DeepAuth: A Framework for Continuous User Re-authentication in Mobile Apps

Sara Amini*
University of Illinois at Chicago
Chicago, IL, USA
samini3@uic.edu

Vahid Noroozi
University of Illinois at Chicago
Chicago, IL, USA
vnoroo2@uic.edu

Amit Pande
Data Science, Target Corporation
Minneapolis, MN, USA
amit.pande@target.com

Satyajit Gupte
Data Science, Target Corporation
Minneapolis, MN, USA
satyajit.gupte@target.com

Philip S. Yu
University of Illinois at Chicago
Chicago, IL, USA
psyu@uic.edu

Chris Kanich
University of Illinois at Chicago
Chicago, IL, USA
ckanich@uic.edu

ABSTRACT

With the increasing volume of transactions taking place online, mobile fraud has also increased. Mobile applications often authenticate the user only at install time. The user may then remain logged in for hours or weeks. Any unauthorized access may lead to financial, criminal or privacy losses. In this work, we leverage currently available built-in motion sensors in smartphones to learn users' behavioral characteristics while interacting with the mobile device to provide an implicit re-authentication mechanism that enables a frictionless and secure user experience in the application. This approach improves the generality as well as power efficiency of the authentication mechanism compared to using the camera feed which involves (a) specific hardware, (b) higher battery usage and (c) privacy concerns. We present DeepAuth as a generic framework for re-authenticating users in a mobile app. In our approach, we use time and frequency domain features extracted from motion sensors and a long short-term memory (LSTM) model with negative sampling to build a re-authentication framework. The framework is able to re-authenticate a user with 96.70% accuracy in 20 seconds from a set of data collected from 47 volunteers.

CCS CONCEPTS

• **Security and privacy → Authentication**; • **Computing methodologies → Neural networks**;

KEYWORDS

user re-authentication; user fingerprinting; mobile motion sensors; user identification

ACM Reference Format:

Sara Amini, Vahid Noroozi, Amit Pande, Satyajit Gupte, Philip S. Yu, and Chris Kanich. 2018. DeepAuth: A Framework for Continuous User Reauthentication in Mobile Apps. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM'18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3269206.3272034

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3272034

1 INTRODUCTION

With the widespread use of mobile devices, a large number of mobile users use their devices for commercial transactions, to access sensitive information, and to store and share personal information. Typical explicit authentication mechanisms ask the user to setup a pass code or use biometrics such as fingerprint or face recognition to sign-in the user. A critical assumption made by such schemes is that only the legitimate user will use the device once having unlocked it and the device will be locked again immediately after the session. Many shopping apps may require the user to re-authenticate before making an online transaction, even if this adds friction to the interaction. Outside of these explicit re-authentications, any intermediate change in user identity will go undetected. Without re-requesting user identification information, an unauthorized user may access sensitive information on the user's mobile device if the user's device is left unlocked and unattended or even hacked. Such unauthorized access may lead to unauthorized use of the application, including purchases, settings changes, and identity theft. In another scenario, one may log into an app and allow a friend to temporarily use it, but may not be comfortable with them accessing one's credentials or history of purchases. Market research indicates that shopping apps are a prime target of mobile fraud [1]. Implicit Authentication (IA) schemes can be used to enhance user experience alongside security by verifying user identity. User identification/authentication is indeed the topmost priority for 53% of digital companies [1].

Traditional authentication methods utilize identifying information such as passwords, touch fingerprints and face recognition [19, 24] to verify the user when accessing sensitive and security-critical parts of an application, for example when checking purchase history or financial and application credential information. However, such methods suffer from the following drawbacks: 1)

*This work is an extension of work done by Ms. Amini in her summer internship at Target.

They heavily rely on the user's direct input and cooperation which inhibits user experience. These inhibitions are so unpalatable to users that 64% users do not use passwords or PINs on their smartphones [10]. 2) Explicit authentication incurs an overhead and increases the latency of device interactions and as a result, degrade the usability of the applications [16, 22]. 3) Although biometric authentication techniques are less frustrating than conventional methods, they still suffer from latency in that they still require the user's direct involvement, and many mobile devices still lack biometric authentication technology. 4) Continuous authentication via technologies like FaceID may require the user to give an app explicit permission to access camera ubiquitously, and would require constant face visibility and camera usage, thus incurring an additional energy usage penalty, constraint and risk to privacy. IA schemes track implicit user activity to use inputs implicitly derived from a user's mobile session. They provide users with a middle-ground, where both usability and security can be achieved. This work proposes a new deep-learning framework to implicitly authenticate users in the course of browsing a shopping app.

Apart from continuous user authentication, this approach can also be used for novel personalization and product recommendation opportunities for online retailers when there is a switch between legitimate users of the same device. For example, over time, a user may have multiple legitimate profiles based on his shopping behaviors. When his partner uses the app, the app may ask to re-login first time and subsequently build a second profile for her. When she uses the app again, she will receive recommendations personalized for her. Similarly, a user may have two profiles simply based on his browsing behavior. For example, he may like to browse for clothes and home accessories at leisure while commuting in a bus, but in another context he may want to pick quick groceries while returning home from the office.

