SenSec: Mobile Security through Passive Sensing

Jiang Zhu, Pang Wu, Xiao Wang, Joy Zhang
Department of Electrical and Computer Engineering
Carnegie Mellon University
Moffett Field, CA, USA
{jiang.zhu, pang.wu, sean.wang, joy.zhang}@sv.cmu.edu

***Abstract*—We introduce a new mobile system framework, *SenSec*, which uses passive sensory data to ensure the security of applications and data on mobile devices. *SenSec* constantly collects sensory data from accelerometers, gyroscopes and magnetometers and constructs the gesture model of how a user uses the device. SenSec calculates the *sureness* that the mobile device is being used by its owner. Based on the sureness score, mobile devices can dynamically request the user to provide active authentication (such as a strong password), or disable certain features of the mobile devices to protect user's privacy and information security. In this paper, we model such gesture patterns through a continuous n-gram language model using a set of features constructed from these sensors. We built mobile application prototype based on this model and use it to perform both user classification and user authentication experiments. User studies show that SenSec can achieve 75% accuracy in identifying the users and 71.3% accuracy in detecting the non-owners with only 13.1% false alarms.**

***Index Terms*—mobile sensing, behavior recognition, user identification, passive authentication**

I. Introduction

Reliable and convenient 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 major sources of security vulnerabilities, as they are often easy to be guessed, 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 serious usability issues. Protecting a user's privacy and ensuring the accountability of mobile applications in a seamless and non-intrusive ways poses great challenges to next generation mobile computing platforms.

The commoditization of sensor technologies coupled with advances in modeling user behavior creates new opportunities for simplifying and strengthening mobile device security. We envision a new mobile system framework, *SenSec*, which uses passive sensory data to ensure mobile application security. *SenSec* collects data from accelerometer, gyroscope and other sensors and builds its owner's gesture model. Once the model is built, *SenSec* also continuously evaluates the *sureness* whether the mobile device is under the control of the owner. This information can be used subsequently by on-device security sub-systems to control the access to resources, services and data as shown in Figure 1. A common use scenario of *SenSec* would like this: You are at a party and

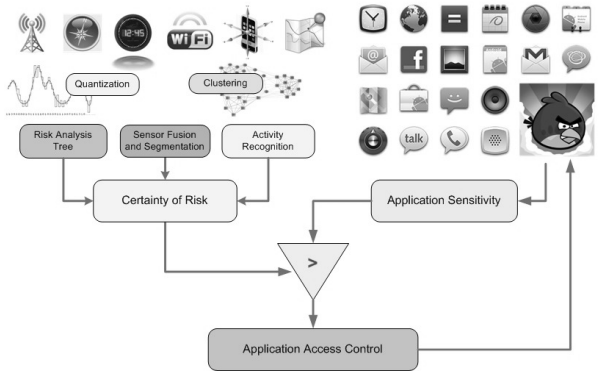

Fig. 1. SenSec Overview: A Motivating Example

mistakenly leave your phone unattended. The phone is picked up by one of your friends. Out of curiosity, she starts to browse your applications and data. You certainly would not mind if she only plays a mobile game, but would feel extremely uncomfortable if she starts to view your contact list and uses your mobile banking apps. Because the gesture patterns are so different between you two, the *SenSec* system would detect this incident and automatically trigger further authentication, if any *sensitive* applications are invoked.

In this paper, we propose a probabilistic approach to model user's gesture patterns using a generative continuos n-gram language model. We built a functional *SenSec* prototype, which keeps track of sensory reading from accelerometer, gyroscope and magnetometer on device, builds the context of user's gesture pattern and uses that to verify user's identity. We conducted two sets of user experiments using this prototype:

1) *Offline User Identification*. We asked users to use the same device to perform the same set of actions for several times (e.g., picking up the device from a table and starting to use a certain application) and collected the motion sensory data during these actions. We then performed user classification task to infer user's gender, occupation, age and its user ID.
2) *Online Anomaly Detection*. With the behavior model trained from one user's sensor data, we evaluated the effectiveness of *SenSec* by testing how well it can detect the incident that the phone is used by another user.

