Continuous Authentication of Smartphones Based on Application Usage

Upal Mahbub*, Jukka Komulainen†, Denzil Ferreira† and Rama Chellappa‡
*‡Department of Electrical and Computer Engineering and the Center for Automation Research,
UMIACS, University of Maryland, College Park, MD 20742
†University of Oulu, Finland
Email: umahbub@umiacs.umd.edu, yty@iki.fi, denzil.ferreira@oulu.fi, rama@umiacs.umd.edu

Abstract—**An empirical investigation of active/continuous authentication for smartphones is presented by exploiting users' unique application usage data, i.e., distinct patterns of use, modeled by a Markovian process. Specifically, variations of Hidden Markov Models (HMMs) are evaluated for continuous user verification, and challenges due to the sparsity of session-wise data, an explosion of states, and handling unforeseen events in the test data are tackled. Unlike traditional approaches, the proposed formulation utilizes the complete app-usage information to achieve low latency. Through experimentation, empirical assessment of the impact of unforeseen events, i.e., unknown applications and unforeseen observations, on user verification is done via a modified edit-distance algorithm for sequence matching. It is found that for enhanced verification performance, unforeseen events should be considered. For validation, extensive experiments on two distinct datasets, namely, UMDAA-02 and Securacy, are performed. Using the marginally-smoothed HMM a low equal error rate (EER)of 16.16% is reached for the Securacy dataset and the same method is found to be able to detect an intrusion within ~ 2.5 minutes of application use.**

Keywords—**Active authentication; application usage-based verification; unforeseen observation handling; hidden markov models; marginal smoothing; markov chains; sequence matching;**

I. Introduction

With the rapid increase of smartphone users worldwide, the mobile applications are growing both in number and popularity [47]. The number of apps in Google Play store is around 2.1 million, while in Apple App Store, Windows Store and Amazon Appstore there around 2.0 million, 669 thousand, and 450 thousand applications, respectively[1]. It has been estimated that a total of 197 billion mobile applications were downloaded in 2017[2]. A retrospective study in 2017 showed that on average a smartphone user uses over 40 different mobile applications per month and has over 80 different applications installed on the phone [42]. As for usage duration in 2017, in the USA, the smartphone users spend on a daily basis around 2 hours 51 minutes on mobile applications, *i.e.*, over one and a half month usage of applications in a year [3]. With growing concerns of smartphone security, monitoring the application usage coupled with the diverse pool of applications can help to make a difference in user authentication systems.

Smartphone application usage data can provide several interesting insights on the device users leading to different use cases of such data. There are several research works on user profiling and predicting behavioral patterns using application usage data [41], [45], [47], [48]. Predicting application usage pattern can also help optimizing smartphone resources and help simulating realistic usage data for automated smartphone testing [5], [9], [17], [18], [20], [22]. The open foreground application can also work as a context for active authentication using other modalities [8], [19], [29], [31], [40]. For example, when verifying with touch and accelerometer data, the application running in the foreground can provide useful context for robust authentication. Intuitively, the way a user handles and swipes in a phone for a banking application is very different from those for a gaming application. The foreground application context can be even more useful for active authentication if some more insightful information about the applications are available as meta data. For example, one key idea of active/continuous authentication is gradually blocking a probable intruder starting from the most sensitive applications, such as banking and social media accounts [26], [36]. If the sensitivity level or the type of application is known as meta data, it would be possible to attain enhanced security. Also, some applications, if permitted, can access the location data and store click information for targeted advertisement and similar applications [23]. A more active use case of application-usage data could be verifying the users solely from the pattern of usage. The different use cases of app-usage data are shown in Fig. 1.

In this paper, the suitability of application-usage data as a modality for smartphone user verification is thoroughly investigated. The main contributions of this paper are:

- An innovative formulation that utilizes application usage data patterns as a biometric for user verification. The formulation tackles key challenges such as data sparsity and accounting for unforeseen test observations. Unlike traditional approaches of using top N-applications for authentication purposes [12], in the proposed formulation the full list of applications are considered for verification models in order to ensure low-latency which is essential for active authentication systems.

[1] https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/

[2] https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/

[3] https://www.comscore.com/Insights/Blog/Mobile-Matures-as-the-Cross-Platform-Era-Emerges

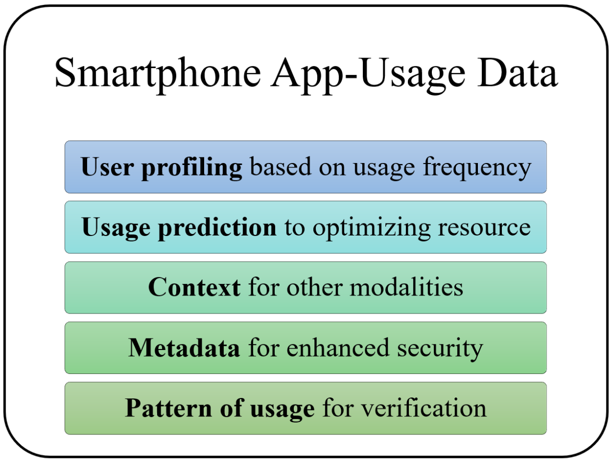

Smartphone App-Usage Data
User profiling based on usage frequency
Usage prediction to optimizing resource
Context for other modalities
Metadata for enhanced security
Pattern of usage for verification

Fig. 1. Use cases for smartphone app-usage data.

- Insight into the application usage similarity among different users and statistics on unforeseen applications.
- A thorough investigation of the impact of unforeseen events, i.e., unknown applications and unforeseen observations, on the verification task. Applications that appear in the test set but were absent in the training set are considered ‘unknown’ to the model for a user. On the other hand, application names combined with certain temporal information are considered as observations in our formulation and observations that were never seen in the training set are considered ‘unforeseen’ to the model.
- A Modified Edit-Distance (M-ED) algorithm and experiments to demonstrate the advantage of including unforeseen events during sequence matching.
- Modeling the Person Authentication using Trace Histories (PATH) problem as a variation of the person authentication using location histories [25].

The paper is organized as follows. In Section II, background and related works on this topic are discussed. In Section III, the approach is explained in detail along with the associated challenges and possible solutions. The impact of unknown application and unforeseen events is investigated in Section IV, and several methods for handling the active authentication problem are described in Section V. Finally, a detailed analysis on the application usage data, experimental results and discussions are presented in Section VI, followed by conclusions and suggestions for future work in Section VII.

II. Related Work

In this section, some of the most recent published literature on active authentication and the utilization of application usage data are reviewed. The section also discusses the exploitation of two active authentication datasets (UMDAA-02 [26] and Securacy [10]) for this research work.

A. Active/Continuous Authentication of Smartphones

Active, continuous or implicit authentication are different terminologies for the same authentication approach in which the rightful user of mobile devices is authenticated throughout the entire session of usage [14], [26], [34]. In recent years, active authentication research has gained a lot of attention because of the increased security risks and complexity of password, token-based, multi-factor and other explicit authentication systems [34]. In active authentication, the wide range of sensor data available on the mobile devices are utilized to learn one or more templates for the legitimate user during a training session. The templates are used in the background to continuously authenticate the user during regular usage and based on the amount of deviation from the templates the device itself starts restricting access to phone applications and utilities starting from the most sensitive ones [26]. Most popular modalities for active authentication are front camera face images [6], [15], [38], touch screen gesture data [7], [11], [49], accelerometer and gyroscope data [13], [32], [35], location data [25] etc. Suitability of different behavioral biometric signatures such as touch and keystroke dynamics, phone pick-up patterns, gait dynamic, and patterns from location trace history have been explored for active authentication [21], [25], [28]. Combinations of multiple biometric have been demonstrated to produce robust authentication on real-life data[4].

B. Prior Research on Application-Usage Data

In recent years, there has been a lot of focus on predicting individual and community-wise application usage patterns [2]. For example, in [47], the authors investigate the ratio of local and global applications in the top usage list, the traffic pattern for different application categories, likelihood of co-occurrence of two different applications and such other patterns in usage. In this work, the authors identify traffic from distinct applications using HTTP signatures. On the other hand, in [43] the authors use mobile in-app advertisements to identify the applications in network traces. Using the ad flow data, the authors analyze the usage behavior of different types of applications. In [48], the authors analyze the application-usage logs of over $4,000$ smartphone users worldwide to develop an app-usage prediction model that leverages user preferences, historical usage patterns, activities and shared aggregate patterns of application behavior.

From the authentication front, in [19], the authors propose an application centric decision approach for active or implicit authentication in which applications are used as context to decide what modalities to use to authenticate a user and when to do it. Application usage data has also been used to generate scores for user authentication in [12]. The authors only consider the frequency of occurrence of an application in the training set to determine the likelihood of being a particular user, missing the temporal variation in the usage pattern.

An interesting use-case of application-usage data is presented in [39]. The authors use a large-scale annotated application-usage dataset to build a predictor that can estimate

[4] http://www.biometricupdate.com/201506/atap-division-head-previews-behavioral-biometrics-system-at-google-io

where a person is (*e.g.*, at home or office) and if he/she is with a close friend or a family member. In [22], the authors use application usage traces along with system status and sensor indicators to predict the battery life of the phones using machine learning techniques.

In Table I, we present a handful of continuous authentication systems and compare them based on some key features. It can be seen from the table that, (a) most of the approaches model the user verification problem as a much simpler one-vs-all classification task, (b) most methods are evaluated on datasets that are not realistic/wild, (c) only a couple of methods other than ours have the capability of leveraging temporal dynamics among consecutive actions, (d) equal error rate (EER) is the most widely used evaluation metric across methods, and, (e) other than our method, only one RNN-based method is capable of handling unforeseen events, while the rest either ignore them or model the problem in an unrealistic way not accounting for unforeseen events.

C. Datasets on Application Usage

Even though there have been diverse research approaches that need application-usage data, there is a scarcity of publicly available datasets. Also, many of the application-usage datasets have limited number of applications or are not unbounded real-life usage data, but instead contain data generated under supervision or by following certain instructions. In this work, all the experiments are performed on two well-known large scale public datasets suitable for investigating the active authentication problem, namely, the application-usage data of University of Maryland Active Authentication Dataset-02 (UMDAA-02)[5] [27] and the Securacy[6] [10] dataset from the Center of Ubiquitous Computing, University of Oulu.

1) UMDAA-02 Application-Usage Dataset: The UMDAA-02 dataset is specifically designed for evaluating active authentication systems in the wild. The dataset consists of 141.14 GB of smartphone sensor data collected from 45 volunteers who were using Nexus 5 phones in their regular daily activities over a period of two months. The data collection application ran completely in the background and the collected data includes the front-facing camera, touchscreen, gyroscope, accelerometer, magnetometer, light sensor, GPS, Bluetooth, WiFi, proximity sensor, temperature sensor and pressure sensor among with the timing of screen unlock and lock events, start and end timestamps of calls and currently running foreground application, etc.