Contributions

In this work, we focus our analysis on continuous user authentication which monitors motion sensor logs and locks the user out of the app once an unauthorized user is detected. Consequently, it is necessary for mobile applications to robustly identify the user. Figure 1 shows a snapshot of the Target shopping app. In our data collection phase, the users were asked to browse a few items on the Target shopping app (as they would routinely do) on a mobile device which collects motion sensor information alongside site browsing behavior. No information about which items or content were being viewed was used to build the models.

In this paper, we present DeepAuth, a novel framework for implicit and continuous mobile user authentication via mobile motion sensors. We leverage deep learning [14] as a computational framework since it has attracted significant attention in the past few years due to its capability to automatically extract features directly from raw data. We use recurrent-neural networks, specifically Long Short-Term Memory (LSTM) [9], which has shown promising performance on sequential data due to its capability of modeling highly non-linear temporal relations. Moreover, we introduce a novel method for sensor data pre-processing by utilizing both time and frequency domain features.

The main contributions of this work are as follows:

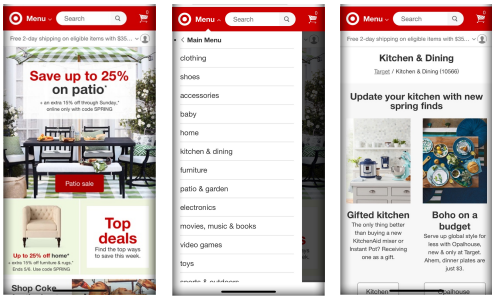


Figure 1: Target Shopping App

(1) We present a LSTM-based authentication framework, called DeepAuth, that leverages a user's passive behavior while shopping online to continuously re-authenticate the user, providing security without compromising usability.
(2) We evaluate DeepAuth on a dataset collected from volunteers in which mobile motion sensors were recorded, while browsing and shopping on the Target.com mobile website to emulate actual users' behavior in real-world browsing and shopping scenarios. Only motion sensor information is captured during browse behavior of 47 volunteers.
(3) We compare DeepAuth to state-of-the-art classification methods such as SVM, Random Forest, Logistic Regression and Gradient Boosting Classifier. DeepAuth can re-authenticate a user with 96.70% accuracy within only 20 seconds, while Gradient Boosting Classifier provides the best accuracy among all baselines with 89.57% accuracy.

The rest of this paper is organized as follows. In Section 2 we give an overview of the related works. In Section 3 the system overview is illustrated. In Section 4, we describe DeepAuth in detail followed by Section 5 where the details on data collection are explained. The evaluation is justified in Section 6, and we conclude and discuss the future work in Section 7.

2 RELATED WORK

Due to growing popularity of smartphones with more than two billion users all around the world, their built-in sensors data has been excessively exploited in various fields of research including security and privacy. User identification and continuous user authentication are two representatives of challenging mobile sensing problems in security and privacy research areas. Based on the user's distinguishable behavioral patterns inferred from sensors data [3]. Continuous user authentication aims to verify whether the current user is the actual legitimate user or not (see Figure 2a), whereas in user identification problem the goal is to identify the current user of the mobile device correctly among all users (see Figure 2b).

Traditional authentication schemes require users to provide a secret in combination of password, session cookies, or both. This requirement is more relaxed in re-authentication procedure to provide a responsive and more convenient experience. Even though the prevalence of biometric authentication in smartphones such as Touch ID has made an effort in creating a robust authentication with less effort, any re-authentication request interrupts user activity and diminishes seamless experience. Fortunately, abundance of

(a) User authentication (b) User identification

Figure 2: Two representatives of challenging mobile sensing problems

input sensors in smartphones and advances in machine learning techniques have paved the way to behavior-based re-authentication methods which requires no user interaction. In this work, our focus is on the former to provide a seamless re-authentication experience to the users.

Previous works have studied context-based user authentication using various smartphone inputs. Lee et al. [15] proposed iAuth, an implicit and continuous user authentication scheme using sensor data of multiple devices such as smartwatches and smartphones. A similar paradigm was suggested by Kayacik et al. [12] to detect the deviation of behavior, e.g. unprivileged physical access, from the learned normal behavior of owner. Their incremental training paradigm learns user's behavior using temporal and spatial data. In contrast to our approach, suggested paradigm by [12] requires a long training time to be able to detect unauthorized users. In [17], authors utilized common classification method to learn user's touch pattern to achieve continuous re-authentication and prevent unauthorized access to Smartphone. Bo et al. [3] proposed a touch-based biometric model using combination of coordinate, pressure and duration of touch behavior and motion data from sensors to train a two-class SVM classifier and eventually identify the Smartphone's owner. There exist some neural network based approaches which follow Siamese architecture [5, 20] and can handle verification tasks. While Siamese architecture is widely used for learning discriminative representations [2] and can handle large number of classes in one single model, their accuracy is not satisfactory for authentication task which needs high accuracy.

Key differences between DeepAuth and the aforementioned studies lie in the following: 1) DeepAuth offers a novel pre-processing method for the sequential sensors data by utilizing both time and frequency domain data. 2) DeepAuth only uses two sensors: accelerometer and gyroscope. 3) It utilizes windowing technique as well as LSTM in order to detect both micro and macro distinguishable behavioral patterns of the users. 4) It is trained and evaluated on real-world data which includes data from users, while interacting and browsing on a shopping app (Target app). 5) It is trained on a small dataset; there is only 10-13 minutes data available from each user which make the authentication task harder.