Results from the offline experiments show that SenSec system achieves 75% accuracy in classifying users, while the results from the online experiments show that the system is able to successfully detect 71.3% cases when the phone is

978-1-4673-5288-8/13/$31.00 ©2013 IEEE

operated by non-owner(anomalies) and only to trigger false alarms for 13.1% cases when it's still used by the owner.

II. Related Work

Biometric authentication, which applies context-awareness [1], [19] to security applications, verifies user identity by leveraging the uniqueness of behavioral characteristics and/or physical trait. Traditional schemes include fingerprint scanners, iris recognition, voice recognition, face recognition and so forth. Since last decade, novel approaches have sprung out. Orr and Abowd [11] designed a system which can identify users based on their footstep force profiles. Work by Peacock et al. [12] concentrated on keystroke patterns and the underlying typing characteristics to perform user identification. Work by Davrondzhon et al. [6] proposed a scheme that uses accelerometers mounted on lower leg for gait recognition and uses it to enforce authentication. Other biometrics including blinking pattern, writing style, etc. were also explored in the past which focus on individuals' behaviors.

More recently, as sensing and computing capabilities become standard on smartphones, researchers have begun to collect more types of sensory data on devices to build user behavior models and use the model to infer certain contexts, including user authentication. In the work of Zheng et al. [20], the GPS readings are used to detect whether a person is walking, running, driving a car, or riding a bus. Schmidt et al. [13] monitors various system parameters on a mobile device, including system free memory, running process count, user inactivity, CPU usage and SMS count, for anomaly detection and IDS. Keng-hao et al. [8] predicts whether the user is interruptible based on the input from microphone and the user's keyboard activities by detecting the unique pattern of holding the television remote control using the accelerometers attached. Jakobsson et al. [9] put forward the notion of implicit authentication in that the authentication strategy is carried out based on what applications and features on a mobile phone are being used. Shi et al. [15] also devised an implicit authentication architecture in which a user model is constructed from a user's past behavior recorded by the mobile device. Then with this model and recently observed user activities, device can score the trustworthiness of current user and respond accordingly. SenGuard [16], a similar system to *SenSec*, leverages availability of multiple sensors on smartphones and passively use them as sources of user identification in background. It invokes active user authentication when there is a mounting evidence that the phone user has changed.

Our work extends beyond the aforementioned efforts in that 1) Instead of using devices mounted to various part of the human body, we explored the possibility of using the sensory data collected from generic smart phones to model user's gesture patterns while users are using the devices. 2) Different from SenGard, which uses JigSaw engine [10] to detect 5 common physical activities, stationary, walking, cycling, running, and in a vehicle (i.e., car, bus), *SenSec* uses a novel and simple n-gram model to capture individual users' motion gesture. 3) One of our experiments was based on the data collected from a group of users performing the same tasks with the device. This further verify that gesture patterns are very personal and can be used for user identification and classification. 4) We integrated our approach into a prototype system and tested it in participants' the day-to-day usage. Our system can identify non-owner anomaly in less than 5 seconds, faster than existing work [9], which makes our approach practical for real deployments.

Natural Language	Behavior Language	Example
Word	Atomic Movement	Device tilt and movement
Phrase	Movement	Picking up the phone
Sentence	Action	Making a phone call
Paragraph	Activity	Search for the nearest pizza place and place a phone order
Document	Event	Prepare for a lunch meeting

TABLE I
BEHAVIOR AS LANGUAGE AT DIFFERENT LEVELS.

III. Mobile Device Gesture Patterns

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 focus on the factors governed by motion and posture sensors, i.e. accelerometers, gyroscope, orientation sensors and magnetometers. These factors describe physical aspects of individuals. Many of the characteristics we explore here are highly related to biometrics, such as how a user retrieves the phone, how she holds it while using certain applications, how she types on the keyboard and press the buttons.