The application usage data from 45 users is summarized in Table II. However, not all the users have adequate amount of usage data. For all the experiments in this paper, a total of 26 users are used who have more than 500 training samples and more than 200 test samples for any sampling rate between $1/5s^{-1}$ to $1/30s^{-1}$. The usage statistics for the top 20 applications for the selected 26 subjects is presented in Table III. The usage rate for the top 20 applications for each user is shown in Fig. 2(a). From the table and the figure, it is readily seen that the applications ranked 6th, 8th, 12th and 20th are in the top list because of excessive usage by very few users, whereas, the remaining applications are genuinely popular among the users.

2) Securacy Application Usage Dataset: The Securacy dataset was originally created within the context of exploring the privacy and security concerns of a smartphone user by analyzing the locations of servers that different applications use and whether secure network connections are used. For a period of approximately six months, the data was collected from 218 anonymous participants who installed the data collection application from the Google Play store. The collected data, 679.90 GB, includes the currently running foreground application, installed, removed or updated applications, application server connections and device location, etc.

Out of the 218 users of the original Securacy dataset, 99 are used for this experiment based on the limits on training and test observations as mentioned for the UMDAA-02 dataset. The application usage data for the 99 subjects in the Securacy dataset are summarized in Table IV and the corresponding usage statistics for the top 20 applications are presented in Table V. The usage rate for the top 20 applications for each user are shown in Fig. 2(b). Note that the top applications ranked 1st, 2nd and 4th in Table V are actually the same application written in Spanish, English and Finnish, respectively. Similarly, rank 12, ‘Horloge’ is ’Clock’ in French, and therefore is the same application as rank 19. However, these applications are shown separately here because, for the active authentication problem, even the preferred language of the user is a type of biometric metadata and can be used to discriminate between users. Also, similar to UMDAA-02 dataset usage statistics, there are several applications in the top 20 rank that were actually used by only a few users very frequently (ranked 1, 4, 9, 12, 16). For this dataset, this phenomenon can be attributed to language difference as well because if the language difference were nullified, then rank 1, 2, 4 will collapse at rank 1 and rank 12 and 19 will collapse at 12 - thereby removing three applications from the list (rank 1, 4 and 12) that has very few users. For the user verification research presented here, the language variation is kept unaltered in order to retain the naturalness of the dataset and the algorithms are expected to learn to discriminate between users based on the language as well as on usage pattern.

III. Problem Formulation

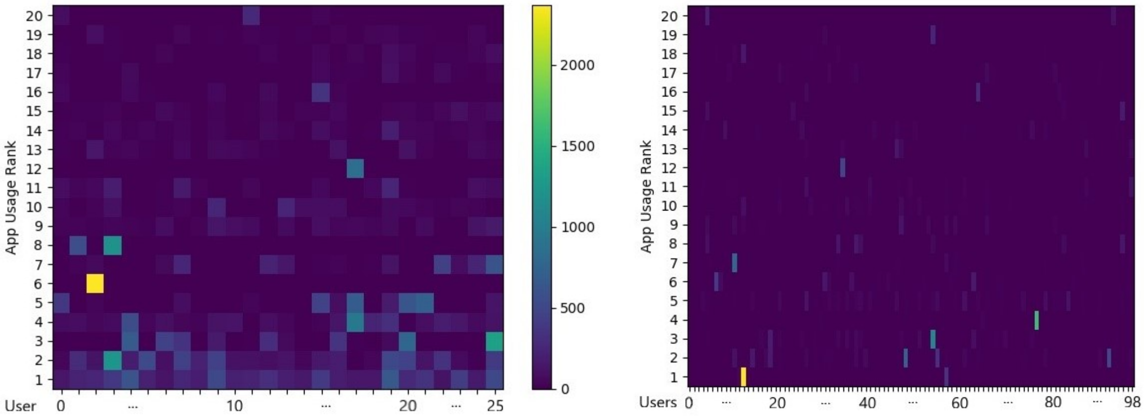
The application usage data from smartphones coupled with the timing information can be used to determine the exact day, time and duration of using any application. It is assumed here that there might be certain patterns in the usage of different applications at different times of the day or during weekdays and weekends. Hence, a state-space model can be intuitively considered for modeling the pattern of application usage for a particular user. Models for different users are assumed to be different because of the differences in lifestyles of each individual. Therefore, the state-space model of a user can

[5] Available at https://umdaa02.github.io/

[6] Available at http://ubicomp.oulu.fi/securacy-understanding-mobile-privacy-and-security-concerns/

TABLE I
COMPARISON AMONG DIFFERENT CONTINUOUS AUTHENTICATION APPROACHES FOR DIFFERENT MODALITIES.

Related work	**Modalities Used**	**Methods and Features**	**Authentication Type**	**Dataset**	**Is the Dataset uncontrolled?**	**Considered Temporal Dynamics among actions?**	**Metric**	**Considered Unforeseen Events?**	**Result**
Gamboa et al. 2004 [16]	Mouse Dynamics	Estimated probability distribution function for each user interaction. Performed multimodal non-parametric estimation and unimodal parametric estimation. 63 Features related to descriptive and higher level statistics of the mouse coordinates.	Verification. Individual model for each user using positive samples only.	50 Volunteer playing several games for 10-15 minutes. \apporx10 hours of interaction, corresponding to ¿400 strokes/user	No	No. Each strokes is considered an independent action. When multiple strokes are considered for evaluation, they are not temporally sorted.	Mean EER	N/A. Feature vector consists of real numbers.	Mean EER of 2% for 50 mouse strokes.
Zheng et al. 2011 [51]	Mouse Dynamics	Three fine grained metrics - direction, angle of curvature and curvature distance in addition to speed and pause-and-click measures. Classification using SVM (RBF kernel).	One-vs-rest Classification	Dataset1: Total of 81,218 point-and-click actions (average 5,801 actions/user, ≈150 hours of raw mouse data). Dataset2: 1,074 users interacting in an online forum for an hour.	No	No. Temporal relation between sessions are not considered.	FAR, FRR, EER	N/A. Feature vector consists of real numbers.	Mean EER of 1.3% for 20 clicks.
Sae-Bae et al. 2012 [37]	Touch gesture on multi-touch devices	Gesture denoted as n consecutive touch sequences (sequence of x, y coordinates of the touches). Used Dynamic Time Warping (DTW) to compute the sum of Euclidean distance between touch sequences of a test gesture and a template..	Verification	34 participant (24 male, 10 female) using an IPAD application producing 22 gestures.	No	No. Temporal relation between touch sequences not considered in gesture.	.EER	Not considered. New gestures would always produce high Euclidean distance irrespective of the user.	10% EER on single gesture and 5% EER on double gesture.
Frank et al. 2012 [11]	Touch gesture on a smartphone	30 statistical features including mid stroke area, stroke direction, velocity and duration, phone and finger orientation etc. kNN and SVM (RBF kernel) classifiers are trained.	One-vs-rest classification	Touchalytic Dataset: 41 users (64% M and 84% right handed) using two custom software for 25-50 mins/user.	No	No. Temporal relation between consecutive strokes are not explored/utilized.	FAR, FRR, EER	Not considered.	2% to 3% inter-session Mean EER and 0%-4% inter-week mean EER.
Luca et al. 2012 [4]	Touch-screen Patterns drawn on smart-phones	Time series of touch screen data (X and Y coordinates), pressure, size, time). Dynamic Time Warping (DTW) to match sequences.	One-vs-rest Classification for one-time-authentication.	48 user (22F, 26M) performing 4 different types of patterns to unlock a smartphone a total of 160 times in two different days Test set has 640 Unlocks per user..	No	Yes, when comparing two sequences DTW considers temporal relation. However, temporal relation between two consecutive patterns are ignored.	Unlock accuracy	N/A. Not a continuous authentication problem.	37%-57% two-day accuracy across different settings.
Monrose et al. 2000 [30]	Keystroke dynamics data	Users are clustered into groups based on similarity in typing rhythm for different letter combinations comprising of (possibly) disjoint feature sets in which the features in each set are pair-wise correlated. Scored based on Euclidean distance, non-weighted probability, weighted probability and Bayesian Classifier.	Classification	Data from 63 users collected over 11 month.Data consists of structured texts. Users had control on when to run the experiments.	No. Experiment was supervised. Only initiation was in the wild.	No, there is no-mechanism to consider the temporal relation between the keystrokes.	Identification accuracy	Not considered. Unforeseen events are ignored in the formulation.	85.63% accuracy for non-weighted probability measure, 92.14% accuracy for Bayesian Classifier.
Fathy et al. 2015 [6]	Face images obtained from the front camera of a smartphone	400D MEEN features obtained from the region surrounding the mouth, eyes and nose. Best performing classifiers are Fisherfaces (FF) [1], Sparse Representation-based Classification (SRC) [46] and Mean-Sequence SRC (MSSRC) [33].	50 class Classification	UMDAA-01 Dataset: A dataset of 50 users (43M, 7F) using smartphones for 5 different tasks and 3 different illumination in portrait mode. A total of 750 front camera videos and 600 txt files recording screen touch data.	No, The data collection was done using specially designed apps for certain tasks.	No, evaluation was done based on individual images without considering temporal aspects. Also, the dataset only contains images for a single day per user. .	Classification accuracy	Unforeseen pose and occlusion might lead to failure in face detection.In that case, the classifier will ignore the frame.	Accuracy range between 54.7% and 74.9% for FF, 24.2% and 73.9% for SRC, and 22.1% and 72.2% for MSSRC (trained on any two sessions and tested on the third).
Zhang et al. 2015 [50]	Touch Dynamics data	27 Features similar to [11]. Used Kernel Dictionary-based Touch-Gesture Recognition (KDTGR) method.	Classification	UMDAA-01 Touch Dynamics Dataset	No	No, randomly selects swipes, ignoring temporal or logical dependencies.	EER, F1-Score	Not Considered.	Average EER of 2.62% 0.65 when trained and tested on all sessions.
Lourenco et al. 2012 [24]	ECG signal obtained non-intrusively from hand palms	Real-time denoising of the ECG waveform followed by an on-line R-peak detection and kNN or SVM classifiers.	Authentication with kNN is in verification setting. With SVM it is One-vs-rest Classification.	Data collected from 32 subjects (25M, 7F, avg. age 31.19.46 years) using a non-intrusive hand-palm ECV device within a 5 minutes period/user.	Data collection setting was supervised.	Not considered.	EER and mean identification error (Eid)	N/A	EER of 2.75%0.29 when average of 5 heartbeats are considered for NN method.
Fridman et al 2017 [12]	Stylometry (text analysis), app usage, web browsing, device location.	Trained binary classifiers for each modality and performed global decision fusion.	One-vs-rest Classification	200 subjects using their personal Android mobile device for a period of at least 30 days.	Uncontrolled	Not considered. Used an one-feature n-gram classifier for text data, prior based decision rule for app and web browsing data, SVM trained on latitude and longitude.	FRR, FAR, Mean EER	Explicitly ignored.	FAR and FRR of 30% and 18%, respectively for the app-usage data. Overall mean EER of 5% after 1 minute of user interaction with the device, and 1% after 30 minutes.
Nevarova et al. 2016 [32]	Accelerometer and gyroscope data from smart-phones	Conv-DCWRNN method: Temporal feature extraction by a modified Clockwork recurrent network followed by classification via a probabilistic generative model.	Verification. But, requires lot of additional user data to train the Universal background network.	Googles project abacus dataset consisting of unconstrained smartphone usage data from 1500 users for an average of 3 months.	Uncontrolled	Yes.	MeanEER, Half total error rate (HTER) in %	Yes, the UBM network should learn more general information.	8.82 % mean EER per session, 15.84% mean EER per device and 18.17% mean EER per user.
Ours	**Application usage data**	**Converted application name and timing information into observations that are suitable for state-space models as well as string matching techniques.**	**Verification. Models per user trained only on the positive samples.**	**UMDAA-02 dataset and Securacy Dataset**	**Uncontrolled**	**Yes**	**Mean EER**	**Yes**	**Mean EEE ≈30% (UMDAA-02) and ≈16% (Securacy) (50 historic observations sampled at $1/30s≙1$).**