3 SYSTEM OVERVIEW

An overview of our intended system deployment is shown in Figure 3. In the *enrollment phase* of the system, whenever a new user signs in to the mobile application, the system continually monitors and collects the sensors' data from the user's mobile device and saves them in the user's data profile in the system. For this paper, we collected data from users while they were browsing on Target

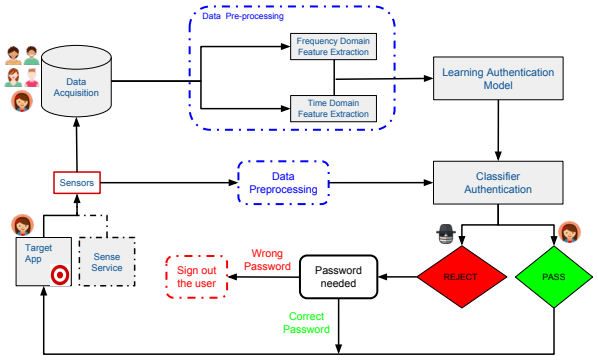


Figure 3: Authentication system overview

app. In order to collect sensor data, we developed a secondary app which could run in the background while the Target app is running in the front and collect sensors data. More details on this will be explained in Section 5. In the *continuous authentication phase* the sensor data is pre-processed and will be input to the machine learning algorithm to learn the authentication model for the user. Once the model is learned, in the *post-authentication phase*, any incoming sensor data from the user's mobile device is continuously monitored. If the incoming data passes the authentication model criteria and the model verifies the current user as the legitimate user, the user can keep working within the app and accessing the sensitive parts of the application. However, if the incoming data fails the authentication model tests, the system locks out the user from the app and asks for an alternative authentication method, such as a password, from the user.

4 METHODOLOGY

4.1 Model Definition

Current mobile devices are equipped with various embedded sensors such as global positioning system (GPS), accelerometer, gyroscope, magnetometer, and others. These sensors are used in conjunction with mobile apps to perform diverse activities for numerous user purposes. For example, accelerometer and gyroscope which identify the movement of the mobile device can be used in game apps. The data derived by such sensors can be utilized to detect individual users' behavioral patterns as they capture interaction and activity of the mobile users and produce a series of samples over time. In this study, we recognize such patterns using the data obtained from the sensors of the users' smartphones.

A segment of data is defined as a series of measurements for a T period of time. Such measurements can be recorded in the form of a matrix $S = [s_1, \ldots, s_T]$, where vector $s_t \in \mathbb{R}^{D\times 1}$ signifies the measurements at t^{th} time step, and D is the number of measurements. The training set is represented as $\mathcal{X} = \{S_i\}_{i=1}^{M}$, where S_i is a training segment sample, $M = N_p + N_n$ is the total number of training segments consisting of N_p segments pertaining to the legitimate user (positive segments) and N_n segments of unauthorized users (negative segments). The label set denoted by $\mathcal{Y} = \{y^i | y^i \in \{pos, neg\}\}_{i=1}^{M}$ specifies the class of each segment.

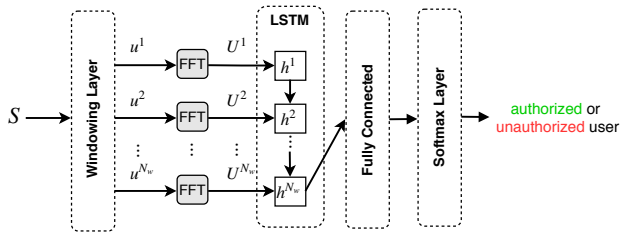


Figure 4: Main architecture of DeepAuth framework.

A positive class indicates that the segment belongs to the authorized user and a negative class indicates that it is pertaining to an unauthorized user.

Our aim is to learn a non-linear function $f_\theta(S_i) : \mathbb{R}^{D\times T} \rightarrow \{pos, neg\}$ parameterized by θ that predicts the class of the data sample S_i. In other words, function $f_\theta(S_i)$ verifies if the segment sample S_i belongs to the authorized user or not.

4.2 Authentication Architecture

The proposed model consists of five primary components: pre-processing and sampling, windowing, feature extraction, modeling, and finally an output layer. Data gathered from different sensors are concatenated to form segments. Consequently, we build positive and negative segments via sampling. In the windowing layer, such segments are divided into smaller unites. These sub-segments which are originally in time domain are then mapped into frequency domain in the feature extraction layer by utilizing the Discrete Fourier Transform (DFT) [4]. The modeling component is a neural network which learns the behavioral patterns for each user and maps the segment to a fixed-size embedding space. The final output layer consists of a fully-connected layer to classify the segment into positive or negative class. An independent model is trained for each user separately. We discuss the details of these components further in the following subsections. A schematic representation of the main architecture of the DeepAuth framework starting from the windowing layer is shown in Figure 4.

4.2.1 Data Pre-processing and Sampling Layer. Most motion sensor measurements, by default, are recorded with predefined sampling rates and at different offsets that are not synchronized across different sensors. As we need synchronized data points, we down-sample and synchronize the samples from different sensors by dividing the whole sequence of samples into smaller intervals and choosing the mean of all values in a given interval as the representative of it. Subsequently, at each time step in the whole time interval, we concatenate these measurements from all sensors to create the sample vector for that time step.

