Using SVM for User Profiling for Autonomous Smartphone Authentication

Trisha Datta
Department of Computer Science
Princeton University
Princeton, NJ, USA
tdatta@princeton.edu

Kyriakos Manousakis
Wireless Networks and Systems Research
Applied Communication Sciences
Basking Ridge, NJ, USA
kmanousakis@appcomsci.com

***Abstract*—While we have all been warned about viruses attacking our computers and hackers stealing our private information, very few of us realize the similar threat to our phones. With the number of smartphones in use growing each day, we now find ourselves to be a society equipped with devices packed with personal information and small enough to be easily stolen or misplaced. Millions of smartphone users are reporting unauthenticated behavior on their phones, yet many refuse to use passcode protection. Our goal is to use information gathered from the phone, specifically app usage statistics, in order to determine if a user is the actual owner of the smartphone. For this project, we created an app that could record a variety of information from smartphones and their sensors and make simple decisions about whether the person using the phone was the actual owner and lock itself accordingly. Because of time constraints, we looked at data sets from Glasgow Caledonian University and LiveLab at Rice University rather than collecting our own data. We used the LIBSVM library to create two-class SVM models for each of the 34 users in the LiveLab datasets. We then constructed testing datasets of both owner and non-owner data and tested the accuracy of the models. Accuracy rates for all 34 users were for the most part over 85%, and while false positive (identifying the owner as non-owner) rates were sometimes high, these false positive diagnoses would not compromise the security of the phone.**

I. INTRODUCTION

According to a 2013 study, 5.6 million smartphone users experienced "undesirable behavior" on their smartphones [1]. Some examples of "undesirable behavior" are unauthorized text messages or the accessing of accounts without permission. Despite the number of people experiencing undesirable behavior, 64% of smartphone users choose not to protect their phones with passcodes [2] for reasons ranging from the "cumbersome" task of entering a password every time a user wants to use the phone to not being "worried about the risk" [3]. Since the majority of smartphone users are unwilling to lock their phones and are thus susceptible to experiencing undesirable behavior, there is an obvious need to develop technologies to augment smartphone user security.

We address this problem by creating a "sensor fingerprint" unique to every user that can then be used to differentiate between the real owner of the phone and anyone else. These sensor fingerprints can be synthesized from information from a variety of sources that could include the phone's accelerometer, magnetometer, and light sensor; app usage statistics; call information; screen information; Wi-Fi access point information; and GPS location information. We created an Android app that records all of this information, but because of time limitations, we used public datasets from Glasgow Caledonian University and LiveLab at Rice University [4] for analysis, and focused only on app usage. For our analysis, we used supervised machine learning techniques based on SVM (Support Vector Machine) [9] to create a model of user behavior. This model was based on features extracted from the raw data that captured information about the length of time that an app was used by the smartphone user, as well as the time of day during which this usage took place. An important aspect of our study that differentiates our research from past work was the use of information-theoretic techniques to select the most informative features. We used mutual information [6] computation techniques to select the features that would be the most discriminative across users. After we generated user models for different users, we tested them against a large number of other users to see if we could identify user and non-user behavior correctly. Our results showed an average accuracy of 85.8% in identifying users and non-users correctly, with a false positive rate of 37.1% and a false negative rate of 13.8%.

The remainder of this paper is organized as follows. Section 2 describes related work. Section 3 describes the public dataset that we leveraged from Rice's LiveLab, and discusses our technical approach for generating user models in detail. Section 4 discusses our evaluation approach, and provides the results that we obtained when we tested our models against a large number of users. Section 5 provides a conclusion and discussion of future work.

II. RELATED WORK

Many researchers have attempted to address the problem of user profiling for user authentication. Researchers at Glasgow Caledonian University (GCU) also used their own data, data from MIT, and the LiveLab data from Rice University for their experiments. They built temporal and spatial models of user behavior from sensor data probability density functions to create a "detection threshold" [2]; if the phone finds that its current sensor data lead to a value below the threshold, the phone will activate its passcode lock. GCU's models were based on several different kinds of information, such as app