(a)

(b)

Fig. 2. Similarity matrix depicting top 20 application-usage rate among users in the training set of the (a) UMDAA-02 dataset and, (b) Securacy dataset. Image is best viewed digitally in high resolution.

TABLE II
GENERAL INFORMATION ON APPLICATION-USAGE DATA AVAILABLE IN THE UMDAA-02 DATASET.

No. of Subjects with $\geq$ 500 training samples and $\geq$ 200 test samples for sampling rate of $1/30s^{-1}$ (Train/Test)	32/26
Avg. No. of Sessions/User with App-Usage Data of the 26 selected subjects (train/test)	$\sim 582/\sim 197$
Train/Test split for the experiment	70%/30%
Total Number of Unique Applications Used by the 26 selected subjects (train/test)	119/67
Average Number of Samples Per User for the 26 selected subjects (train/test)	$\sim 4307/\sim 1399$

TABLE IV
GENERAL INFORMATION ON APPLICATION-USAGE DATA AVAILABLE IN THE SECURACY DATASET.

No. of Subjects with $\geq$ 500 training samples and $\geq$ 200 test samples for sampling rate of $1/30s^{-1}$ (Train/Test)	201/99
Avg. No. of Sessions/User with App-Usage Data of the 26 selected subjects (train/test)	$\sim 119/\sim 96$
Train/Test split for the experiment	70%/30%
Total Number of Unique Applications Used by the 26 selected subjects (train/test)	1340/554
Average Number of Samples Per User for the 26 selected subjects (train/test)	$\sim 2235/\sim 1745$

TABLE III
APP-USAGE STATISTICS FOR THE TOP 20 APPS FOR THE 26 SELECTED USERS OF THE UMDAA-02 DATASET.

Rank	App Name	No. of Users	Per User Usage	Overall Usage
1	com.google.android.googlequicksearchbox	26	283.27	283.27
2	com.android.dialer	25	255.24	245.42
3	com.whatsapp	15	303.6	175.15
4	com.android.chrome	26	141.42	141.42
5	com.facebook.katana	11	308.18	130.38
6	com.nextwave.wcc2	1	2366	91
7	com.google.android.youtube	16	144.38	88.85
8	com.ea.game.pvzfree	2	872.5	67.12
9	com.google.android.gm	24	51.04	47.12
10	com.android.mms	22	52.09	44.08
11	com.google.android.talk	18	62.28	43.12
12	com.andrewshu.android.reddit	1	842	32.38
13	com.nextbus.mobile	19	41.89	30.62
14	com.google.android.apps.docs	24	33	30.46
15	com.android.settings	24	27.71	25.58
16	com.google.android.apps.maps	14	44	23.69
17	com.android.camera2	22	20.5	17.35
18	com.google.android.gallery3d	17	24.94	16.31
19	com.android.vending	21	20.1	16.23
20	com.viber.voip	5	74.6	14.35

TABLE V
APP-USAGE STATISTICS FOR THE TOP 20 APPS FOR THE 99 SELECTED USERS OF THE SECURACY DATASET.

Rank	App Name	No. of Users	Per User Usage	Overall Usage
1	Sistema Android	4	9972.25	402.92
2	Android System	80	480.44	388.23
3	com.android.keyguard	34	802.79	275.71
4	Android-jrjestelm	5	4820.8	243.47
5	System UI	80	242	195.56
6	Nova Launcher	19	794.79	152.54
7	Maps	38	363.08	139.36
8	Google Search	53	214.3	114.73
9	Launcher	12	650	78.79
10	Chrome	60	128.2	77.7
11	Facebook	49	154.53	76.48
12	Horloge	1	7328	74.02
13	YouTube	49	144.94	71.74
14	TouchWiz home	20	348.3	70.36
15	Securacy	84	75.39	63.97
16	Internet	16	371.25	60
17	WhatsApp	37	154.62	57.79
18	Google Play Store	72	71.83	52.24
19	Clock	44	113.89	50.62
20	Package installer	36	138.69	50.43

effectively be considered as a model for the pattern of life of that user and can be used to differentiate the user from others. There are however several challenges to this approach towards solving the authentication problem using application usage:

- Forming observation states from the application data and corresponding timing information.
- Training a state-space model in a way that it can handle unforeseen observations during testing.
- Generating verification scores from sequential observation data.

Each of these challenges and the proposed solutions are discussed here.

A. Application Names to Observation States

Incorporating the temporal information with the application name is a challenge because the user can use an application at any time, and therefore the power set of all applications and all probable time is intractable even if we sample at a relatively high frequency. For example, if there are N number of applications and if we sample every 5 minutes, then there would be 480 unique time stamps in a day and 3360 timestamps in a week. This would mean a total of $3360 \times N$ observation states for the applications in a week. However, for a single application, most of these observation states will either not occur or occur very infrequently in the training set. Hence, training a reliable state-space models with this sparsely occurring observation states will be difficult.

In this regard, the time-zone and weekday/weekend flag idea are adopted from [25]. By dividing the day into three distinct time zones (TZs), namely, TZ_1 (12:01 am to 8:00 am), TZ_2 (8:01 am to 4:00 pm) and TZ_3 (4:01 pm to 12:00 pm), and denoting weekday/weekend with a flag $W(t) \in W_D, W_E \forall t$, respectively, the total number of possible observation states is kept limited to $6N$. The functions $TZ(t)$ and $W(t)$ map any time t into one of the corresponding timezone and weekday/weekend, respectively. The impact of converting application tags into observations on verifying the users of the UMDAA-02 app-usage data and the Securacy datasets can be visualized from Figs. 3(a)-(b) and 3(c)-(d), respectively. The similarity matrix in Figs. 3(a) depicts the percentage of common applications between two users in UMDAA-02 training dataset, whereas, the similarity matrix in Fig. 3(b) depicts the percentage of common observations between any two users on the same dataset. It is clear that the similarity of observations between two different users is less than the similarity of applications. The effect is less visible on the Securacy dataset (Figs. 3(c)-(d)) because the subjects came from a diverse population than the subjects of the UMDAA-02 dataset. Hence, the similarity of applications is less pronounced, yet, the differences between application similarity and observation similarity are still present.

B. Taking Unknown Applications into Account

Now, in order to handle unknown applications that might be present in the test set, an additional application name U is considered. The U application adds 6 observation states when combined with TZs and W. Note that in the training set there is no probability of having any U application, and all the observations with U are assigned a very small prior probability $(10e - 20)$ when state-space models are trained. Also, it is ensured for state-space models that the emission probability for the states with U application does not go to zero, in order to prevent zero probability score during testing when unknown applications are encountered. If the total number of unique applications used by user X in the training set is A_x, then any application α_y of the test user Y in the test set $\bar{A}_y$ will be denoted as U if $\alpha_y \notin A_x$. In [25], the authors addressed similar issues for geo-location data by considering even more additional states such as nearby unknowns. However, proximity is a vague concept for application data and therefore only U is considered here. Note that, any observation with an unknown application is unforeseen by default, but an unforeseen observation with some other application name is not unknown.

Note that, apart from U, unforeseen observations might be present in the test set. For example, in the training set an application α_x might only occur in weekdays at timezones TZ_1 and TZ_2 while the same application might be used in the test set at time zone TZ_3 on a weekday. In that case, the test observation (α_x, TZ_3, W_D) would be unforeseen in the training set. For state space models, this problem is handled by generating all possible combinations of applications, time zone and day flag and use them to construct the model. If one such observation is not present in the training set, it is assigned non-zero prior and emission probabilities to ensure that they do not bring down the probability of a test sequence to zero.