We assume that there are K sensors on the phone, and the k^{th} sensor generates d_k different samples at each time step. For example, each sample of the accelerometer sensor includes measurements from three dimensions x, y and z which specifies the acceleration of the phone movement in three dimensional space. We denote the total number of measurements in each time step as $D = \sum_{k=1}^{K} d_k$. Such measurements are then split into fixed-size segments of size T time steps (see Section 4.1). Since, sensors' samples may be very long and have diverse length among different users. We segment the data by moving a fixed-size window of size T over the sequential data with a predefined shift T_Δ to build overlapping fixed-sized segments. Thus for each user, we have a set of segments with size $D\times T$. The number of segments in each user's sequence depends on the length of the input data for that user denoted as N_p to represent positive segments of the user. To train the model for a user, we also need to provide the model with negative segments which are those provided by other users. We use a controlled random sampling to create the group of negative segments for each user by randomly selecting segments of other users. In the sampling process, the ratio of negative to positive segments is kept to a fixed number r, which we call the negative factor. All segments, both positive and negative, are then passed into the windowing layer to build the training data.

4.2.2 Windowing Layer. The segments resulted from the pre-processing and sampling layer are then passed to the windowing layer. Individual measurements of segments do not reveal sufficient information about the user's behavior to distinguish authorized users from unauthorized ones. Rather, these measurements must be explored sequentially to reveal behavioral patterns of the user, which include some measurements related to various actions like touch events. We posit that some such behavioral patterns involve a small portion of measurements which we call micro patterns (e.g. individual tap force or swipe length), whereas some include a bigger portion of measurements reflected by the inter-segment relationship of the micro patterns denoted as macro patterns (e.g. long term average angle at which the phone is most often held).

In the windowing layer, segments are prepared for the detection of micro and macro patterns. We leverage windowing technique to divide segments into smaller windows to detect micro patterns. The windowing process is done by moving a fixed-size window with size l over each segment with a pre-defined shift l_Δ to produce windows, i.e. $Windowing(S) = \{u^i\}_{i=1}^{N_w}$, where S is a segment, $N_w = \lfloor \frac{T}{l_\Delta} \rfloor$ is the number of windows in segment S, and $u^i \in \mathbb{R}^{D\times l}$ is the i^{th} window of the segment S.

Macro patterns which reflect the inter-segment relationships between subsequent windows are modelled by the LSTM layer of DeepAuth. More details about this step will be discussed further in Section 4.2.4.

4.2.3 Feature Extraction. The outputs of the windowing layer are windows of sensors measurements in their original domain which is time. However, frequency domain data has some advantages when it comes to working with noisy mobile sensors' sequential data [21]. These advantages include being able to handle and remove noise, as well as better detection of distinctive behavioral patterns within sequential data.

Discrete Fourier Transform (DFT) is employed to convert the time domain signals to frequency domain signals. The N points one-dimensional DFT of a signal is defined in the form:

$$X_k = \sum_{n=0}^{N-1} x[n] \times \exp\left\{-2\pi i(\frac{nk}{N})\right\} \quad k = 0, \ldots, N-1 \qquad (1)$$

where $x[0], x[1], \ldots, x[N-1]$ are N discrete points of a signal in time domain and $X_0, X_1, \ldots, X_{N-1}$ are N points of the signal in

frequency domain. Fast Fourier Transform (FFT) is one of the most common and numerically efficient algorithms to calculate DFT.

FFT is employed on each dimension of the window to convert their measurements in the time domain with size $D \times l$ to the frequency domain. We consider only half of the spectrum as they are symmetric. Subsequently, the output FFT vectors are concatenated to form a feature vector of size $\frac{l}{2} \times D$. This gives us a sequence of windows $\{U^i\}_{i=1}^{N_w}$ for segment S in the frequency domain, where U^i is the frequency form of window u^i. Putting all of these vectors into a two dimensional matrix gives S^f with dimension $(\frac{l}{2}D) \times N_w$, the frequency domain of segment S. Time and Frequency domain windows are concatenated and passed into LSTM layer.

4.2.4 LSTM Layer. The main goal of DeepAuth is to distinguish different users from one another via their distinct behavioral patterns while interacting with the mobile device. As discussed earlier in Section 4.2.2, we divide such patterns into two categories: micro and macro patterns. Micro patterns are revealed in the windowing layer, whereas the relationships between subsequent segments (macro patterns) are modelled by utilizing a recurrent layer.

Recurrent Neural Networks (RNNs) have attracted significant attention recently because of their simplicity and power in sequence learning, prediction and classification. They have been used in various fields of research from natural language processing [6] to speech recognition [7]. However, they suffer from some serious limitations such as vanishing gradient [8] and incapability in capturing long-term dependencies.

We leverage Long Short-Term Memory (LSTM) [9] to detect macro patterns. LSTM is a variant of RNNs which was intentionally designed to overcome RNNs' aforementioned drawbacks by having a longer memory. Specifically, LSTM outperforms other versions of RNNs when it comes to sequential data with larger time intervals due to its capability of learning long range dependencies through its use of memory cell units and gating mechanism [11].

LSTM is used to map the input segments into a sequence of lower-dimensional feature vectors via learning a non-linear embedding function $f_\theta(.)$ where θ is the set of all parameters of the model. The output feature vectors are denoted as $\left[h_1, h_2, \ldots, h_{N_w}\right]$, where each h_i corresponds to the i^{th} input sequence and has a predefined length of C.