/01•2•3405•166/•47267852)99•:;926••••

information, Wi-Fi information, cell tower information, call history, CPU load, magnetometer data, noise data, light sensor data, rotation sensor data, and accelerometer data. At Princeton University, researchers tried to differentiate users based solely on data from their phones' magnetometer, accelerometer, and orientation sensor [5]. They also used SVM to train and test their models. They achieved accuracy similar to our technique, but they did not report their false positive or false negative rates. Further, they only reported results for 4 users in two different datasets with data collected over a few weeks. The major contributions of our paper are (i) the features used for user profiling, which focus on temporal app usage statistics, (ii) the aggregation of usage statistics over different epochs capturing different times of day, thus capturing temporal characteristics of the data, and (iii) our use of information-theoretic techniques for feature selection, prior to using SVM to create user profiles.

III. Technical Approach

This section describes our technical approach. Section 3.A describes an app that we created to collect data and make simple decisions. Section 3.B describes the datasets that we used from Glasgow Caledonian University and Rice University's LiveLab. Section 3.C details how we constructed our features for SVM training. Section 3.D discusses how we used mutual information techniques to find the most informative features. Section 3.E describes the creation of models using two-class SVM techniques

A. Data Collection and Decision-Making App

We created an Android app to collect data from smartphones; these data include accelerometer data, magnetometer data, light sensor data, application use statistics, call information, screen information, Wi-Fi access point information, and GPS location information. Due to time and device constraints, we then used public datasets that contained similar data to that collected by our app, as we did not have the resources for a large data collection effort. In addition to collecting data, as a proof of concept, the app that we developed also makes very simple decisions about whether the current user of the phone is the legitimate owner of the phone, based on its GPS location and Wi-Fi access points that are within range of the phone. Our app starts with a list of known GPS locations and Wi-Fi access points; if the app detects unknown Wi-Fi access points, it automatically locks and asks the user for the passcode. If the user is able to successfully enter the passcode, thus authenticating himself or herself as the true owner of the phone, these new Wi-Fi access points are added to the list of known Wi-Fi access points. Similarly, if the user is in a place whose distance from all known access points is greater than a certain radius, the phone locks itself and then adds to its list of known locations if successfully unlocked. Of course, this is a rather trivial decision-making algorithm; however, our goal is to replace this algorithm with testing based on user profiles that are constructed as described in the remainder of this section. The purpose of constructing this app was to develop a framework within which we plan to insert more complex decision-making in the future.

B. Description of Dataset

We analyzed a public dataset from LiveLab at Rice University [4]; the LiveLab data were collected from 34 users using iPhones over the course of one year. The LiveLab data contain information regarding which apps were used and for how long, when calls were received or made and the associated phone numbers, when the phone was in sleep mode and for how long, when the phone was charging and for how long, the phone's display status, CPU disk utilization, accelerometer readings, the cell tower to which the phone was connected, the cell signal strength, the associated Wi-Fi access point, the available Wi-Fi access points, when the data logger was running, and web browsing history. The data regarding the accelerometer, CPU disk utilization, cell towers, cell signals, associated Wi-Fi access points, and available Wi-Fi access points were recorded periodically. These periods were originally 15 minutes but could be adjusted by the logger as needed. The rest of the data were event-driven, meaning that every time there was new information regarding that data, it was recorded.

We chose to analyze LiveLab data because its features overlap largely with the ones our app collects. We chose to focus on app usage first in order to create our sensor fingerprints based on some preliminary analysis conducted on another public dataset from Glasgow Caledonian University [2]. These data contained information on Wi-Fi networks, cell towers, app usage, the light sensor, the magnetometer, the rotation sensor, the accelerometer, and device system statistics. The GCU dataset had information from four users over the course of three weeks. The preliminary analysis conducted on the GCU dataset helped us to decide to focus on app usage. Figure 1 shows some results of this preliminary analysis. It shows the app usage statistics for the four users for four different apps. The graph clearly shows that each of the four users uses the apps very differently in terms of the amount of time spent per day using these apps, thereby permitting straightforward user profiling based on app usage statistics. This motivated us to look at app usage to differentiate users.