A. A Language Approach on Behavior Modeling

The similarity between human behavior and language had been articulated by Burke [2] and Wertsh [18]. They are both "mediational means" or tools by which we achieve our ends. They exhibit structure and satisfy "grammars". Table I illustrates that ambulatory behavior share a lot in common with natural languages at all levels. Atomic movements form the vocabulary of the behavior language. A sequence of atomic movements performed in meaningful order creates a *movement* such as an *action* of "picking up the device". *Actions* such as "making a phone call" are created by performing actions in a right order similar to create a "sentence". A sequence of actions builds up an *activity*. Higher level behavioral concept *event* is composed of a series of activities in a similar way as a *document*.

Shannon in [14] established that a language could be approximated by an n-th order Markov model (n-gram). Using an n-gram model trained on the English text, we can estimate whether "United" or "house" is more likely to follow the phrase "the president of the" by comparing the probability P("United" | "the president of the") and P("house" | "the president of the"). Such an n-gram model has also been proven to be very robust in modeling sequences of data other than

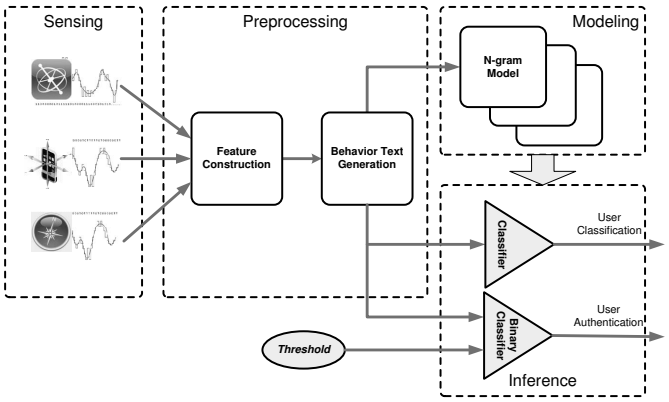

Fig. 2. SecSec: modeling motion and posture patterns using a language approach

language. Similar to [3]–[5], [21], in our work, we adopt a scheme by converting real-valued sensor readings to *symbols* through quantization, vectorization or clustering and using n-gram models to capture the patterns in a user's motion gesture.

In *SenSec* framework, as shown in Figure 2, we convert the raw sensory data into ***behavior text representation*** as sequences of behavior labels. Each behavior label is considered as a "word" in the language. We then train a ***continuos n-gram language model*** on those traces and use the trained model for user identification and anomaly detection.

1) Behavior Text Representation: Data produced by motion sensors usually are multi-dimensional and real-valued. This may prevent us from using them directly in our language approach due to computation complexity. To convert the sensory readings into behavior text, we first segment the raw sensor readings into a series of chucks of fixed size in time. For each chunk, we then construct a set of features and build feature vector for that time slot. After that, these feature vectors are clustered into V classes using K-means algorithm. The centroids of the resulting V classes form the vocabulary **V**. Thus, the sequence of feature vectors can be mapped to a series of class labels $\{l_0, l_1, l_2, \dots, l_n, \dots\}$, where $l_i \in \mathbf{V}$. This forms the behavior labels for the n-gram processing.

2) Continous n-gram Model: An n-gram model is a n-order Markov model for predicting the next item in a sequence. The two core advantages of n-gram models are 1) relative simplicity and 2) the ability to scale up. Moreover, n-gram model is a generative model. Compared with discriminative models such as SVM, it also enables us to perform user authentication task with only positive training samples collected from the owner.