4.2.5 Output Layer. The output of the recurrent layer is a sequence of feature vectors $\left[h_1, h_2, \ldots, h_{N_w}\right]$. The output of the last layer, h_{N_w}, will then be fed into the output layer which includes a fully connected layer with one neuron. The output of the single neuron of the fully connected layer will be fed into the sigmoid layer to generate the predicted category of the data (positive or negative). The loss function on the output of last layer is a binary cross-entropy function denoted as L.

4.3 Training DeepAuth

We train the model as a whole using back-propagation with respect to the binary cross-entropy loss function. Given a set of M training samples for a user, we optimize the model's loss function using an adaptive version of the stochastic gradient descent method called Adam [13]. Moreover, we apply dropout [23] techniques to prevent overfitting. The training procedure of DeepAuth is shown in Algorithm 1.

Algorithm 1: Training Procedure of DeepAuth

Input: Training set: $\mathcal{X} = \{S_i\}_{i=1}^{M}$,
Label set: $\mathcal{Y} = \{y^i | y^i \in \{pos, neg\}\}_{i=1}^{M}$,
Number of epochs: K
Batch size: m
Output: Model's parameter: θ
Number of batches is calculated by $B = M/m$
for $k = 1, 2, \ldots, K$ **do**
  for $b = 1, 2, \ldots, B$ **do**
    b^{th} batch is generated randomly;
    Feedforward propagation of the b^{th} batch;
    Calculate $L^{(b)}$;
    Estimate gradients $\frac{\partial L^{(b)}}{\partial \theta_k}$ via backpropagation;
    Compute θ_{k+1} using Adam;
  end
end
return θ_K

5 DATA COLLECTION

To collect sensor data from the mobile device we develop an app called *Sense Service* using Android Studio which contains a Java code snippet/activity that accesses data from the device sensors (such as accelerometer and gyroscope) and writes those data into a file in the mobile device. In our study, we used Nexus 5X and had the *Sense Service* app installed and running in the background, while Chrome was running in the front when handed to the users. Users were asked to login to *target.com* on Chrome browser using the provided pseudo username and password. Each participant was asked to sit on a chair and interact with Target app about 10-13 minutes, thus emulating the real-world online shopping settings. They were asked to browse for a few items and try to add a couple items that they really want to buy to the shopping cart in order to capture their real interacting behavior with the mobile device while shopping on the Target app. Optional items were provided to participants in case they did not have anything special in mind to browse.

In total, we had 47 volunteers. The volunteers did not login using their personal information but using pseudonyms and used the rooted device provided to them (An unrooted device does not allow a secondary app to collect motion sensor or touch information while the primary app is running on screen). Collected motion sensors are as follows:

- **Accelerometer:** Measure the acceleration force that applied to the device, including force of gravity in three axes (X, Y, Z)
- **Gyroscope:** Measure the device's rotation in three axes (X, Y, Z).

The sampling rate for accelerometer and gyroscope data stream is 100 Hz in the collected dataset. After synchronizing and down-sampling, the resolution of the data as input into the training and evaluation processes is 50 Hz. The dataset is named TargetAuth dataset.

6 EVALUATION

6.1 Baselines

We experiment with different machine learning methods for classification tasks to explore the effectiveness of our model. The baselines used in these experiments are as follows: 1) Support Vector Machine (2-SVM), 2) Random Forest (RF), 3) Logistic Regression (LR), and 4) Gradient Boosting Classifier (GBC). The problem is modeled as a two-class classification task. For each user, one model is trained on positive and negative segments to predict the class of the segment.

6.2 Experimental Settings

For each user, DeepAuth is trained on 70% of the segments, validated on 15% of them and tested on the remaining 15% of the segments. After synchronizing and down-sampling the data, segmentation step is done over the sequential data to generate overlapping segments of size T with segment shift of T_Δ. We experimented with different values of segment shift T_Δ and determined that when it is equal to $\frac{1}{5}$ of the segment size T, the accuracy is at its best. We fix the negative factor $r = 1$ to create an even ratio of positive and negative segments in the training, validation and test phases. All segments are normalized to have zero mean with variance of one by calculating the mean and variance on the training data.

We use the default parameter of Adam optimizer as provided in the original paper [13]. Batch size is 512 for all the training of the neural networks. We experimented with different values of the C parameter (from 5 to 100) and selected $C = 25$ which showed the best result in most scenarios on the validation set. Note that C is the hidden size of the LSTM which is the length of the feature vectors. Dropout of the LSTM is set to 0.2. We experimented with different window sizes:$\{20, 40, 100, 200, 500\}$ which are equivalent to $\{0.4, 0.8, 2, 4, 10\}$ seconds and selected $l = 100$. We experimented with different variants of window shift and determined that when it is equal to window size, the accuracy is at its best, i.e. $l_\Delta = l$. For all baselines, we set identical values for segment size, window size and window shift, while segment shift is set to $T_\Delta = \frac{1}{5} \times T$. The C parameter for 2-SVM, RF, LR, GBC is set to $0.1, 200, 100, 100$ respectively.