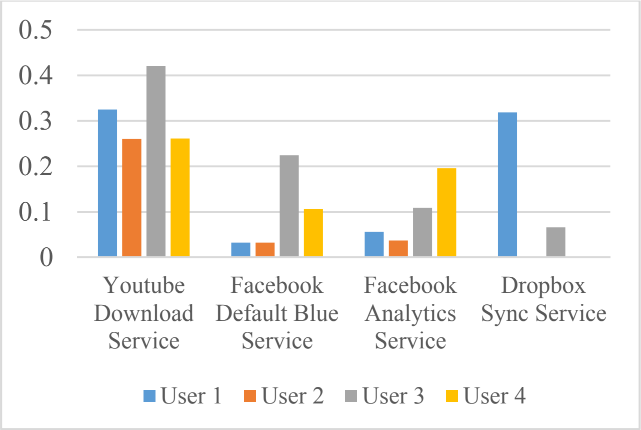

Fig. 1. Percentage of Time Used vs. Selected Apps.

C. Feature Construction

Although our initial analysis was conducted on the GCU dataset, this dataset only contains four users with data collected for these users over a relatively short time period (a few weeks). We therefore conducted more detailed analysis on the LiveLab data. The data on app usage statistics from LiveLab contain the name of the user, the name of the application, the time at which the data was recorded, and the duration in seconds for which the app was in use. For the purposes of this project, we decided to aggregate these data in order to compare usage patterns across different users in terms of their usage volume (total time spent using the app) as well as temporal variations in app usage (i.e., time of day when the app is used). We chose to do this by compiling a list of all the apps used by each user, while taking into account temporal patterns. The temporal patterns were selected to differentiate app usage across different times of day, such as when the user is expected to be asleep, at work or performing other activities, and during evening hours. We therefore divided the day into three eight-hour epochs, and then counted the number of seconds each app was used during these epochs every day. The rationale for dividing the day into three epochs was because we reasoned that people use their phones very differently during the day, evening, and night. For example, those who work probably use social networking apps less frequently during work hours than they do at home or at night. Moreover, many people routinely check certain apps in the morning, such as news or weather apps, that they do not necessarily use throughout the day.

D. *Feature Selection*

The 34 users used a total of 2,302 apps, which means that with the three eight-hour epochs, we had a total of 6,906 features for each user. We wanted to reduce the number of features for several reasons. First, this large number of features would drastically increase the SVM training time. Next, and more importantly, many of the apps were never used or were only used by a user a few times and would thus not be useful in differentiating users. Furthermore, reducing the number of features would reduce noise introduced by considering hundreds of apps that are not used with any regularity, and reduction of noise would thus increase accuracy. In order to narrow down the number of features while retaining the most informative features, we decided to use a concept from information theory called mutual information [6]. Mutual information uses the probability densities of different features to calculate the features best suited to discriminate between different users. Essentially, this technique studies the statistical dependence of a variable on others [7]; by studying these dependencies, we are able to select features that are likely the best suited to differentiate users. From [7], if we have data $\{(\mathbf{x}_i, y_i)\}_{i=1}$ where $\mathbf{x}_i$ is a d-dimensional vector, mutual information is computed as:

$$I(x_j) = \int_{x_j} \int_y p(x_j, y) \log \frac{p(x_j, y)}{p(x_j) p(y)} dx\, dy, \qquad (1)\ [7]$$

where $p(x_j)$ and $p(y)$ are the probability densities of x_j and y, and $p(x_j, y)$ is the joint density. It is difficult to estimate (1) in the continuous case since the densities $p(x_j)$, $p(y)$, and $p(x_j, y)$ are unknown and hard to estimate. Fortunately, we are dealing with discrete data and can therefore replace the above integral with a sum, thereby making the computation simpler. We approximate probabilities by computing observed frequencies calculated from the data. The discrete computation is [7]:

$$I(x_j) = \sum_{x_j} \sum_y P(X = x_j, Y = y) \log \frac{P(X = x_j, Y = y)}{P(X = x_j) P(Y = y)}, \qquad (2)$$

We performed the mutual information calculations on one user and then used the features with the highest mutual information values for training.