In our context, the n-gram model estimates the probability of the next behavior label l_i given the previous $n-1$ behavior labels from the traces denoted as in Equation 4.

$$P(l_i|l_{i-n+1}^{i-1}) = P(l_i|l_{i-n+1}, l_{i-n+2}, \dots, l_{i-1}) \tag{1}$$

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:

$$P_{\text{MLE}}(l_i|l_{i-n+1}^{i-1}) = \frac{C(l_{i-n+1}, \dots, l_{i-1}, l_i)}{C(l_{i-n+1}, \dots, l_{i-1})} \tag{2}$$

B. User Classification

We can build n-gram models for M different users or user groups (or classes). Without losing generality, let's denote **V** as the vocabulary and **G** as the set of all n-grams from such a behavioral text data set, then the models of these M classes can be described as probability vectors as $\overrightarrow{P}_0, \overrightarrow{P}_1, \overrightarrow{P}_2, \dots, \overrightarrow{P}_{M-1}$ where

$$\overrightarrow{P}_m = \{P_m(l_i|l_{i-n+1}^{i-1})| \quad l_i \in \mathbf{V}, l_{i-n+1}^{i-1} \in \mathbf{G}\} \tag{3}$$

For a sequence of behavioral labels of size N, $L = \{l_1, l_2, \dots, l_N\}$, we estimate the probability that L is generated by a given n-gram model $\overrightarrow{P}_m$ as

$$P(L, m) = P(l_1, l_2, \dots, l_N, m) = \prod_{i=1}^{N} P_m(l_i|l_{i-n+1}^{i-1}) \tag{4}$$

or average log probability as

$$\frac{1}{N}\sum_{i=1}^{N} \log P_m(l_i|l_{i-n+1}^{i-1}) \tag{5}$$

where $P_m(\dots)$ are the model probabilities of user m as in Equation 3.

Given a behavior text sequence L, the use classification problem can be formulated as

$$\hat{u} = \arg\max_m P(L, m) \tag{6}$$

where $P(L, m)$ is the probability the behavior text sequence L is generated by mth user's n-gram model as denoted in Equation 4. Therefore,

$$\hat{u} = \arg\max_m \frac{1}{N}\sum_{i=1}^{N} \log P_m(l_i|l_{i-n+1}^{i-1}) \tag{7}$$

$$= \arg\max_m \sum_{i=1}^{N} \log P_m(l_i|l_{i-n+1}^{i-1}) \tag{8}$$

C. User Authentication

User authentication problem can be formulated as binary classification problem, classifying a user as the owner ($\hat{a} = 1$) or not ($\hat{a} = -1$). We can model this scheme probabilistically as shown in Equation 9: given an observation r, evaluate the probability of a given user is the owner ($\hat{a} = 1$) and check to see if it exceeds a certain threshold θ.

$$\hat{a} = sign[P(u = 1|r) > \theta] \tag{9}$$

Specifically, given a sequence of behavior text L, and a sensitivity threshold θ, we want to validate if these sequence is generated by user m as

$$\hat{a}(L|m, \theta) = sign[P(L, m) > \theta)] \tag{10}$$

while
$P(L, m) = P(l_1, l_2, \dots, l_N, m) = \prod_{i=1}^{N} P_m(l_i|l_{i-n+1}^{i-1})$ as in Equation 4:

$$\hat{a}(L|m, \theta) = sign(\sum_{i=1}^{N} \log P_m(l_i|l_{i-n+1}^{i-1}) > \log \theta)$$

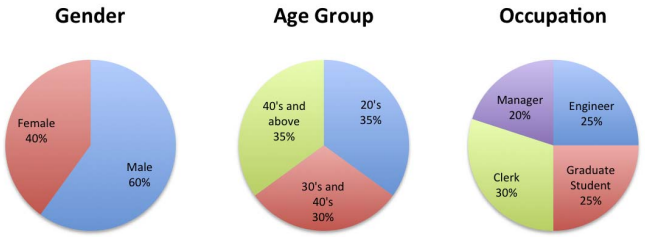

Fig. 3. Distributions of demographic groups for user classification

IV. Experiments

SenSec enables us to model user's gesture patterns and use it to do user classification and user authentication. For our next step, we want to investigate how effective and feasible this framework might be in practice.

