, which is necessary for non-trivial user behavior modeling. We will specify the data types in the following subsection. Under MobiSens framework, each type of data is uploaded by certain rate which is called sampling rate. The corresponding interval between consecutive transmissions is sampling interval. In our settings, we design the finest sampling interval as 200 *ms*. This rate is for the motion sensory data which may vary frequently. Sharp vibrations within short duration can be captured only if suitable sampling rate is applied. The accuracy of behavior modeling will benefit from the fine precision in user data flow.

After collection, the total dataset size is around 4 *GB*. We preprocess the user data to prepare for further feature construction. Two heavy users are removed. Their excessively large datasets may cause bias when fed to the training model as the system gains much more information from them compared with other light users. Except for the heavy users, we also remove those users who do not upload application and traffic data due to the older version of MobiSens installed in their Android phones. At last, we obtain 25 user datasets with comparable size. The data duration for different users varies from 4 hours to 2.5 days. It can be demonstrated in our performance evaluation that with only days of training data we can achieve sound anomaly detection accuracy.

5.2 Feature Extraction

To prepare training data, we extract features from selected raw data and generate feature matrix for each user. The construction method specified for each data type will be elaborated.

In our dataset, 3 out of all 7 types of data come from on-device motion sensors including accelerometer, gyroscope and compass. Each of them is consisted of 3 dimensions. While constructing features, for every single dimension, we calculate 11 statistics as shown in Table 1 to generate a feature vector with 99 elements in total, which is then combined with the timestamp to obtain the complete feature vector for motion sensory data. When calculating statistics, we apply a step window to the dataset. Window size is adjustable and here is assigned as 5 seconds. The window steps along the time dimension according to the timestamp. User data within this window will be processed as a whole to calculate the feature vector. Timestamp for this vector is the initial timestamp of the current window. Upon completing processing the whole dataset, a feature matrix can be generated for the user.

Feature	Description	D/M
RMS	The Root-Mean-Square value	D
RMSE	The Root-Mean-Square error	D
Min	The minimum value	D
Max	The maximum value	D
AvgDeltas	The average sample-by-sample change	D
NumMax	The number of local peaks	D
NumMin	The number of local crests	D
TTP	The average time from a sample to a peak	D
TTC	The average time from a sample to a crest	D
RCR	The RMS cross rate	D
SMA	The signal magnitude area	D

Table 1: Features To Be Extracted From Motion Sensory Data

The raw data from GPS receiver has 4 dimensions. By applying density based clustering, we convert the data to a single location label which is appended to current timestamp to form the feature vector of GPS readings.

WiFi data is in the form of pair (SSID, RSSI) where SSID is the WiFi Service Set Identifier and RSSI is the Received Signal Strength Indication. From the collected WiFi information, we select the top 3 pairs ranked by signal strength. The size of final vector with timestamp is 7 then.

Under the new MobiSens framework, application (app) information and app traffic are uploaded to server. Basically, app status at sampling point is encapsulated into app information dataset. In our analysis, we care more about the 59 most popular apps among all the users in our experiment. Hence, a bitmap with 59 bits is generated for those popular apps. All the other apps will be classified as one category and get represented by 1 bit. The value for each bit indicates the status of an apps, where '0' describe the fact that this app is off at certain

Feature Set	Leaf Count	Tree Size	Accuracy
All	56	111	99%
Non-Motion	3	5	96%
Motion only	267	533	98%

Table 2: Results from Simple Experiments

time, '1' means the app is on and '2' tells us that the app is not only running but providing important service. The size of the whole feature vector is 61.

For those 59 apps and all the other apps, we also collect their traffic information. Both transmitted and received traffic in bytes are included as the traffic feature vector for users.

With feature matrix for each type of data prepared, we merge them into a single feature matrix for each user. As motion sensory feature vector has the highest granularity, the merging is based on the timestamp in sensory feature matrix. For each vector with certain timestamp, we search in the 4 other feature matrices to find the vector with its timestamp matching the time duration indicated by the timestamp in sensory feature vector which is $timestamp + step\ window\ size$. The resulting vectors will be appended to sensory vector to generate the complete feature vector with size 287. Ample feature input will enhance the accuracy in training and modeling.