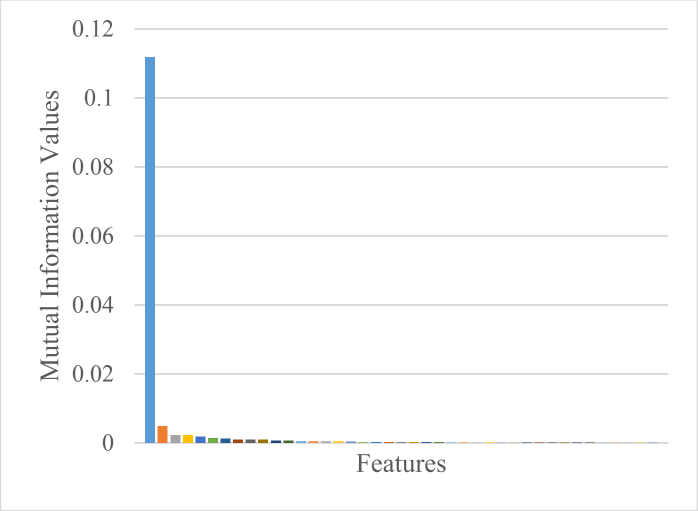

Fig. 2. Percentage of Time Used vs. Selected Apps.

Figure 2 shows the 41 highest mutual information values, obtained; the corresponding 41 features were the ones that we used for SVM training. While Figure 2 shows all 41 values, Figure 3 excludes the highest value in order to show the other values more clearly.

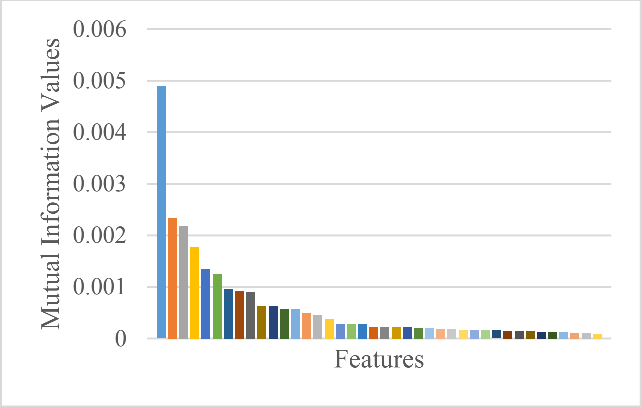

Fig. 3. Highest Mutual Information Values vs. Features, *without* highest value

E. *SVM Training*

We used the LIBSVM library [8] for our analysis. We considered the use of both one-class [10] and two-class SVM [9] for this work. One-class SVM is essentially a clustering method that defines all points as either part of the class or not part of the class and is useful when training data is available for the user being profiled and a representative sample is not available for the rest of the users. All of the training data belongs to one class, that of the user being profiled. One-class SVM at first seemed like the obvious choice for our purposes because theoretically we would only have information from the actual owner of the phone. However, one-class SVM presents several challenges of its own. One-class SVM models depend on two parameters, ν and γ, and finding the right combination of these two parameters can prove to be time-consuming and can produce unsatisfactory results. We attempted to use one-class SVM, and high cross-validation rates (80-90%) usually led to high false negative rates (here, we take a false negative to be a non-user data point being classified as the real owner). High false negative rates are much more risky than high false positive rates, because they lead to the erroneous belief that the rightful owner is in possession of the phone.

We then decided to experiment with two-class SVM. For training, two-class SVM takes data that represent two different classes and creates a model that classifies new data points as one of the two classes. In order to use two-class SVM, we created files where half of the data corresponded to a certain user and the other half of the data was randomly selected from all the other users. The half of the data that corresponded to a certain user was only half of the data that we had on that user. The other half was used for testing. To find the best parameters for creating a model, we used a grid search for the values of C and γ using training data for User 1. After we found values that gave us reasonably accurate results, we used these parameters for the rest of the users. We decided to use the radial basis function kernel because it is suited for relatively small numbers of features and training instances.

IV. Evaluation

In order to evaluate our models, we constructed testing data sets and then used our models to determine whether the data in the testing sets belonged to the user to which the model belonged.

A. *Testing Data*

To test our models, we created test data sets from the remaining LiveLab data. These remaining data consisted of the remaining half of the data from the particular user on which the model was trained as well as the data from the rest of the users that did not appear in the training data. We used our models to predict whether or not the data in these test data sets were from the actual owner of the phone.