In our first experiment, we investigated *SenSec*'s effectiveness in conducting user identification. The goal is to evaluate how well we can distinguish users by their gesture patterns when they are performing the same tasks.

In our second experiment, we put the *SenSec* framework in a real world scenario where the *SenSec* Application is performing user authentication on the device while users are using it.

A. Offline User Classification

20 volunteers [1] from different demographic groups (genders, occupations, and age) participated in the study. The distribution of their demographics are shown in Figure 3.

All participants were given the same Google Nexus S smartphone loaded with our sensing application and were asked to perform the same task for 5-10 times. Specifically, they were asked to perform the following tasks in a sequential order:

1) Pick up the device from a desk
2) Unlock the device using the right slide pattern [2]
3) Invoke Email app from the "Home Screen"
4) Lock the device by pressing the "Power" button
5) Put the device back on the desk

Traces for each experiment were matched with the participant's class labels and were sent to the server for offline modeling and classification. In this section, we first describe our data collection and preprocessing methods and then we present the results from offline user classifications.

1) Data Collection and Preprocessing Method: During the experiments, our sensing application monitors the accelerometers, orientations, gyroscopes and magnetometers. We set the sample rate of these sensors at 4Hz, i.e. the application produces 4 samples per second and each sample contains 12 real valued readings, including Accelerometer (X, Y, Z), Orientation (Azimuth, Pitch, Roll), Compass (X, Y, Z), and Gyroscope (X, Y, Z). Each experiment lasts about 20-30 seconds, which produces about 1250 time-stamped samples.

[1]User studies were conducted with the approval of IRB application HS11-094, "Human Behavioral modeling using Mobile Sensors" from the IRB Board at Carnegie Mellon University.

[2]No password or lock patterns are configured on the phone

Classification Target	No. of Classes	Accuracy
Gender	2	0.81
Age Group	3	0.79
Occupation	4	0.76
User ID	20	0.75

TABLE II
Experiment Results for User Classifications

These samples are then uploaded to our server and labeled with the hashed user ID to hide participant's real identity.

During offline preprocessing phase, we adopt a sliding window approach in constructing the features. We chose the window size of 2 second and stepping offset of 500 ms. Using the data in the window, each feature vector is composed by various statistical features [7], including Root Mean-Square, Root Mean-Square error, minimum value, maximum value, average sample-by-sample change, number of local peaks, number of local crests and combined signal magnitude, etc., to capture sufficient *statistics* of those micro movements.

2) Training and Testing: These motion feature vectors are clustered in V classes using K-means clustering algorithm. The centroids of the resulting V Classes are mapped to behavior text labels and become the dictionary of the model.

For a given classification tasks to predict participant's gender, occupation, age group and user ID, we randomly reserve 20% of the samples from each class as testing set $\{E\}$ and use the rest as training set $\{R\}$ to train a N order n-gram model for that class. For each sample L from $\{E\}$, we calculate $P(L, m)$ for the models of all the classes where $m = 1, 2, \ldots, M$ and select m with the maximum $P(L, m)$ as the class label prediction. We chose $V = 200$ and $N = 5$ due to performance and complexity trade-offs.

3) Results: We repeated above experiments for 5 times, which is equivalent to a 5-folder cross validation (CV). The results are shown in Table II. By using the data from a small group of participants, we can achieve a reasonable accuracy in classifying users of different classes. We also found that most of the misclassifications happened when the experiments were conducted back-to-back. There were chances that the latter participants may have observed previous participants' motion and posture and subconsciously tried to mimic what he or she have done [17]. We could re-evaluate our approach by conducting experiments in an isolated environment and with a large number of participants.