C. Handling Uncertainty

Now that unknown applications and unforeseen observation states are addressed, we tackle the creation of observation states via binning of time-stamped data. In most cases, the data collection is done in sessions, where a session starts with unlocking the phone and stops when the phone is locked again. Even if this is not the case, there can be very long idle times between consecutive usage of a phone, during which, authentication is a redundant operation and no application is running in the foreground [44]. Hence, there can be a big gap between the start-time for an application and the stop time of the previous application in the data log. This time gap might be as short as several seconds or as long as several days even for a user who owns a smartphone for regular use [3]. The sparsity introduced by this time gap is handled in two ways. At the beginning of each session (unlocking of the phone) a dummy observation state Ψ is introduced. The state-space model is expected to learn that Ψ is a time gap which might or might not cause a change in the time zone. For example, the last used application might be in TZ_1 before the closing of a session. Then the next session may occur in either TZ_1 or TZ_2 or TZ_3 of the same day. If the next session is in the next day or if the day changes within a running session, then

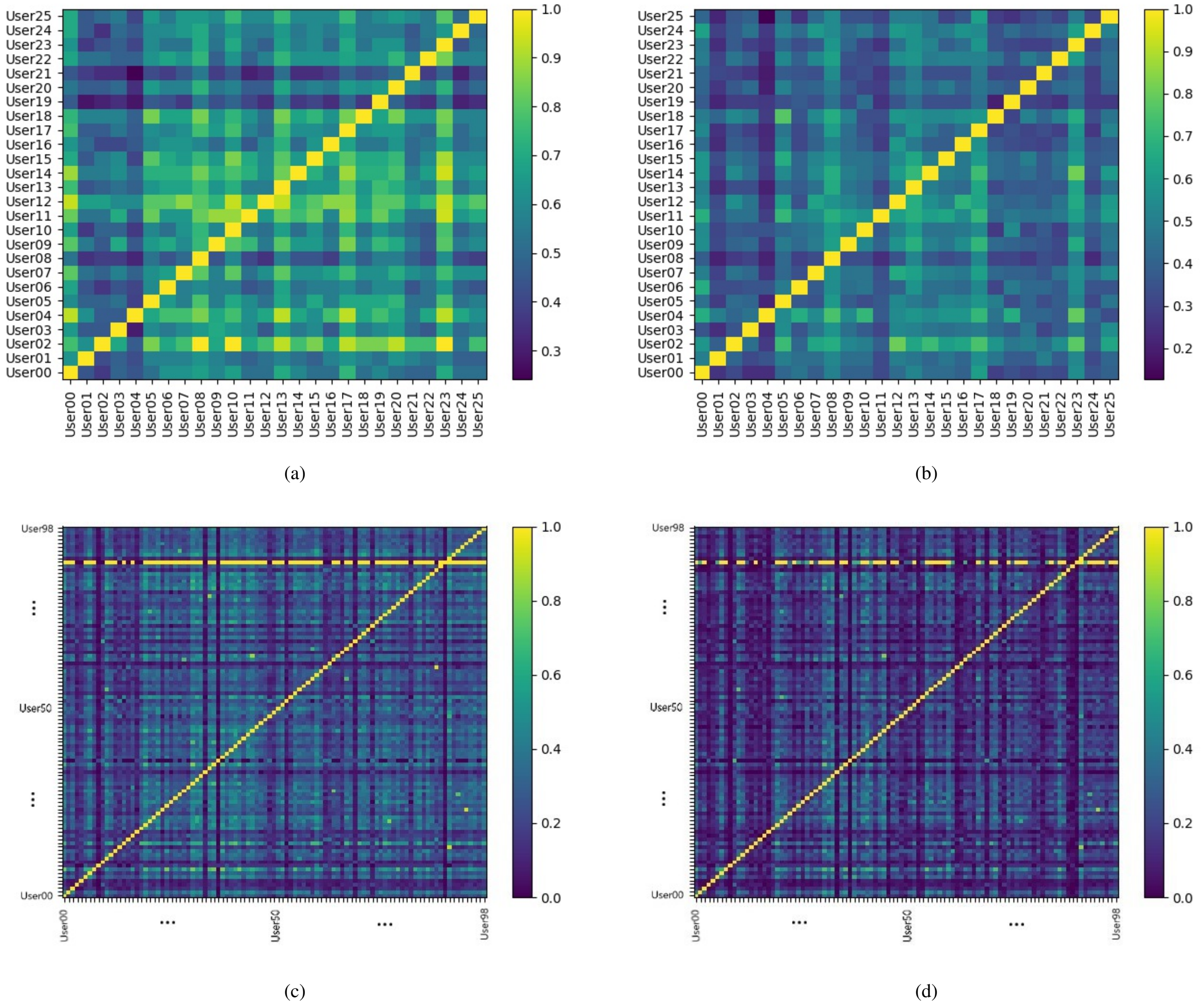


Fig. 3. Similarity matrix depicting (a) application name overlap, and (b) observations overlap for the training set of the UMDAA-02 dataset. Similarly, (c) and (d) depict the application overlap and observations overlap for the training set of the Securacy dataset. Image is best viewed digitally in high resolution.

an additional flag Δ is introduced which denotes the transition into next day. The time zone and weekday/weekend flags are ignored for observations Ψ and Δ.

So, taking the six probable observations for U and the Ψ and Δ observations into consideration, the total number of possible observation states for user X would be $6N + 6U + \Psi + \Delta$.

D. System Overview

A diagram depicting an application-usage-based user verification system is shown in Fig. 4. Once the observation sequence is extracted, a verification model can be trained based on the patterns in the sequence. The verification model can be a state space model, a string matching approach or even a recurrent neural network, depending on data availability and need. For state-space models, once training for a user is done, the model can be used to generate scores for last n test observation sequences created using the same protocol that was used during the training phase. The score can be thresholded to obtain the verification decision. For simpler methods such as sequence matching, unknown applications and unforeseen observations are difficult to handle. For the authentication problem, the unknown and unforeseen play key roles, described in the next section.

IV. The Role of Unknown Application and Unforeseen Observations in User Verification

In this section, we investigate the impacts of unknown applications and unforeseen observations on the verification task. We first look into the prevalence of unknown applications and try to get an intuitive idea about their trend. Then, we take the formal approach of designing three simple separate verification experiments to evaluate performances with and without unforeseen events. The outcome of these experiments

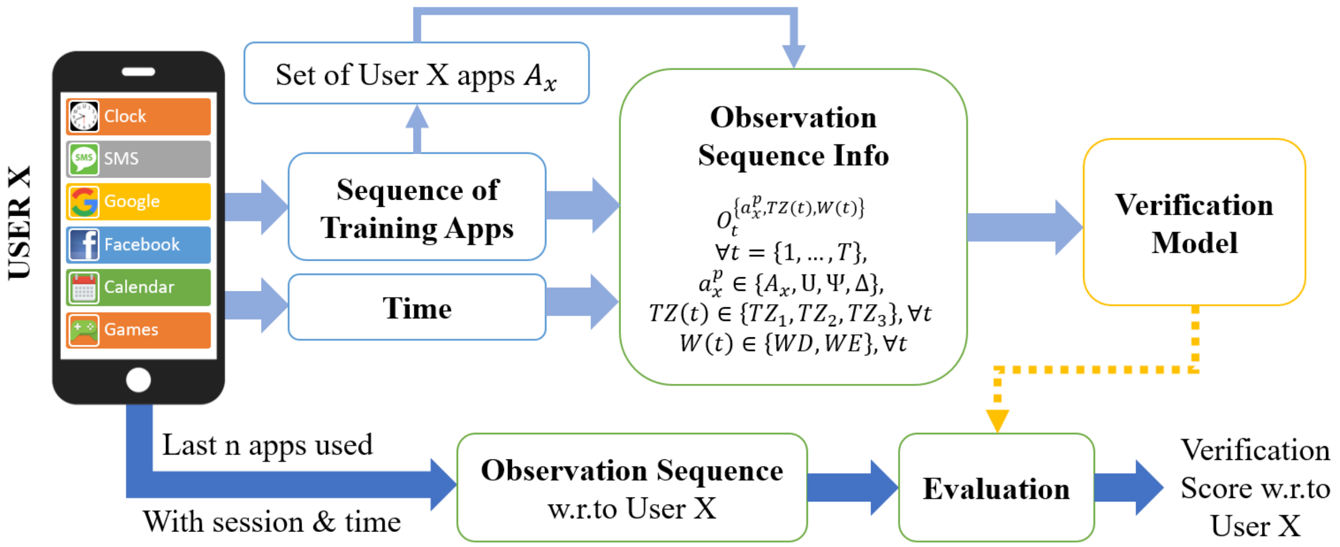

USER X
Clock
SMS
Google
Facebook
Calendar
Games
Set of User X apps A_x
Sequence of Training Apps
Time
Observation Sequence Info
$O_t^{\{a_x^p, TZ(t), W(t)\}}$
$\forall t = \{1, \ldots, T\}$,
$a_x^p \in \{A_x, U, \Psi, \Delta\}$,
$TZ(t) \in \{TZ_1, TZ_2, TZ_3\}, \forall t$
$W(t) \in \{WD, WE\}, \forall t$
Verification Model
Last n apps used
With session & time
Observation Sequence w.r.to User X
Evaluation
Verification Score w.r.to User X

Fig. 4. Overview of an application-usage-based user verification system for mobile devices.

will clearly show the extent of influence that unforeseen observations might have on verification tasks.

A. Statistics of unknown applications in the test data

If an application is present in the test set but not encountered in the training set, the application is denoted with U as the unknown application in the proposed formulation. Intuitively, the prevalence of U will be much higher if the test set comes from a different user or from an intruder of the phone, while for the legitimate user the test set will have fewer unknown applications. This intuition is verified on the application usage data from both UMDAA-02 and Securacy datasets, as can be seen from the box plots in Fig. 5.

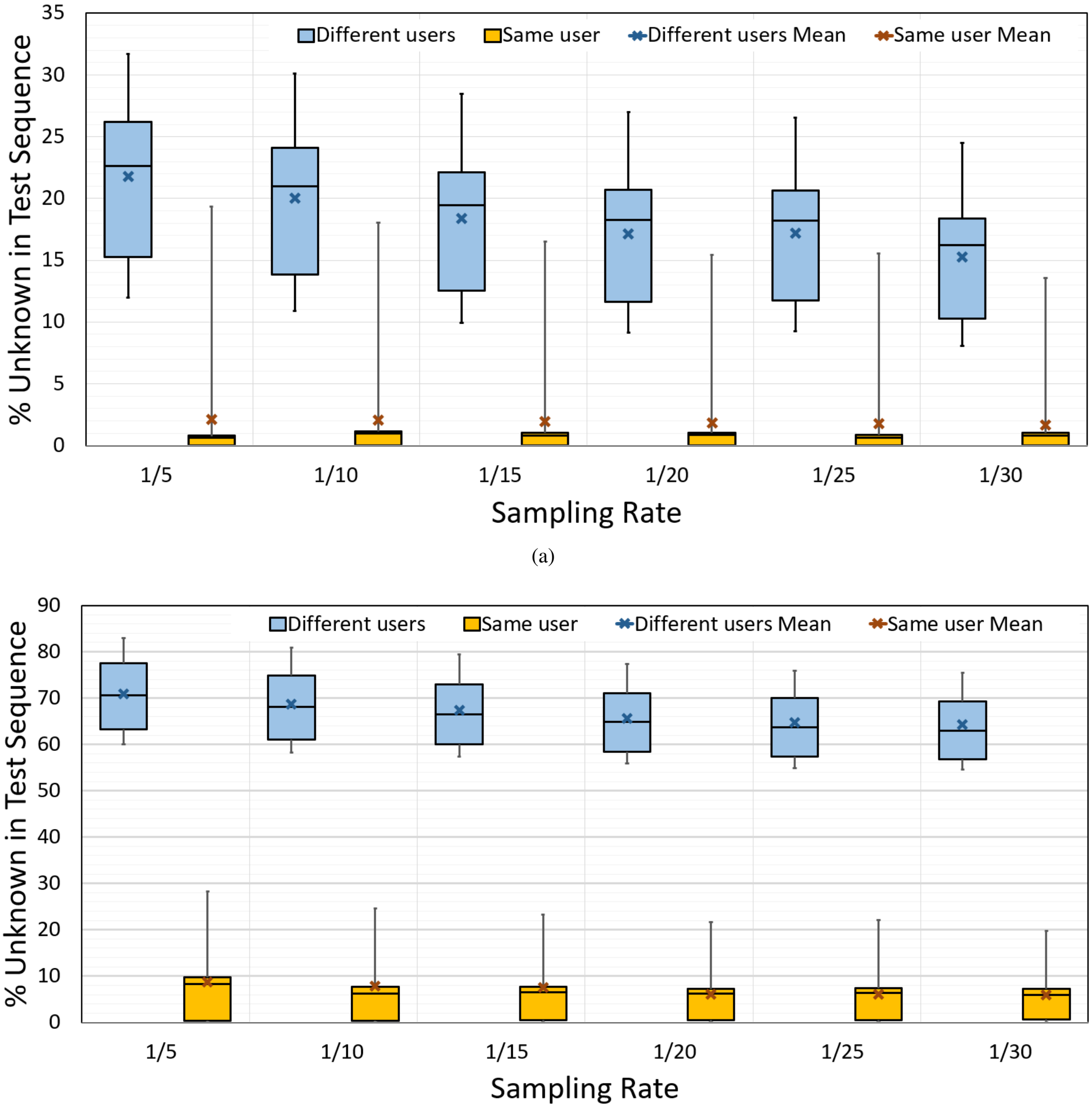
Note that the gap between the whisker plots for same user and different users is larger for the Securacy dataset in comparison to UMDAA-02 dataset. Securacy is a larger dataset with more users, more data per user and more variation in user demographies compared to UMDAA-02 in which the subjects were from a narrow age range and were all affiliated with the same institution. Hence, it shows that among the general population, even the selection of applications varies widely between users.