B. *Analyzing Data*

Figure 4 displays our results. The blue bars indicate the overall accuracy rate on the test data; the orange bars indicate the false positive rate (number of user data points predicted as non-user); and the gray bars indicate the false negative rate (number of non-user data points predicted as user). As explained earlier, because of time limitations, we only used mutual information to select features for User 1. We then used the same features for the rest of the users. This could be one reason for the higher false positive rates obtained for some of the models.

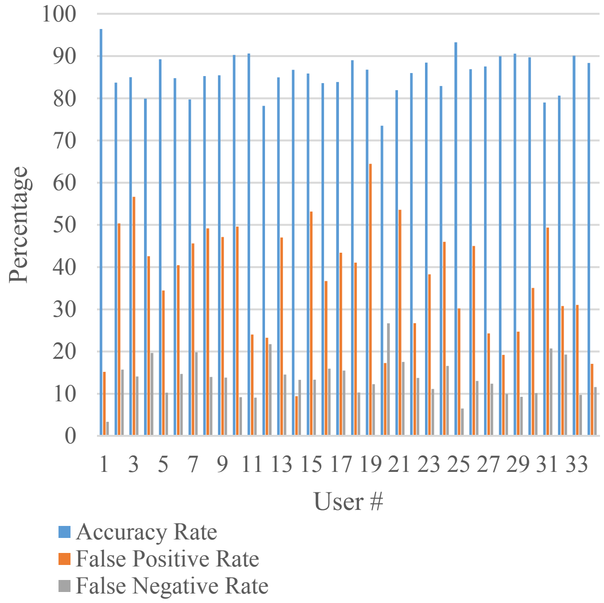

Fig. 4. Accuracy Rates for Different Users

The high false positive rates, while certainly a problem, are actually not as problematic as the high false negative rate given by the one-class SVM models. This is because the false positive rate indicates that the behavior of the phone's actual owner is not being recognized as legitimate behavior; when our app detects this, it will automatically lock. While the phone's owner may be annoyed at having to enter the passcode, the security of the phone is not compromised. Conversely, the one-class SVM models with the high false negative rates are far more risky because they would recognize the behavior of someone other than the actual owner as legitimate behavior and thus allow unauthorized users to access the phone.

V. Summary and Future Work

Preliminary tests indicate a high success rate in detecting when someone other than the owner of the phone is using it. Although we had a fairly high false positive rate, which we plan to address in future work, in real life, false positives would not compromise security.

Our future work plans include performing mutual information computations for all users in order to find the best features for differentiating users for all users. We also plan to analyze additional information, such as Wi-Fi access points and GPS location, and combine the output of multiple classifiers to increase accuracy. We also plan to conduct this future analysis on data that we collect ourselves.

References

[1] "Keep Your Phone Safe: How to Protect Yourself From Wireless Threats." Consumer Reports. Consumer Reports, June. 2013. Web.

[2] Kayacik H. G., Just M., Baillie L., Aspinall D., Micallef N., "Data Driven Authentication: On the Effectiveness of User Behaviour Modelling with Mobile Device Sensors", Proceedings of the 3rd Mobile Security Technologies Workshop , Held as part of the IEEE S&P Symposium, (MoST-2014), May 2014.

[3] "Survey Shows Smartphone Users Choose Convenience Over Security." Confident Technologies. Confident Technologies, Inc., 28 Sep. 2011.

[4] Clayton Shepard, Ahmad Rahmati, Chad Tossell, Lin Zhong, and Phillip Kortum, "LiveLab: measuring wireless networks and smartphone users in the field", in ACM SIGMETRICS Perform. Eval. Rev., vol. 38, no. 3, December 2010.

[5] W. H. Lee and R. B. Lee, "Multi-sensor authentication to improve smartphone security," in Proceedings of the 1st International Conference on Information Systems Security and Privacy, pp. 270–280, Feburary 2015.

[6] T. M. Cover and J. A. Thomas. Elements of Information Theory. New Jersey: Wiley 2006.

[7] Akshay Vashist and Rauf Izmailov. Evaluating Information Content and Information Gain. Technical Report, Applied Communication Sciences, 11/11/2013.

[8] Chih-Chung Chang and Chih-Jen Lin. LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[9] C. Cortes and V. Vapnik, "Support-vector networks". Machine Learning 20 (3): 273, 1995.

[10] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution. Neural Computation, 13(7):1443–1472, 2001.