B. Online User Authentication

We integrated the aforementioned data preprocessing and modeling schemes into the sensing application and add a user authentication logic with a behavior-based authentication GUI. This resulted a new *SenSec* App as shown in Figure 4. We loaded this application to a smartphone and gave it to a group of participants to conduct the online user authentication experiments. [3]

1) Training and Testing: These experiments were carried out by the following settings:

1) Training Stage: Each of the participants uses the phone for 24 hours while the *SenSec* app is collecting sensory

[3]Youtube Video: http://goo.gl/Fq9Fi

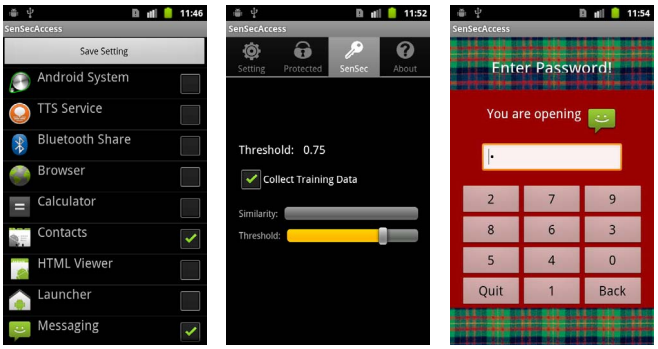

Fig. 4. The *SenSec* mobile application, from left to right: 1) Interface to select applications to be protected by *SenSec*. 2) *SenSec* control panel: start training and set the sensitivity threshold. 3) When an application is invoked while an anomaly is also detected (sureness score is low), a password screen is pop up for user authentication.

information in the background and build the behavioral n-gram model.

2) Positive Testing Stage: In this stage, each participants continue uses her phone for 24 hours. This time the *SenSec* app is switched to testing mode. It collects the sensors reading the same way as in training mode, but also construct behavior text sequence and feed it to the learned n-gram model. A *sureness* score is calculated as described in Equation 4. If it falls below a preset threshold while certain operation is performed, an authentication screen will be pop up asking user to enter a passcode. The *sureness* score and the authentication decision are recorded for logging and result reporting purpose.
3) Negative Testing Stage: The phones are given to other participants and let them use it for another 24 hours. As in the previous stage, the same operations are performed on the phone and all authentication events will be record for further analysis.

2) Results: We examined the logs generated by these experiments. At each authentication decision point, the sureness score is recorded. By varying the threshold, we can evaluate how well our *SenSec* models perform user authentication under different threshold values. For each threshold value, we can calculate False Positive Rate or FPR and True Positive Rate or TPR and plot Receiver Operating Characteristic (ROC) curves.

Figure 5 shows ROC curves for user authentication experiments. Each points on the ROC curve corresponds to a certain threshold. The upper left corner represents perfect authentication, while diagonal represents the model that is performing no better than a coin-flip.

True Positive Rate (TPR) represents the cases that system successfully detect non-owner access, while False Positive Rate (FPR) represents the false alarms when the owner is detected as a non-owner. To ensure the security of the system while still keeping the usability to a certain level, we need to maximize TPR and minimize FPR. The ROC curves provide a guideline on choosing the right thresholds to fulfill such a requirement. For example, the system may have a requirement to effectively detect 70% of the unauthorized access, but it

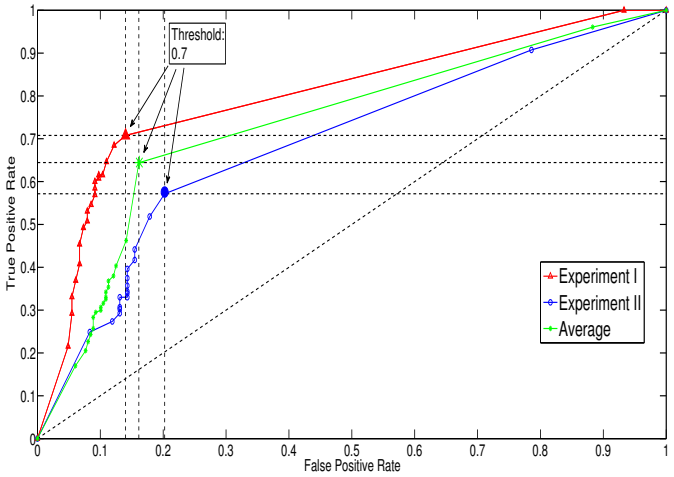

Fig. 5. Receiver Operating Characteristic (ROC) curves of the n-gram model for training size of 4 and 12 hours.