B. Impacts of unforeseen events on binary decision performance

Two simple experiments with unknown applications U and unforeseen observations are performed on the UMDAA-02 and Securacy datasets to evaluate their role in user verification. The observations for each user are chronologically sorted and the earliest 70% observations are considered for training and the rest for testing. Now, for any user i in the training set, a sequence of training observations S_i^{tr} is obtained along with the set of unique applications A_i. Now, each test sequence of a user is compared with the training sequence and application lists of the training subjects and different binary hard decision rules are applied in two experiments. In the first experiment, the binary decision rule is based on occurrence of an application in the test set that is not present in the training set. In the second experiment, the decision is taken based on the occurrence of an unforeseen observation in the test set. In both cases, if there is even a single occurrence of an unknown application or an unforeseen observation, then the match score is set to 0.0, otherwise it is set to 1.0. The matching algorithms for the two experiments are shown in (1) and (2), respectively. The data sampling rates for both these experiments were set to $1/30$ per second, which resulted in $\sim$ 16863 training-test sequence pairs for the UMDAA-02 application-usage dataset and $\sim$ 846331 training-test sequence pairs for the Securacy dataset. The number of users with adequate training and test data is 26 in UMDAA-02 and 99 in Securacy, leading to an average of $\sim$ 647 and $\sim$ 8549 pairs per user, respectively.

Algorithm 1 Binary Decision Rule based on Unknown Applications

```
procedure BinUnk($A_i$, $S_j^{te}$) ▷ List of unique applications of user $i$ ($A_i$), n-last Test Sequence Vector of user $j$ ($S_j^{te}$)
    for $v^{te} \in S_j^{te}$ do    ▷ Loop through all test observations
        $a^{te} \leftarrow v^{te}[0]$    ▷ Get the application name from the test observation
        if $a^{te} \notin A_i$ then
            return 0.0    ▷ Return score 0.0 if any unknown application is encountered
        end if
    end for
    return 1.0    ▷ Return score 1.0 if no unknown application in test sequence
end procedure
```



(a)

(b)

Fig. 5. Boxplots depicting the percentage of unknown application in test data for (a) UMDAA-02 dataset, and (b) Securacy dataset, for different sampling rates. Note that the average percentage of unknown applications used by the the different user is much larger than that for same user on both datasets.

Algorithm 2 Binary Decision Rule based on Unforeseen Observations

```
procedure BinUnfore($S_i^{tr}$, $S_j^{te}$)    ▷ Sequence of training observations for user $i$ ($S_i^{tr}$), n-last Test Sequence Vector of user $j$ ($S_j^{te}$)
    for $v^{te} \in S_j^{te}$ do    ▷ Loop through all test observations
        if $v^{te} \notin S_i^{tr}$ then
            return 0.0 ▷ Return score 0.0 if any unforeseen observation is encountered
        end if
    end for
    return 1.0    ▷ Return score 1.0 if no unforeseen observation in test sequence
end procedure
```

Results for several evaluation metrics namely, sensitivity, specificity, F1-score and accuracy - all in percentage, obtained through the two experiments on the two datasets are shown in Fig. 6(a)-(d). The definition of these metrics are as follows:

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \quad (1)$$

$$Specificity = \frac{TN}{TN + FP} \times 100\% \quad (2)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \quad (3)$$

$$F1 - Score = \frac{2TP}{2TP + FP + FN} \times 100\% \quad (4)$$

where, TP, FP and FN are the numbers of true positive, false positive and false negative detections, respectively. High Sensitivity implies smaller number of false-negatives, while

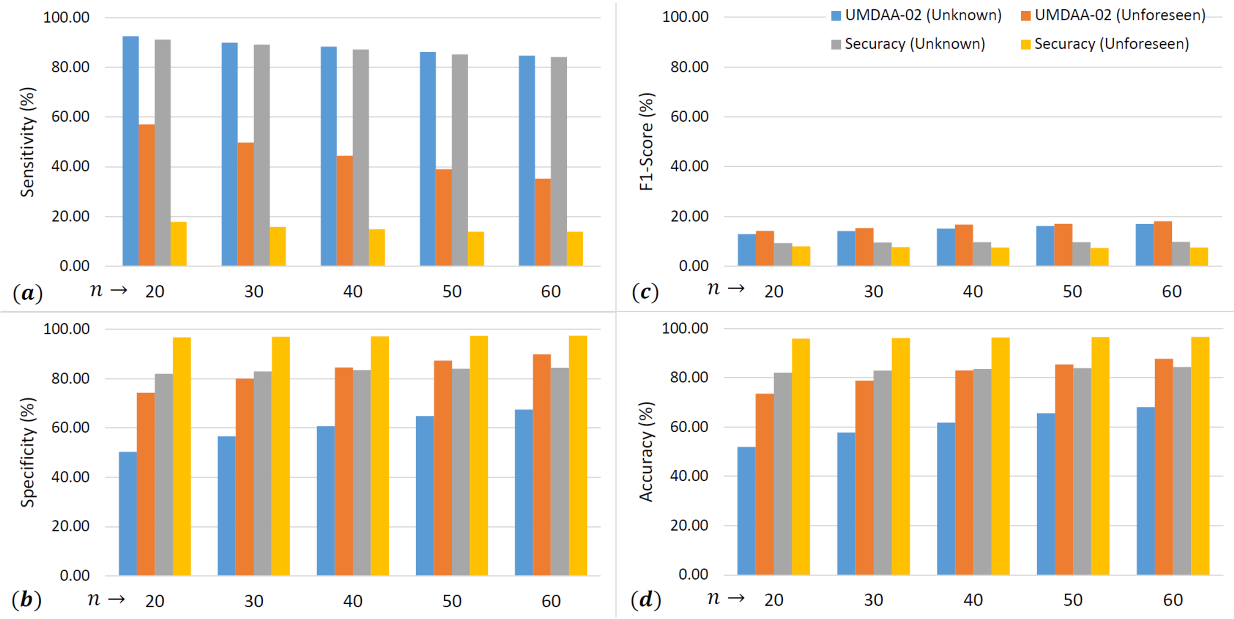


Fig. 6. (a) Sensitivity, (b) Specificity, (c) F1-Score, and (d) Accuracy (in %) obtained by varying sequence length n for Securacy and UMDAA-02 application-usage data for using the Binary Hard Decision rule based on unknown applications and unforeseen observations.

high Specificity implies less false-positives. Accuracy over 50% denotes that the true values outweighs the false predictions. Finally, F1-Score implies better overall precision and recall.

Fig. 6 gives the following interesting insights about the impact of the unknown applications and unforeseen observations on the performance metrics for the two datasets.

- With increasing sequence length n, the specificity is increasing gradually for all the cases, while sensitivity is decreasing. The decrease in sensitivity is probably due to the fact that the probability of having an unknown application in the sequence increases with increasing sequence size, thereby increasing the number of false negatives. On the other hand, with increasing n more sequences are denoted as negatives, which in effect reduces the number of false positives and therefore increases the specificity.
- The sensitivity drops drastically when unforeseen observations are used instead of unknown applications as decision criteria. This is understandable, since the number of false negatives increases rapidly when any sequence with at least one unforeseen observation is marked as data from a different user.
- The number of false positives decreases when unforeseen observations are considered for decision instead of unknowns. This leads to a jump in specificity for a fixed n. In general, the specificity is much higher for Securacy dataset in comparison to UMDAA-02. This proves that there are more unknown applications and unforeseen applications in Securacy when comparing a user with others. Securacy being a more diverse and larger dataset has wider variation of information, which leads to this phenomenon. Here, the training data for each user is longer, meaning that they are much closer representation of real life and therefore, an unknown application or unforeseen observation is actually a different user's data in most cases.
- Higher sensitivity, however, does not mean that for real life data a simple binary classifier based on unforeseen observations is reasonably good. The F1-Score is very low for both datasets, which means either precision or recall or both of therm are very low. Since in the active authentication using application usage, the number of positive pairs is largely outweighed by the number of negative pairs, it can be assumed that $FP >> FN$ and $TN >> TP$. Since Precision=$\frac{TP}{TP+FP}$ and Recall=$\frac{TP}{TP+FN}$, that means, Recall>Precision. With increasing n, FN increases, while FP decreases, leading to reduction in recall and increase in precision. However, given the fact that the F1-Score does not improve much with increasing n, it can be assumed that Recall reduces steeply while Precision does not improve much.
- Irrespective of deciding with unknown or unforeseen, the accuracy is always lower for the UMDAA-02 dataset in comparison to Securacy dataset. Even though the application-usage information in Securacy is much larger than UMDAA-02, probably due to the high demographical similarity among the subjects of UMDAA-02, the binary hard measure performs poorly in comparison to Securacy. In practice, there could not be any assumption made about the demographic similarity or dissimilarity of an user and an intruder - hence, using neither unknown applications nor unforeseen observations as a hard decision metric cannot be a practical solution to the active authentication problem.