6 Experiments and Results

6.1 Simple Experiments and Observations

We started with a small scale experiment with a small data set. We conducted both User Identification and Owner Detection tasks using partial data set from 4 randomly selected users. We split the data equally into training and testing sets. We ran experiments with 3 different combinations of features:

1. all features, including motion, location, application and communications
2. features from non-motion sensors, including location, application and communications
3. features from motion sensors only

If we use all the features, RAT yields accuracy at 99% with a modest complex model. When we remove all motion features from the data set, the same tasks result in a significant reduction in model size with a slightly lower accuracy at 96%. If we only use motion features, the model still can distinguish different users with about 98% accuracy. However, the trained model that produces this result is a relatively large tree. The results are shown in Table 2

From the results, we observe that our approach can detect user behavior anomaly with reasonable accuracy. Particularly, we confirm that we can either identify users or detect non-owner using only motion features. This shows that we can rely on the way how mobile device users hold and use their devices to passively authenticate users. We also observe that different combination of features will lead to similar accuracy results but with very different models in size. If only motion feature can be used, i.e. only relying on user's motion behavior, the resulting model size is too large to be used on mobile device. In order to put our approach into practical use, we may have to rely on other features to collaboratively assess the risks.

6.2 Results and Performance Evaluation

The simple experiments show promising results. However, we are facing an issue in term of fairness of the experiments. Intuitively, some aspects of user behavior are more user-specific than others. For example, motion patterns are quite unique to each person if sensors can capture it accurately. On the other hand, WiFi SSID and signal strength factor may not be powerful to distinguish two users as this factor comes from the common environment, such as university campus or office building, which is shared by many users. Cross entropy provides a way to measure the correlations of two distributions, and it is a good fit for our problem. We can quantify the discriminative power of each contextual factor by calculating the averaged cross-entropy between two user's the behavior models. That is, for a pair of users A and B, the idea is to estimate the likelihood of using A's model to explain B's data. If the contextual factor is very user-specific, then A's model should not be able to explain B's data well and should result in a low likelihood. This gives us a low cross-entropy between A and B for that particular factor.

We calculated the cross entropy of the data sets for all 25 users and chose the 12 users with least cross entropy. This is to ensure these users behaviors are relatively similar and it will provide fair evaluation with the simulated anomaly. To evaluate the accuracy and performance of our approach, we use these 12 users' trace, with total 210,000 data instances to train a user identification model. We use 66% data for training and the rest for testing.

The results reported in Table 3 shows similar trend as what we obtained in the simple experiments described in 6.1. It shows that after filtering out users with higher cross-entropy in their behavior features, the accuracies of all these experiments did drop, but still within a reasonable range. This indicates that our approach can still be used to identify users using various passive sensory data even if the users shares some aspects of their be-

[width=5in]

Features	Leaf Count	Tree Size	Accuracy	Exp. Time (sec)
All	111	221	84.8%	378.17
Non-Motion	18	35	83.5%	198.06
Motion only	777	7649	79.3%	402.18

Table 3: User Identification Experiments and Performance Evaluations

haviors. It also shows that the performance of our model is very sensitive to the selection of the features. If we were to only use motion-only sensors, we will result in a very large model. Although the training and testing time for such a model is not significant longer than that of other feature configurations, the size of storage to hold the model itself would become an issue on the mobile device. Moreover, traversing a large RAT would cause performance degradation and more power consumption on the mobile device, which prevent this approach from large scale deployments. On the other hand, non-motion features provide a very compact model with least resource constraints and still produce reasonable accuracy. This shows that using these sets of features would enable us to efficiently estimate risks on mobile device in real time.

6.3 Working Prototype

In order to mitigate the aforementioned problem and make it feasible to do risk assessments on the mobile device, we trained a RAT model for each user using the non-motion features and a n-gram model using the motion features. These models are pushed to the SenSec application on the mobile device. SenSec application constantly feed all the sensors data to these two models and combines the outcome to produce a weighted average certainty score. SenSec application also provides a user an interface to adjust risk threshold. When the threshold is met, the application would trigger an alert both in form of intent broadcast and on-screen notification. A demo video [3] shows a typical scenario of SenSec application when a risk is present.