may need to keep the false alarm rate below 15%. As shown in Figure 5, we can choose a data point from the allowable region on the ROC curve: threshold 0.7 can achieve 71.3% TPR and only 13.1% FPR. We also examined the detection delay at this threshold by deliberately passing the phone between the two users more often and recording the time. By comparing these recorded time and the timestamps of triggered anomalies from the log, we found *SenSec* bears an average 4.96 seconds detection delay.

V. Discussions and Future Work

In the first experiment, we were able to classify users into different pre-defined categories by monitoring the motion and posture when they perform a given task using their smart phones. The level of accuracy shows that a small segment of user's gesture trace collected from the mobile device is able to capture the physical characteristic of the person. On one hand, these results show promising potentials of mobile sensing in various applications with high level of personalization: it enables the mobile application developers and mobile service providers to know more about the users. Thus, they can provide more customized services and assistances to the users. This indeed would improve the usability of the mobile devices. However, on the other hand, such capability may also reveal security vulnerability and raise privacy concerns, since it may allow the 3rd parties obtain too much personal information. Moreover, since the mobile sensing is usually a background or passive task, it may not raise enough user awareness so that they can take actions to protect their privacy and their data security. It is our belief that novel approaches to balance these two trade-offs would be of great research value. We have made the android app available to a small group of trial users and we are planning on a large scale study to further validate our hypothesis and improve our model in the presence of large data sets.

The second experiment served as an early trial of a personalized application mentioned above in a security context. While it is able to detect majority of the non-owner cases, the false alarms also decreased the usability dramatically. Even though

the overall FPR of our system is only 13.1%, participants in our experiments still complained that the system had triggered the false alarms too often. In one particular case, a participant was trying to answer a call while driving and the system triggered the passcode authentication multiple times to protect the dialer application. This is not only an annoyance, but also a safety hazard. To mitigate such a problem, we are planning to add other contexts along side with the motion and posture sensing to determine if further authentication is needed. The added context may be able to provide a more accurate results. Additionally, we observed that different applications on a mobile device may have different sensitivities (i.e. threshold) towards the threats and data loss. We are investigating an adaptive scheme to determine the thresholds for different applications based on applications profile and user's usage patterns.

VI. Conclusion

In this paper, we propose a probabilistic approach to model user's gesture patterns using continuous n-gram language model. We built a functional prototype *SenSec*, which keeps track of sensory reading from accelerometer, gyroscope and magnetometer on device, builds the context of user's gesture patterns and use them to perform user classification or to identify whether it is the owner. The results showed that *SenSec* system achieves 75% accuracy in user classification tasks and 71.3% accuracy in user authentication tasks with only 13.1% false alarms and 4.96 detection delay.

The experiment results demonstrated that using sensory data collected from the mobile devices, we can model user's motion and posture patterns when they use the devices. Using these models, we can infer user's identity or category. This approach may not only enable more sensing-based personalization application and services, but also introduce privacy risks and security concerns, because such a passive sensing may reveal too much information about the user without the proper consent or permission from them. The potential benefits and risks and their trade-offs deserve further and deeper investigations.

VII. Acknowledgments

This research was supported in part by the CyLab Mobility Research Center at Carnegie Mellon under grants DAAD19-02-1-0389 and W911NF0910273 from the Army Research Office and by the Cisco Research Award. The authors would also like to thank Professor Patrick Tague at Carnegie Mellon University and Dr. Yue Chuan She at Facebook Inc. for their valuable comments and suggestions.