- The experiment once again proves that 'accuracy' is not a good performance metric when the number of samples between classes is severely biased. In this example, the average percentage of positive pairs in the dataset is $\sim 3.85\%$ on UMDAA-02 dataset and $\sim 1.01\%$ in the Securacy dataset. Being an open set problem, the task is to deal with heavily biased data towards negative samples and better performance measures in this regard would be receiver operating characteristic (ROC) curves and equal error rates (EER) instead of accuracy.

C. Impacts of ignoring unforeseen events

Now that the impact of unforeseen events on the authentication problem is established, a slightly more advanced sequence matching approach based on Levenshtein Distance a.k.a Edit-Distance (ED) is performed to study the impact of ignoring the unknown observations and unforeseen events. When matching sequence s_1 to another sequence s_2 of the same length, the original ED calculates the number of deletions, insertions, or substitutions required to transform s_1 to s_2. For the active authentication problem, let's assume that a test observation sequence S^{te} of length n is to be matched with any training observation sequence S^{tr} of length N, where, intuitively $N > n$. Since each observation consists of an application name, timezone and day flag, when a mismatch occurs, the distance can be assumed to be different depending on the amount of match. For example, if only the application name matches, then the timezone and day flag needs to be substituted, leading to two operations. Mathematically, the modified edit distance between S^{tr} and S^{te} can be expressed recursively as

$$\mathrm{ED}(i,j) = \min \begin{cases} \mathrm{ED}(i-1,j)+1 \\ \mathrm{ED}(i,j-1)+1 \\ \mathrm{ED}(i-1,j-1)+ \\ \qquad\qquad \delta(S^{tr}(i-1), S^{te}(j-1)), \end{cases} \tag{5}$$

where, $\mathrm{ED}(i,0) = i$ for $i = 0, 1, \dots, N$, $\mathrm{ED}(0,j) = j$ for $j = 0, 1, \dots, n$, and

$$\delta(a,b) = \begin{cases} 0 & \text{if } a^{[A]} = b^{[A]},\, a^{[T]} = b^{[T]},\, a^{[W]} = b^{[W]}) \\ 1 & \text{if } a^{[A]} = b^{[A]},\, (a^{[T]} = b^{[T]} \text{ or } a^{[W]} = b^{[W]}) \\ 2 & \text{if } a^{[A]} = b^{[A]},\, a^{[T]} \neq b^{[T]},\, a^{[W]} \neq b^{[W]}) \\ 3 & \text{otherwise.} \end{cases} \tag{6}$$

Here, a and b denote observations which consists of application name A, time zone T and day flag W. The penalty term δ is 3 when there is no match at all between a and b, 2 when only the app names matches, 1 when either time zone or day flag matches in addition to matching app names and 0 when there is no mismatch among the A, T and Ws of a and b. Based on these equations, a modified algorithm for edit distance (M-ED) is presented in (3) that calculates the distance using an iterative dynamic programming approach to directly calculate the final distance without formulating the entire transition matrix.

Using this algorithm, three different tests are performed on the UMDAA-02 dataset, the results for which are given in Table VI. In the first test, all test observations are included, while in the next two tests, the observations with unknown applications, and the unforeseen observations are ignored. In order to ignore the unforeseen observations, for any training sequence, each test sequence is compared to find the unforeseen observations and removed from the test sequence. For unknown applications, the corresponding observation is removed. This operation reduced the number of samples per user from 891 to 458 and 245, respectively, and the number of unique application in the test data went from 61 to 60 and 45. As can be seen from Table VI, the lowest EERs for any value of n are obtained when all observations are considered. The table indicates that ignoring both unknown applications and unforeseen observations make the verification task difficult. This is because, the distribution of unknowns presented in Section IV-A showed that the existence of unknown applications can be very useful to differentiate between users. So, even though, a binary decision solely based on unknowns would be misleading (according to our findings in Section IV-B), an intelligent decision incorporating the unknown applications and unforeseen events could be rewarding as can be seen in Table VI. Also, for practical purposes, ignoring samples will cause latency in decision making, which can greatly reduce the recall of an active authentication system.

TABLE VI
PERFORMANCE OF THE M-ED ALGORITHM IN TERMS OF EER (%) FOR THREE TYPES OF TEST SEQUENCES - ALL OBSERVATIONS, ALL EXCEPT THE ONES WITH UNKNOWN APPLICATIONS AND ALL WITHOUT UNFORESEEN OBSERVATIONS. EXPERIMENT PERFORMED ON THE UMDAA-02 DATASET WITH FIXED SAMPLING RATE AT $1/30s^{-1}$.

n	%EER		
	All Obs.	No Unknown Apps.	No Unforeseen Obs.
20	43.20	49.22	48.96
30	39.03	44.72	46.70
40	36.97	43.64	45.01
50	35.53	42.19	44.16
60	34.31	42.47	43.29

V. Suitable Modeling Techniques

In this section, some suitable modeling approaches for the application usage-based active authentication problem are discussed. In light of the outcomes of the experiments presented in the previous section, it can be asserted that the application-usage-based verification models must be capable of taking into account unknown applications and unforeseen observations. A popular approach to model temporal data sequences is to use state-space models such as Mobility Markov Chains or Hidden Markov Models (HMM) which can model time variation of the data. However, these methods are not capable of handling unforeseen events by default. For example, any unforeseen event will be given a zero emission probability in these models, and therefore, the models will be somewhat like the binary decision model that was discussed earlier.

Algorithm 3 Pseudo code for the modified edit-Distance algorithm.

```
procedure M-ED($S^{tr}$, $S^{te}$)▷ Training observation sequence of a user ($S^{tr}$) of length $N$, $n$-last Test observation Sequence of any user ($S^{te}$), where $N > n$.
    $D \leftarrow [1, 2, \ldots, n]$
    for $j = 0$ to $n - 1$ do
        $d \leftarrow zeros[0 : N]$
        $d[0] \leftarrow [j + 1]$
        for $i = 0$ to $(N - 1)$ do
            if $S^{te}[j] == S^{tr}[i]$ then
                $d[i + 1] \leftarrow D[j]$    ▷ Exact match, no operation needed.
            else
                $A_1, T_1, W_1 \leftarrow S^{tr}[i]$    ▷ Extract application name, timezone and day flag from the observations.
                $A_2, T_2, W_2 \leftarrow S^{te}[j]$    ▷ Extract application name, timezone and day flag from the observations.
                NOp $\leftarrow 0$
                if $A_1 == A_2$ and $(T_1 == T_2$ or $W_1 == W_2)$ then
                    NOp $\leftarrow 1$    ▷ One substitution needed if only timezone or day does not match.
                else if $A_1 == A_2$ then
                    NOp $\leftarrow 2$    ▷ Two substitution needed if neither timezone nor day are matching.
                else
                    NOp $\leftarrow 3$    ▷ Three substitution operation for no match.
                end if
                $d[i + 1] \leftarrow$ NOp+ $\min(D[j], D[j + 1], d[i])$
            end if
            $D[j] \leftarrow d[i + 1]$
        end for
    end for
    return $D[n - 1]$
end procedure
```

However, simple modifications to these models can improve the usability of these methods when unforeseen events are present as discussed in [25] for geo-location data. In this paper, the three state-space models namely, the Markov Chain (MC)-based Verification, HMM with Laplacian Smoothing (HMM-lap) and Marginally Smoothed HMM (MSHMM), described in [25] are employed on the application-usage-based verification task and the performances are compared.

For the MC method, the prior probability for unknown and unforeseen events are set to a very small nonzero probability of $\delta = e^{-20}$ (Laplace-smoothing) when training a model X_T for observation sequences of length T. For MC, the probability of transitioning to an observation state o_j depends only on the probability of the last observation state o_i, i.e.

$$\tau_{i,j} = Prob(X_T = o_j | X_{T-1} = o_i). \tag{7}$$

If the prior probability of entering any state i is $p_i = Prob\{X_0 = i\}$ with respect to the set of observations for user-z O_T^z , then the total probability of traversing any sequence of n consecutive observations $i_0, \ldots, i_n \in O_T^z$ is calculated as

$$Prob(X_0 = i_0, \ldots, X_n = i_n) = p_{i_0} \tau_{i_0,i_1} \ldots \tau_{i_{n-1},i_n} \tag{8}$$

Similar to the MC method, in HMM-lap method Laplacian Smoothing of the emission probabilities is considered with HMM to incorporate unforeseen observations as discussed in [25]. The number of hidden states is fixed to 20 for all the experiments and the maximum number of iteration is set to 50.

The most suitable approach for handling unforeseen observations is the Marginally Smoothed Hidden Markov Model (MSHMM) introduced in [25]. To adopt the approach for the active authentication problem, the marginal probabilities of the presence of an application in the training sequence of a user for each time-zone and day flags are precomputed. Assuming that the probability of user-x using application a_x^i at time-zone $TZ(t)$ at time t, $P(a_x^i, T_j)$ is independent of the probability of user-x using the application at location $W(t)$, $P(a_x^i, W(t))$ at time t, the emission probability from state s to observation o_t, $\hat{e_s}(o_t)$ is

$$\hat{e_s}(o_t) \quad = \quad P(O_t^{\{a_x, TZ(t), W(t)\}} = o_t^{\{a_x, TZ(t), W(t)\}} | X_t = s) \tag{9}$$

if $o_t \in O_t^{\{a_x^p, TZ(t), W(t)\}}$. Otherwise,

$$\begin{aligned} \hat{e_s}(o_t) \quad = \quad & P(O_t^{\{a_x, TZ(t)\}} = o^{\{a_x, TZ(t)\}} | X_t = s) \times \\ & P(O_t^{\{a_x^p, W(t)\}} = o_t^{\{a_x, W(t)\}} | X_t = s), \end{aligned} \tag{10}$$

where $P(o_t^{\{a_x, TZ(t)\}} = max(\delta, P(a_x, TZ(t)))$ and, $P(o_t^{\{a_x, W(t)\}} = max(\delta, P(a_x, W(t)))$. By definition, the MSHMM approach is capable of differentiating between unknown applications and unforeseen observations with

TABLE VII
PERFORMANCE OF THE MSHMM ALGORITHM IN TERMS OF EER (%) FOR THREE TYPES OF TEST SEQUENCES - ALL OBSERVATIONS, ALL EXCEPT THE ONES WITH UNKNOWN APPLICATIONS AND ALL WITHOUT UNFORESEEN OBSERVATIONS. EXPERIMENT PERFORMED ON THE UMDAA-02 DATASET WITH FIXED SAMPLING RATE AT $1/30s^{-1}$.

n	%EER		
	All Obs.	**No Unknown Apps.**	**No Unforeseen Obs.**
20	35.53	45.38	46.09
30	36	45.13	40.47
40	33	44.03	44.09
50	31.82	44.23	49.35
60	30.3	45.95	38.33

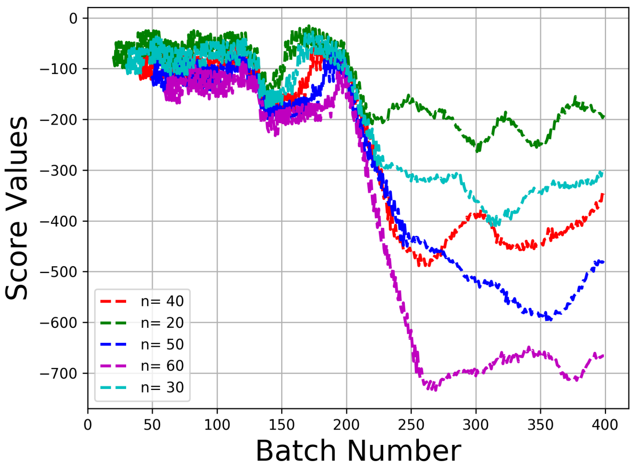


Fig. 7. Average change in MSHMM scores in response to intrusion on the UMDAA-02 application-usage data.

known applications, as well as, the more frequent vs. less frequent applications occurring at different time zones and days.