7 Conclusion

We investigated and evaluate a number of inexpensive and easily acquired passive factors, rigorously examining how well these factors differentiate between people (including for example GPS location, known characteristics of that location, IP address, and App traffic). The degree of discrepancy help assess the certainty of risk. We also investigated behavioral models of user (such as frequency of being at a given location, how recently a person authenticated on another device, time since last activity, device holding patterns and movement and habit in using Apps) as well as characteristics of places (such as the number of unique people seen in a given location and the WiFi readings), empirically validating the usefulness and feasibility of various models.

We developed new methods for fusing these passive factors and models of people and places together in a manner that is effective, robust, and reliable. Mobile applications, upon being invoked, can obtain the certainty of risk and compare it with its application level sensitivity. If the application is very sensitive and certainty of risk is high, certain authentication mechanism can be put in place to secure the access. We built a functional SenSec prototype based on our MobiSens framework [7] and evaluate it both for security and for usability.

After collecting the dataset of 12 users, we conduct experiment to evaluate our approach. With only days of training data, our result achieves around 80% accuracy in discovering anomaly. As future work, we plan to extend the data set for feature construction. The App traffic dataset will be expanded to include TCP and UDP readings. To better assess the environment that users interact with, we may also consider ambient lighting and sound. In addition, user pattern can be assessed by exploring battery status. In modeling and training phase, we will try to gain more insights into the data, features and factorized relationships among various sensors. Some other classification methods including LR, SVM, Random Forest, etc. may be exploited. We can compare the results for better accuracy. Also, to enhance the security of SenSec components, the integration with Android security framework and other Apps should be explored. Seamless connection is essential to prevent traditional attacks. Besides security, privacy issues also deserve to receive attention. We will try to develop privacy policy in data collection and processing phase. One additional future direction is energy related. To endow SenSec with more practical implication, energy consumption in on-device data collection and processing should be taken into account. We will envision an more energy-efficient framework to improve the usability of SenSec for mobile devices.

[3] YouTube Video: http://youtu.be/TA4_SLzihHo

References

[1] DAVRONDZHON GAFUROV, K. H., AND SONDROL, T. Biometric gait authentication using accelerometer sensor. *Journal of Computers 1* (2006).

[2] FOGARTY, J., HUDSON, S. E., ATKESON, C. G., AVRAHAMI, D., FORLIZZI, J., KIESLER, S., LEE, J. C., AND YANG, J. Predicting human interruptibility with sensors. *ACM Trans. Comput.-Hum. Interact. 12* (March 2005), 119–146.

[3] GELLERSEN, H. W., SCHMIDT, A., AND BEIGL, M. Multi-sensor context-awareness in mobile devices and smart artifacts. *Mob. Netw. Appl. 7* (October 2002), 341–351.

[4] HAO CHANG, K., HIGHTOWER, J., AND KVETON, B. Inferring identity using accelerometers in television remote controls. In *In Proceedings of the International Conference on Pervasive Computing* (2009).

[5] JAKOBSSON, M., P. S. E., GOLLE, AND CHOW, R. Implicit authentication for mobile devices, 2009.

[6] ORR, R., AND ABOWD, G. The smart floor: A mechanism for natural user identification and tracking. *ACM Press* (2000).

[7] PANG WU, J. Z., AND ZHNAG, Y. Mobisens: a versatile mobile sensing platform. In *submitted to The International Conference on Mobile Systems, Applications, and Services* (2012).

[8] PEACOCK, A., KE, X., AND WILKERSON, M. Typing patterns: a key to user identification. *Security Privacy, IEEE 2*, 5 (sept.-oct. 2004), 40 –47.

[9] ZHENG, Y., LI, Q., CHEN, Y., XIE, X., AND MA, W.-Y. Understanding mobility based on gps data. In *Proceedings of the 10th international conference on Ubiquitous computing* (New York, NY, USA, 2008), UbiComp '08, ACM, pp. 312–321.