6.3 Accuracy Metrics

We consider two evaluation scenarios in our experiments. First, we assume that the current user is the actual legitimate user. Second, we consider the scenario where the current user is not the actual legitimate user. Therefore, we define two accuracy metrics to report the performance of re-authenticating users:

- **Negative accuracy (N):** This metric refers to the ratio of correctly detected unauthorized users to the total number of invalid requests made by imposters trying to access the system. The N accuracy is calculated for each user and their mean is reported as the overall N accuracy.
- **Positive accuracy (P):** This is the ratio of correctly classified authorized user to the total number of valid requests made by legitimate uses trying to access the system. The P accuracy is calculated for each user and their mean is reported as the overall P accuracy.
- $F1$ **accuracy:** We also report the harmonic mean of negative and positive accuracy to show the overall performance which is in the form of: $F1 = 2 \cdot \frac{P \cdot N}{P+N}$
- **Area under curve of ROC (AUC):** AUC is also calculated for each user and their mean is reported as the overall AUC.

6.4 Performance Evaluation

The performance of DeepAuth and all baselines are presented in Table 1. The results are reported for three different numbers of segments sizes:$\{500, 1000, 1500\}$ which are equivalent to 10, 20 and 30 seconds respectively. The best accuracy for each case is depicted in bold. As it is shown in the table, DeepAuth outperforms SVM, LR, RF, and GBC. One of the advantages of DeepAuth over linear techniques like SVM and LR is that it leverages a recurrent layer which is capable of detecting both micro and macro patterns as discussed earlier in Section 4 while other baselines can only detect macro patterns. DeepAuth can model the sequential information in the segments while other baselines are linear non-sequential modeling techniques which ignore the sequential pattern of the data.

Among other baselines, RF and GBC outperform LR and SVM. This observation is consistent with many other studies' discovery that ensemble decision trees perform better compared to regression and SVM models on classification problems.

6.5 Sensor Analysis

To further investigate the suitability of using motion sensor measurements to distinguish different users, we perform an empirical sensor analysis by plotting the sensor data streams collected in the TargetAuth dataset. Figure 5 shows the accelerometer sensor streams corresponding to its three dimensions: x, y and z. We randomly select two users among all users in the data set and randomly pick two signal screenshots from the same user and one signal screenshot corresponding to the other user for comparison.

The screenshots of the accelerometer sensor signal for the same user are more similar along dimensions x and y compared to those pertaining to different users which confirms the intuition of using such information to differentiate users. It also shows that accelerometer measurements along the z dimension may not be substantially helpful in distinguishing users because all records of these users are visually similar.

While Figure 5 shows the preliminary results of the effectiveness of the accelerometer sensors data streams in distinguishing users, we explore the impact of both accelerometer and gyroscope motion sensors more precisely on DeepAuth performance. As shown in Table 2, using more than one source of data helps the performance and leads to higher accuracy.

6.6 Feature Analysis

In this section, we evaluate the performance of the DeepAuth in terms of accuracy when it comes to different feature domains: time

Table 1: Performance of different methods on TargetAuth dataset in terms of accuracy for different segment sizes.

Segment Size	500				1000				1500			
Performance	N	P	$F1$	AUC	N	P	$F1$	AUC	N	P	$F1$	AUC
DeepAuth	**92.73**	**99.20**	**95.85**	**99.05**	**94.37**	**99.16**	**96.70**	**99.19**	**93.52**	**98.22**	**95.81**	**98.61**
2-SVM	91.40	73.07	81.21	79.65	93.51	71.64	81.13	78.85	92.95	68.74	79.03	76.21
RF	89.55	85.43	87.44	87.53	90.11	85.68	87.84	87.78	87.50	84.63	86.04	85.98
LR	91.39	73.22	81.30	79.75	93.57	71.43	81.01	78.66	93.33	68.35	78.91	75.91
GBC	92.72	87.58	90.08	89.98	92.32	86.97	89.57	89.40	89.52	84.78	87.09	86.89

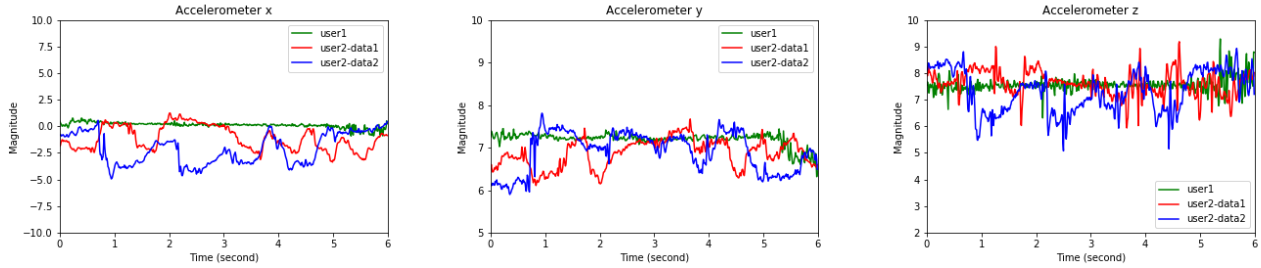


Figure 5: Preliminary results for Accelerometer x, y and z

Table 2: Performance of DeepAuth for different input sensors.