VI. EXPERIMENTAL RESULTS

In this section, the experimental results for the different verification methods are discussed in detail for performance comparison.

First, the impact of ignoring unknown applications and unforeseen observations on the overall verification of MSHMM, a state-space method, for various n is presented in Table VII. As can be seen from the table, both ignoring unknown apps and unforeseen observations results in much lower EER in comparison to the cases when all test observations are considered. There results resonate with our findings in Table VI using the M-ED algorithm.

The performances of M-ED, MMC, HMM-lap and MSHMM algorithms for the full test sequences of the UMDAA-02 application usage dataset are shown in Table VIII, where, the sampling rate has been varied from one sample every 5 seconds to one sample every 30 seconds with intervals of 5 seconds, while the number of previous observations n is varied from 20 to 60 with intervals of 10. It can be seen from the table that with smaller sampling rate and bigger n, the EER drops for all the methods. The MSHMM outperforms every other method in every case, which can be attributed to the improved modeling capability of the method due to marginal smoothing. For a practical verification system, the sampling rate and value of n would determine the latency of decision making. In many cases, a sample every 30 second might be too late and therefore the system designer should choose these parameters carefully.

As for n, intuitively with more historical data the performance should improve all the time. In order to determine the impact of n and also to get an idea about the latency of MSHMM when intrusion occurs, a different experiment was performed where a different user's data is appended with the legitimate user's data to simulate intrusion. To be more precise, for each user of the UMDAA-02 dataset, 200 consecutive observations from the test sequence starting from a random index are appended with 200 consecutive observations from the test sequences of all the other users (start index picked randomly) and the whole sequence is evaluated using MSHMM for different n values. The average score values across all users are plotted in Fig. 7 for different n values. When the observations from a different user starts to enter a batch (at 200-th batch), the average scores returned by MSHMM for each batch drops vividly, as can be seen from the figure. Also, the figure clearly shows the drop is larger for large n values - proving the intuition that considering more historical data is advantageous in this regard. As for latency, if the score of -200 is considered as a threshold for decision making, then for all $n = 60$, the intrusion will be detected within ~ 5 batches, i.e. withing 2.5 minutes from the inception of intrusion.

Finally, for the Securacy dataset, the performances of MSHMM, HMM-lap, MMC and M-ED are presented in Table. IX. Similar to the UMDAA-02 dataset results, MSHMM outperforms the other methods by a good margin. Note that the EER values are much lower for this dataset for the state-space models, which is understandable since it has already been demonstrated in Fig. 3(c) that the users are quite separable in this dataset even if only application names are considered. However, M-ED faces difficulty in exploiting the separability of the observations since is not capable of modeling temporal variations as effectively as state-space models.

Based on results of the experiments presented in this paper, it can be asserted that application-usage data might be useful as a soft biometric for bolstering the decision in a multi-modal user authentication scenario. Given the fact that the application-usage data is readily available and easy to track without using much battery or computational power, real-time score generation is possible. The experiments also depict that the verification scores show rapid change for intrusion within several minutes. Hence, the latency is not too high for a soft biometric measure. However, even though state-space models

TABLE VIII
APPLICATION-USAGE-BASED VERIFICATION PERFORMANCE COMPARISON FOR UMDAA-02 DATASET ACROSS DIFFERENT METHODS BASED ON EER (%) FOR VARYING SEQUENCE LENGTH (N) AND SAMPLING RATE. THE NUMBER OF HIDDEN STATES IS FIXED AT 20 AND MAXIMUM NUMBER OF ITERATION IS 50 FOR HMM-BASED METHODS.

n	**Method**	**Sampling Rate**					
		1/5	**1/10**	**1/15**	**1/20**	**1/25**	**1/30**
20	**M-ED**	42.96	42.92	44.12	43.64	43.09	43.2
	MMC	40.86	40.53	40.27	39.48	40.39	36.78
	HMM-lap	38.49	38.35	37.82	37.39	38.83	36.77
	MSHMM	37.3	37.3	36.67	35.93	35.63	34.82
30	**M-ED**	42.7	41.71	40.18	38.17	37.58	39.03
	MMC	40.29	39.18	38.21	40	39.04	36.82
	HMM-lap	37.28	37.2	36.68	37.73	37.89	37.45
	MSHMM	36.23	36.87	35.74	36.87	35.99	35.79
40	**M-ED**	41.7	38.64	38.41	38.13	37.45	36.97
	MMC	39.29	40.57	38.13	39.62	41.97	35.89
	HMM-lap	37.37	37.88	36.75	36.07	39.11	34.62
	MSHMM	35.4	35.65	34.026	34.4	36.58	32.54
50	**M-ED**	40.69	37.98	36.19	35.55	35.58	35.53
	MMC	40.34	37.92	38.67	36.96	39.57	33.56
	HMM-lap	36.97	36.01	36.48	34.72	36.7	33.95
	MSHMM	35.95	34.41	34.67	32.41	35.27	30
60	**M-ED**	38.69	35.93	35.32	35.72	34.97	34.31
	MMC	38.33	37.5	37.5	38.01	35.91	34.35
	HMM-lap	35.31	35.48	34.18	33.15	36.05	34.35
	MSHMM	34.036	34.92	32.78	33.33	34.3	31.93

TABLE IX
APP-BASED VERIFICATION EER(%) COMPARISON FOR SECURACY DATASET ACROSS DIFFERENT METHODS [25] FOR DIFFERENT n VALUES. NUMBER OF HIDDEN STATES IS SET TO 20 AND SAMPLING RATE IS $1/30s^{-1}$.

n	**MSHMM**	**MMC**	**HMM-lap**	**M-ED**
20	17.23	19.286	19.66	35.09
30	16.75	18.9967	19.59	32.88
40	16.38	18.7074	19.19	31.4
50	16.26	17.9475	19.22	30.53
60	16.16	17.6443	18.38	30.58

can be made to work well with some modifications, the equal error rate for a diverse dataset is still around $\sim 16\%$, which needs further improvement. In this regard, bigger training datasets and keeping longer usage history might be helpful. In addition, if computational constraints can be loosened, then more sophisticated high-performance methods such as deep neural networks can be employed to minimize the EER.

VII. CONCLUSION

In this paper, the challenging problem of active authentication using application usage data has been formulated and systematically tackled to obtain viable solutions. Through several experiments, the impact of unknown applications and unforeseen observations on the authentication problem has been investigated and it is shown that for this problem inclusion of the uncertain events are necessary to obtain better performances. In this regard, a modified edit distance algorithm has been introduced, the performance of which is compared with three state-space models namely, Markov Chain, HMM with Laplacian Smoothing and Marginally-Smoothed HMM, in terms of EER. Experiments were performed on the UMDAA-02 and the Securacy application-usage datasets. The experiments revealed some very interesting insights about the differences between the two datasets. Also, the paper addressed different aspects of important practical considerations such as intrusion detection, latency, observation history and sampling rate. As for future work, the M-ED method might be further improved by varying the distances for the three different cases based on the marginal probabilities. Also, recurrent neural network (RNN)-based models might be able to learn more discriminative properties of application-usage patterns. However, RNNs require huge amount of data for training, which the two datasets presented here lack. Another interesting research direction would be the joint training of application sequence and some other sequential data such as the location data to improve the authentication performance. Finally, since application information are also suitable context for other modalities, application data sequences can have dual utilization (as a separate modality and also as context) in more advanced active authentication schemes.

Acknowledgment

This work is partially funded by the Emil Aaltonen Foundation (170117 KO), the Finnish Foundation for Technology Promotion (PoDoCo program), Academy of Finland (Grants 286386-CPDSS, 285459-iSCIENCE, 304925-CARE, 313224-STOP, SENSATE), and Marie Skodowska-Curie Actions (645706-GRAGE)

References

[1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997.

[2] H. Cao and M. Lin. Mining smartphone data for app usage prediction and recommendations: A survey. *Pervasive and Mobile Computing*, 37:1 – 22, 2017.

[3] K. Church, D. Ferreira, N. Banovic, and K. Lyons. Understanding the challenges of mobile phone usage data. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '15, pages 504–514, New York, NY, USA, 2015. ACM.

[4] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann. Touch me once and i know it's you!: Implicit authentication based on touch screen patterns. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 987–996, New York, NY, USA, 2012. ACM.

[5] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in smartphone usage. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 179–194, New York, NY, USA, 2010. ACM.

[6] M. E. Fathy, V. M. Patel, and R. Chellappa. Face-based active authentication on mobile devices. In *IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP)*, 2015.

[7] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen. Continuous mobile authentication using touchscreen gestures. In *Homeland Security (HST), 2012 IEEE Conf. on Technologies for*, pages 451–456, Nov. 2012.

[8] T. Feng, J. Yang, Z. Yan, E. M. Tapia, and W. Shi. Tips: Context-aware implicit user identification using touch screen in uncontrolled environments. In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, HotMobile '14, pages 9:1–9:6, New York, NY, USA, 2014. ACM.

[9] D. Ferreira, E. Ferreira, J. Goncalves, V. Kostakos, and A. K. Dey. Revisiting human-battery interaction with an interactive battery interface. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pages 563–572, New York, NY, USA, 2013. ACM.