References

[1] M. Baldauf and S. Dustdar. A survey on context-aware systems. *INTERNATIONAL JOURNAL OF AD HOC AND UBIQUITOUS COMPUTING*, page 2004, 2004.

[2] K. Burke. *Language as Symbolic Action*. University of California Press, 1966.

[3] S. Buthpitiya, Y. Zhang, A. Dey, and M. Griss. n-gram geo-trace modeling. In *Proceedings of Ninth International Conference on Pervasive Computing*, San Francisco, CA, June 12-15 2011.

[4] P.-W. Chen, S. K. Chennuru, and Y. Zhang. A language approach to modeling human behavior. In *Proceedings of The seventh international conference on Language Resources and Evaluation (LREC)*, Valletta, Malta, May 19-21 2010.

[5] K. Farrahi and D. Gatica-Perez. Extracting mobile behavioral patterns with the distant n-gram topic model. *Wearable Computers, IEEE International Symposium*, 0:1–8, 2012.

[6] D. Gafurov, K. Helkala, and T. Sondrol. Biometric gait authentication using accelerometer sensor. *Journal of Computers*, 1, 2006.

[7] J. Han, E. Owusu, L. T. Nguyen, A. Perrig, and J. Zhang. Accomplice: Location inference using accelerometers on smartphones. In *COMSNETS*, pages 1–9, 2012.

[8] K. hao Chang, J. Hightower, and B. Kveton. Inferring identity using accelerometers in television remote controls. In *In Proceedings of the International Conference on Pervasive Computing*, 2009.

[9] M. Jakobsson, E. Shi, P. Golle, and R. Chow. Implicit authentication for mobile devices. In *Proceedings of the 4th USENIX conference on Hot topics in security*, HotSec'09, pages 9–9, Berkeley, CA, USA, 2009. USENIX Association.

[10] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 71–84, New York, NY, USA, 2010. ACM.

[11] R. Orr and G. Abowd. The smart floor: A mechanism for natural user identification and tracking. *ACM Press*, 2000.

[12] A. Peacock, X. Ke, and M. Wilkerson. Typing patterns: a key to user identification. *Security Privacy, IEEE*, 2(5):40 –47, sept.-oct. 2004.

[13] A.-D. Schmidt, F. Peters, F. Lamour, and S. Albayrak. Monitoring smartphones for anomaly detection. In *Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*, MOBILWARE '08, pages 40:1–40:6, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[14] C. E. Shannon. A mathematical theory of communications. *Bell Sys. Tech. Journal*, 27:379–423, 623–656, 1948.

[15] E. Shi, Y. Niu, M. Jakobsson, and R. Chow. Implicit authentication through learning user behavior. In *Proceedings of the 13th international conference on Information security*, ISC'10, pages 99–113, Berlin, Heidelberg, 2011. Springer-Verlag.

[16] W. Shi, J. Yang, Y. Jiang, F. Yang, and Y. Xiong. Senguard: Passive user identification on smartphones using multiple sensors. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on*, pages 141 –148, oct. 2011.

[17] L. Van Boven, A. Kamada, and T. Gilovich. The perceiver as perceived: Everyday intuitions about the correspondence bias. *Journal of Personality and Social Psychology*, 77(6):1188–1199, 1999.

[18] J. V. Wertsch. *Mind As Action*. Oxford University Press, USA, 1998.

[19] K. Wrona and L. Gomez. Context-aware security and secure context-awareness in ubiquitous computing environments.

[20] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma. Understanding mobility based on gps data. In *Proceedings of the 10th international conference on Ubiquitous computing*, UbiComp '08, pages 312–321, New York, NY, USA, 2008. ACM.

[21] J. Zhu and Y. Zhang. Towards accountable mobility model: A language approach on user behavior modeling in office wifi networks. In *Proceedings of The IEEE International Conference on Computer Communications and Networks (ICCCN 2011)*, Maui, Hawaii, July 31 - August 4 2011.