Segment Size	500				1000			
Performance	N	P	$F1$	AUC	N	P	$F1$	AUC
DeepAuth(A+G)	**92.73**	**99.20**	**95.85**	**99.05**	**94.37**	**99.16**	**96.71**	**99.19**
DeepAuth(A)	92.77	99.16	95.86	99.02	94.51	99.02	96.70	99.16
DeepAuth(G)	92.70	99.20	95.84	98.99	94.16	99.16	96.59	99.11

and frequency. In Table 3, we show the accuracy of our model using three different feature sets: 1) time domain features (DeepAuth(t)), 2) frequency domain features (DeepAuth(f)), and 3) combined time and frequency domains (DeepAuth(f+t)). While, DeepAuth(t) and DeepAuth(f) show promising accuracy, DeepAuth(f+t) has the highest accuracy among all variants. Therefore, the combined time and frequency domain features provide more distinguishable behavioral patterns compared to the time or frequency domain features on their own. DeepAuth(f+t) outperforms DeepAuth(t) which depicts the superiority of the frequency domain compared to the time domain. Although DeepAuth(f) delivers slightly better performance than DeepAuth(t), there is still some useful information in the time domain feature set which may have been missed in the frequency domain representation. Since, the accuracy of DeepAuth(t) is also promising. Moreover, as the size of the segment increases, accuracy improves which implies that behavioral patterns are more detectable within longer segments.

In Figure 6, we show the effectiveness of the features of DeepAuth compared to features in the time and frequency domains in discriminating the samples pertaining to a random (authorized) user and those belong to other (unauthorized) users. We obtain the features of our model by removing the last layer of DeepAuth network and use the output of the immediate previous layer. We visualize the features in a 2-dimensional space by exploiting the t-Distributed Stochastic Neighbor Embedding (t-SNE) [18] which is a dimensionality reduction algorithm. t-SNE is an unsupervised representation learning that maps a given high-dimensional feature vector into a lower-dimensional new space in which the similarity of samples is preserved as much as possible.

As shown in Figure 6, DeepAuth fingerprint features are the most discriminative among all feature spaces, which depicts the effectiveness of our model's features. Moreover, frequency domain features as shown in the figure are slightly better than those of time domain and yet not discriminative enough to distinguish authorized and unauthorized users from each other.

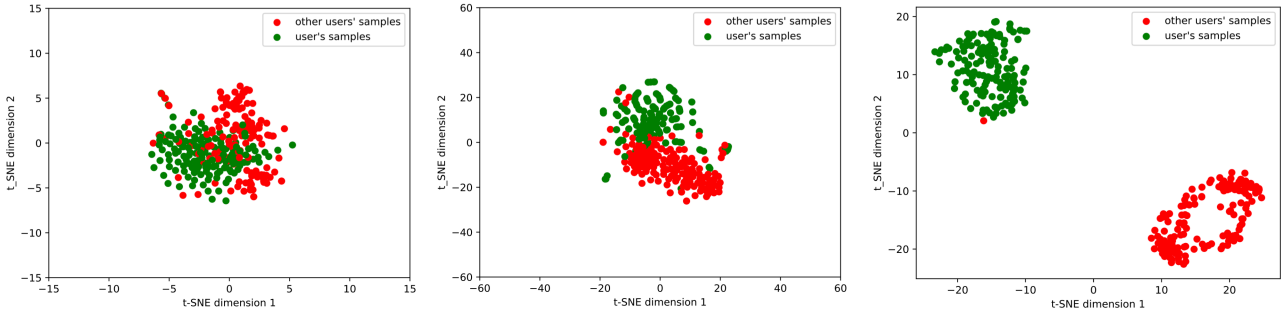
We plotted the subspaces for three randomly selected authorized users to verify the effectiveness of the feature vectors pertaining to the three users in time and frequency domains as well as the DeepAuth feature subspace. As shown in Figure 7, DeepAuth fingerprint subspace is much more discriminative compared to time and frequency subspaces. While these features are slightly discriminative, they are insufficient for providing acceptable overall accuracy.

7 CONCLUSION

DeepAuth provides a deep model in order to re-authenticate mobile app users implicitly and continuously via behavioral patterns while interacting with a mobile phone that are extracted from mobile motion sensors. DeepAuth can re-authenticate a user with 96.70% accuracy within only 20 seconds using only the accelerometer and gyroscope motion sensor data. Even though DeepAuth is trained

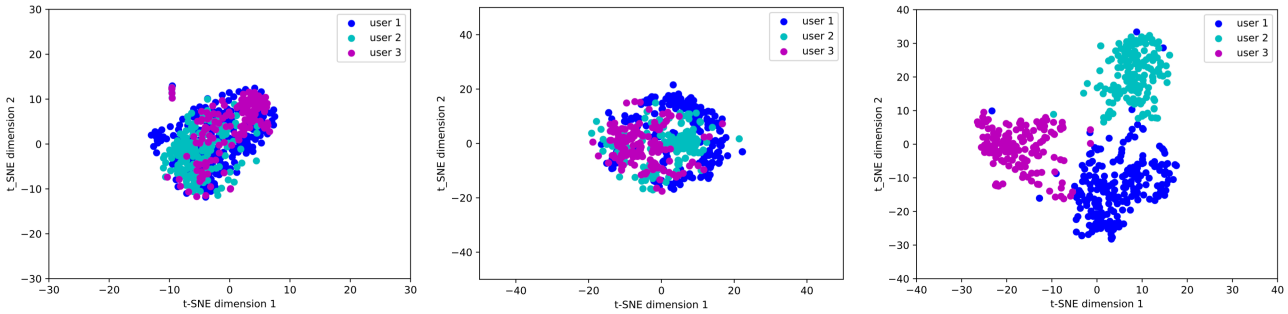
Table 3: Performance of DeepAuth for different feature domains.