[10] D. Ferreira, V. Kostakos, A. R. Beresford, J. Lindqvist, and A. K. Dey. Securacy: An empirical investigation of android applications' network usage, privacy and security. In *ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '15, pages 11:1–11:11. ACM, 2015.

[11] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions on Information Forensics and Security*, 8(1):136–148, Jan. 2013.

[12] L. Fridman, S. Weber, R. Greenstadt, and M. Kam. Active authentication on mobile devices via stylometry, application usage, web browsing, and GPS location. *IEEE Systems Journal*, 11(2):513–521, 2017.

[13] S. Gambs, M.-O. Killijian, and M. N. n. del Prado Cortez. Show me how you move and i will tell you who you are. In *Proc. 3rd ACM SIGSPATIAL Int. Workshop on Security and Privacy in GIS and LBS*, SPRINGL '10, pages 34–41, 2010.

[14] S. Gupta, A. Buriro, and B. Crispo. Demystifying authentication concepts in smartphones: Ways and types to secure access. *Mobile Information Systems*, 2018:16 pages, 2018.

[15] A. Hadid, J. Heikkila, O. Silven, and M. Pietikainen. Face and eye detection for person authentication in mobile phones. In *Distributed Smart Cameras. ICDSC '07. First ACM/IEEE Int. Conf.*, pages 101–108, Sept. 2007.

[16] A. F. Hugo Gamboa. A behavioral biometric system based on human-computer interaction, 2004.

[17] S. L. Jones, D. Ferreira, S. Hosio, J. Goncalves, and V. Kostakos. Revisitation analysis of smartphone app use. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, pages 1197–1208, New York, NY, USA, 2015. ACM.

[18] J. Kaasila, D. Ferreira, V. Kostakos, and T. Ojala. Testdroid: Automated remote ui testing on android. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, pages 28:1–28:4, New York, NY, USA, 2012. ACM.

[19] H. Khan and U. Hengartner. Towards application-centric implicit authentication on smartphones. In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, HotMobile '14, pages 10:1–10:6, New York, NY, USA, 2014. ACM.

[20] V. Kostakos, D. Ferreira, J. Goncalves, and S. Hosio. Modelling smartphone usage: A markov state transition model. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 486–497, New York, NY, USA, 2016. ACM.

[21] W.-H. Lee, X. Liu, Y. Shen, H. Jin, and R. B. Lee. 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*, SACMAT '17 Abstracts, pages 67–78, New York, NY, USA, 2017. ACM.

[22] H. Li, X. Liu, and Q. Mei. Predicting smartphone battery life based on comprehensive and real-time usage data. *CoRR*, abs/1801.04069, 2018.

[23] Y. Liu and A. Simpson. Privacy-preserving targeted mobile advertising: requirements, design and a prototype implementation. *Software: Practice and Experience*, 46(12):1657–1684.

[24] A. Loureno, H. Silva, and A. Fred. Ecg-based biometrics: A real time classification approach. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, Sep. 2012.

[25] U. Mahbub and R. Chellappa. Path: Person authentication using trace histories. In *2016 IEEE 7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pages 1–8, Oct 2016.

[26] U. Mahbub, S. Sarkar, V. M. Patel, and R. Chellappa. Active user authentication for smartphones: A challenge data set and benchmark results. In *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 7th Int. Conf.*, Sep. 2016.

[27] U. Mahbub, S. Sarkar, V. M. Patel, and R. Chellappa. Active user authentication for smartphones: A challenge data set and benchmark results. In *IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2016.

[28] A. Mahfouz, T. M. Mahmoud, and A. S. Eldin. A survey on behavioral biometric authentication on smartphones. *Journal of Information Security and Applications*, 37:28 – 37, 2017.

[29] S. Mondal and P. Bours. Does context matter for the performance of continuous authentication biometric systems? An empirical study on mobile device. In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–5, 2015.

[30] F. Monrose and A. D. Rubin. Keystroke dynamics as a biometric for authentication. *Future Gener. Comput. Syst.*, 16(4):351–359, Feb. 2000.

[31] R. Murmuria, A. Stavrou, D. Barbará, and D. Fleck. Continuous authentication on mobile devices using power consumption, touch gestures and physical movement of users. In *International Workshop on Recent Advances in Intrusion Detection*, pages 405–424. Springer, 2015.

[32] N. Neverova, C. Wolf, G. Lacey, L. Fridman, D. Chandra, B. Barbello, and G. Taylor. Learning human identity from motion patterns. *IEEE Access*, 4:1810–1820, 2016.

[33] E. G. Ortiz, A. Wright, and M. Shah. Face recognition in movie trailers via mean sequence sparse representation-based classification. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3531–3538, June 2013.

[34] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello. Continuous user authentication on mobile devices: Recent progress and remaining challenges. *IEEE Signal Processing Magazine*, 33(4):49–61, July 2016.

[35] A. Primo, V. Phoha, R. Kumar, and A. Serwadda. Context-aware active authentication using smartphone accelerometer measurements. In

Comput. Vision and Pattern Recognition Workshops, IEEE Conf., pages 98–105, June 2014.

[36] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos. Progressive authentication: Deciding when to authenticate on mobile phones. In *Proc. of the 21st USENIX Conf. on Security Symp.*, Security'12, pages 15–15, Berkeley, CA, USA, 2012. USENIX Association.

[37] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon. Biometric-rich gestures: A novel approach to authentication on multi-touch devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 977–986, New York, NY, USA, 2012. ACM.

[38] P. Samangouei, V. M. Patel, and R. Chellappa. Facial attributes for active authentication on mobile devices. *Image and Vision Computing*, 58:181 – 192, 2017.

[39] A. Shema and D. E. Acuna. Show me your app usage and i will tell who your close friends are: Predicting user's context from simple cellphone activity. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, pages 2929–2935, New York, NY, USA, 2017. ACM.

[40] P. Siirtola, J. Komulainen, and V. Kellokumpu. Effect of context in swipe gesture-based continuous authentication on smartphones. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2018.

[41] V. Srinivasan, S. Moghaddam, A. Mukherji, K. K. Rachuri, C. Xu, and E. M. Tapia. Mobileminer: Mining your frequent patterns on your phone. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 389–400, New York, NY, USA, 2014. ACM.

[42] L. Sydow and S. Cheney. 2017 retrospective: A monumental year for the app economy. Technical report, App Annie, 2018.

[43] A. Tongaonkar, S. Dai, A. Nucci, and D. Song. Understanding mobile app usage patterns using in-app advertisements. In M. Roughan and R. Chang, editors, *Passive and Active Measurement*, pages 63–72, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[44] N. van Berkel, C. Luo, T. Anagnostopoulos, D. Ferreira, J. Goncalves, S. Hosio, and V. Kostakos. A systematic assessment of smartphone usage gaps. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4711–4721, New York, NY, USA, 2016. ACM.

[45] P. Welke, I. Andone, K. Blaszkiewicz, and A. Markowetz. Differentiating smartphone users by app usage. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 519–523, New York, NY, USA, 2016. ACM.

[46] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, Feb 2009.

[47] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, and S. Venkataraman. Identifying diverse usage behaviors of smartphone apps. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 329–344, New York, NY, USA, 2011. ACM.

[48] Y. Xu, M. Lin, H. Lu, G. Cardone, N. Lane, Z. Chen, A. Campbell, and T. Choudhury. Preference, context and communities: A multi-faceted approach to predicting smartphone app usage patterns. In *Proceedings of the 2013 International Symposium on Wearable Computers*, ISWC '13, pages 69–76, New York, NY, USA, 2013. ACM.

[49] H. Zhang, V. M. Patel, and R. Chellappa. Robust multimodal recognition via multitask multivariate low-rank representations. In *IEEE Int. Conf. Automat. Face and Gesture Recogn.* IEEE, 2015.

[50] H. Zhang, V. M. Patel, M. E. Fathy, and R. Chellappa. Touch gesture-based active user authentication using dictionaries. In *IEEE Winter Conf. Applicat. of Comput. Vision*. IEEE, 2015.

[51] N. Zheng, A. Paloski, and H. Wang. An efficient user verification system via mouse movements. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS '11, pages 139–150, New York, NY, USA, 2011. ACM.

Dr. **Upal Mahbub**, currently a Senior Engineer at Qualcomm, San Diego, California, received the M.Sc. and Ph.D. degrees in Electrical and Computer Engineering from the University of Maryland College Park in 2017 and 2018, respectively. His doctoral research was primarily focused on multi-modal active user authentication on smartphones using computer vision and machine learning techniques. Upal is the recipient of the best paper award at IEEE UEMCON 2016, best poster award at BTAS 2016, best paper award at ICCIT 2011 and distinguished graduate fellowship from the A. James Clark School of Engineering at the University of Maryland.

Jukka Komulainen received the M.Sc. and D.Sc. degrees in information engineering from the University of Oulu, in 2010 and 2015, respectively. He is currently a Computer Vision Engineer at Visidon Ltd, Oulu, Finland. His general research interests include machine learning, computer vision, image processing and computational imaging. His particular focus has been on biometrics with emphasis on presentation attack detection (face, iris and audiovisual) and multi-modal active authentication. He was a recipient of the IET Biometrics Premium (Best Paper) Award in 2013 and the Five Year Highest Impact Award for BTAS Paper in 2016.

Denzil Ferreira is Adjunct Professor, Senior Research Fellow and an Academy of Finland Research Fellow at the University of Oulu, Faculty of Information Technology and Electrical Engineering (ITEE), the Deputy Director of the Center for Ubiquitous Computing and the Principal Investigator of the Community Instrumentation and Awareness (CIA) research group. His main research interest is on technology-driven human behavior sensing and modeling, where he juxtapose methods from large-scale data analysis, sensor instrumentation, applied machine learning, mobile and ubiquitous computing to understand and study a variety of human behavioral and social phenomena in naturalistic settings.

Prof. **Rama Chellappa** is a Distinguished University Professor and a Minta Martin Professor of Engineering and in the Department of Electrical and Computer engineering at the University of Maryland (UMD). He is a recipient of the K.S. Fu Prize from the International Association of Pattern Recognition (IAPR), the Society, Technical Achievement and Meritorious Service Awards from the IEEE

Signal Processing Society and the Technical Achievement and Meritorious Service Awards from the IEEE Computer Society. He also received the Inaugural Leadership Award from the IEEE Biometrics Council. At UMD, he has received college and university level recognitions for research, teaching, innovation and mentoring of undergraduate students. He has been recognized with an Outstanding ECE Award and a Distinguished Alumni Award from Purdue University and the Indian Institute of Science, respectively. He is a Fellow of IEEE, IAPR, OSA, AAAS, ACM, and AAAI and holds six patents. His current researcher interests are computer vision, pattern recognition and machine intelligence.