Segment Size	500				1000			
Performance	N	P	$F1$	AUC	N	P	$F1$	AUC
DeepAuth(f+t)	**92.73**	**99.20**	**95.85**	**99.05**	**94.37**	**99.16**	**96.70**	**99.19**
DeepAuth(f)	92.83	97.81	95.26	98.88	93.25	98.40	95.76	99.06
DeepAuth(t)	88.84	97.25	92.85	97.52	90.54	98.19	94.21	97.83



(a) time subspace **(b) frequency subspace** **(c) DeepAuth fingerprint subspace**

Figure 6: t-SNE plots for different feature spaces for a randomly selected user.



(a) time subspace **(b) frequency subspace** **(c) DeepAuth fingerprint subspace**

Figure 7: t-SNE plots of different feature spaces for 3 randomly selected users.

on only a 10-minute data streams per user, this performance shows promising accuracy. DeepAuth is deployable in real-world scenarios and can provide a fast, secure and frictionless online shopping experience for users. This approach is also suitable for tasks beyond user authentication, as it can be used for user identification in other contexts as well.

8 ACKNOWLEDGEMENTS

This work would have not been possible without the help of the Data Science team at Target Corporation who took the time and participated in our data collection. We especially would like to thank Ramasubbu Venkatesh and Janet Keel for their support and assistance. We would also like to show our gratitude to Nicholas Eggert, Xiao Pu and Mohammad Ghasemisharif who provided comments that greatly improved the manuscript. We would also like to thank the anonymous reviewers for their extensive and helpful feedback. This material is based upon work supported in part by the National Science Foundation under Grant Nos. CNS-1351058 and CNS-1409868. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Dan Alaimo. 2018. Shopping apps leading target as mobile fraud escalates. https://www.retaildive.com/news/shopping-apps-leading-target-as-mobile-fraud-escalates/520451/. Accessed: 2018-05-02.

[2] Sara Bahaadini, Vahid Noroozi, Neda Rohani, Scott Coughlin, Michael Zevin, and Aggelos K Katsaggelos. 2018. DIRECT: Deep Discriminative Embedding for Clustering of LIGO Data. In *Proceedings of the 25th IEEE International Conference on Image Processing (ICIP)*. IEEE.

[3] Cheng Bo, Lan Zhang, Xiang-Yang Li, Qiuyuan Huang, and Yu Wang. 2013. SilentSense: Silent User Identification via Touch and Movement Behavioral Biometrics. In *Proceedings of the 19th Annual International Conference on Mobile Computing; Networking (MobiCom '13)*. ACM, New York, NY, USA, 187–190. https://doi.org/10.1145/2500423.2504572

[4] Boualem Boashash. 2015. *Time-frequency signal analysis and processing: a comprehensive reference.* Academic Press.

[5] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a" siamese" time delay neural network. In *Advances in Neural Information Processing Systems*. 737–744.

[6] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1724–1734.

[7] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6645–6649.

[8] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[10] CR Investigates. 2013. Keep your phone safe: How to protect yourself from wireless threats. *Consumer Reports* 78 (2013), 6–18.

[11] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*. 2342–2350.

[12] Hilmi Günes Kayacik, Mike Just, Lynne Baillie, David Aspinall, and Nicholas Micallef. 2014. Data Driven Authentication: On the Effectiveness of User Behaviour Modelling with Mobile Device Sensors. *CoRR* abs/1410.7743 (2014). arXiv:1410.7743 http://arxiv.org/abs/1410.7743

[13] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[14] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436.

[15] Wei-Han Lee and Ruby Lee. 2016. Implicit Sensor-based Authentication of Smartphone Users with Smartwatch. In *Proceedings of the Hardware and Architectural Support for Security and Privacy 2016 (HASP 2016)*. ACM, New York, NY, USA, Article 9, 8 pages. https://doi.org/10.1145/2948618.2948627

[16] Wei-Han Lee, Xiaochen Liu, Yilin Shen, Hongxia Jin, and Ruby B Lee. 2017. Secure pick up: Implicit authentication when you start using the smartphone. In *Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies*. ACM, 67–78.

[17] Lingjun Li, Xinxin Zhao, and Guoliang Xue. 2013. Unobservable Reauthentication for Smartphones. In *In NDSS. The Internet Society*.

[18] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[19] Koichiro Niinuma, Unsang Park, and Anil K Jain. 2010. Soft biometric traits for continuous user authentication. *IEEE Transactions on information forensics and security* 5, 4 (2010), 771–780.

[20] Vahid Noroozi, Lei Zheng, Sara Bahaadini, Sihong Xie, and Philip S Yu. 2017. SEVEN: deep semi-supervised verification networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 2571–2577.

[21] Lawrence R Rabiner and Bernard Gold. 1975. Theory and application of digital signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.* (1975).

[22] Elaine Shi, Yuan Niu, Markus Jakobsson, and Richard Chow. 2010. Implicit authentication through learning user behavior. In *International Conference on Information Security*. Springer, 99–113.

[23] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research* 15, 1 (2014), 1929–1958.

[24] Kai Xi, Jiankun Hu, and Fengling Han. 2012. Mobile device access control: an improved correlation based face authentication scheme and its java me application. *Concurrency and Computation: Practice and Experience* 24, 10 (2012), 1066–1085.