IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 8, 2020, accepted May 14, 2020, date of publication May 21, 2020, date of current version June 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2996233

A Multi-Feature User Authentication Model Based on Mobile App Interactions

YOSEF ASHIBANI, (Member, IEEE), AND QUSAY H. MAHMOUD (Senior Member, IEEE)
Department of Electrical, Computer and Software Engineering, Ontario Tech University, Oshawa, ON L1G 0C5, Canada
Corresponding author: Yosef Ashibani (yosef.ashibani@ontariotechu.net)

ABSTRACT Knowledge-based authentication approaches such as the use of passwords and personal identification numbers (PINs) are the most common ways of authenticating users. The main problem with such approach is relying on simple authentication login credentials at the login stage, and assuming the user is still the same between access sessions makes applications and networks vulnerable to access by unauthorized users. Application-level access patterns on smartphone and tablet devices can be utilized to provide an approach for continuously authenticating and identifying users. This paper presents a user authentication and identification method based on mobile application access patterns, and throughout the paper we use a smart home environment as a motivating scenario. To enhance the classification process, many features have been extracted and utilized which considerably improved differentiating between users and eliminating similarities in the access usage patterns. The proposed model has been evaluated using two datasets, and the results show an ability to authenticate users with high accuracy in terms of low false positive, false negative, and equal error rates. Overall, the statistical analysis of the extracted multi-features and the results show that the feasibility of decision-making based on app interactions can lead to high accuracy.

INDEX TERMS Mobile app interactions, continuous user authentication and identification, multi-class classification, smart home networks.

I. INTRODUCTION

A smart home can be defined as a home equipped with connected Internet-of-Things (IoT) devices that can be remotely accessed and controlled. In addition to accessing, operating, and controlling home appliances, smart home networks provide many other services to home residents, such as entertainment storage information and personal files. For example, Wink [1], Samsung's SmartThings and HomeKit [2] are smart home platforms. These home platforms are built based on the cloud backend service where control management and authentication are performed mostly through an installed application on end-user devices, such as smartphones and tablets. Consequently, access to smart home services is mostly achieved remotely through the users' end-devices which have become essential tools for accessing and operating smart home networks. Although smart home systems conveniently provide services to home residents, there are many security issues that need to be considered. One of the issues is the fact that the smartphone is susceptible to loss or theft. Therefore, there is a need for a transparent authentication and identification mechanism that can implicitly authenticate and identify the user without requiring more explicit intervention. Thus, smart home networks should be enhanced with security measures because some of the common authentication mechanisms that are still in use today have weaknesses. For example, knowledge-based authentication approaches, such as the use of passwords and PINs, are considered convenient for users. However, it is impractical for users to continuously provide knowledge-based credentials for continuous authentication. Another approach, which uses object-based authentication such as tokens, has been developed. Another approach such as fingerprint authentication can add more security levels to the mobile phone itself, but this approach is considered as a point of entry. Furthermore, in addition to the difficulty of utilizing this approach for continuous authentication, it cannot guarantee that the right user is still using the device. The focus of both academic research and commercial products has turned to behavioral-based authentication as a result of the weakness inherent in both knowledge-based and object-based authentication mechanisms, as well as the complications involved

The associate editor coordinating the review of this manuscript and approving it for publication was Liang-Bi Chen.

in employing physiological biometrics for continuous and implicit authentication. The change in approaches aims to find security solutions that have the capability to:

- Identify that access request is coming from the registered and authorized user;
- Continuously authenticate and identify the user;
- Block access requests that come from unauthorized users.

The behavior based authentication process verifies the user's identity, namely whether legitimate or unknown, while the user identification process verifies the current user's identity among other enrolled users. Hence, continuous authentication is the process of repeatedly checking the user's identity at and beyond the login stage, whereas continuous identification is the process of regularly checking user identity from among other enrolled users. One advantage of implicit authentication and identification, which is based on user interaction (access) to apps, is that it provides security and usability while reducing explicit user intervention unless it is required. However, there are factors, such as the similarity between users' routines in accessing apps and access duration, that need to be considered in order to generate multi-features that can enhance the classification process.

Due to the advantages of the functions that the mobile device can provide, considerable information can be used to support continuous authentication and identification. One such approach is application (app) based behavior profiling, which utilizes app access history to classify user access patterns. For instance, the ability to continuously retrieve app usage profile data on mobile devices while running in the background strengthens the argument for employing behavioral-based mechanisms for continuous user authentication and identification. Additionally, utilizing user app interaction can be profiled on smartphones/tablets based on functions provided by most of the operating systems on these devices. This approach can be utilized at the entry point to the network and beyond, thereby providing more security protection.

A. MOTIVATION

Mobile phones and tablets are becoming global control-end devices in smart home networks. In addition to the pre-installed apps by mobile companies, over two million mobile apps are available in major app stores as well as those added daily [3]. As well as the apps available in stores, the number of Android apps at the Google Play are approximately two milling [4], in addition to those new apps that are added daily. Moreover, the number of apps used is increasing and most interactions to mobile phone devices are related to foreground apps. The interaction intervals that range from user to user can be used to differentiate users. With an increasing number of apps being used on mobile phones, app usage behavior may provide information to differentiate between users. Furthermore, app-based access exemplifies information on user patterns while interacting on mobile devices. Several proposed solutions are targeted for use on mobile phones and utilized for only the owner (single) user of the mobile device. Hence, a continuous authentication and identification approach based on app usage behavior is required as a second layer authentication and identification for smart home networks. Our target is to protect access to home appliances and services against situations listed in the threat model part of this paper in Section III. As a consequence, the main objective of this research is to design a continuous and transparent authentication and identification method that requires minimal user intervention in order to protect smart home devices from unauthorized access in addition to protecting the access of users who have weak or no security protection on their mobile devices.

This paper presents a user authentication and identification approach, utilizing app access events. Ultimately, the main contribution of this paper is an authentication model to continuously authenticate and identify users by utilizing apps used on their mobile devices. This approach:

- Has the advantage of being able to work by including most mobile devices regardless of the operating system or the hardware on these devices.
- Is able to authenticate and identify registered users utilizing app interactions on their mobile devices with a considerable high accuracy.
- Ensures that the utilized features are generalized, hence can be extracted from most mobile devices regardless of the device operating systems on these devices or type of hardware.
- Offloads the computation of model building and testing of the mobile device to the smart home (hub) controller. Consequently, all the required processes and memory are performed at the home hub side, thus removing the burden from the mobile device.

Moreover, while it does not need much power consumption, the presented data collection mechanism on the mobile device considers privacy protection of the collected information and enables the authentication and identification of users based on a small number of app access events prior to the access request and beyond the login stage. The proposed approach preserves users' privacy by keeping their information on a secure server on the home network, not on the mobile device. Therefore, in the case of loss or theft of the device, no data related to the user behavior will be leaked. Furthermore, as the proposed solution is centralized, the previously built model is already available to the home hub, with no need to rebuild the model template for users. In addition, each user will have a single behavior profile that can be utilized for authentication and identification when accessing from different devices without the need to retrain a new model for the user.

The rest of this paper is organized as follows. Section II discusses related work. The proposed model is presented in Section III, while Section IV presents the experimental evaluation and results. Finally, Section V concludes the paper and offers ideas for future research.

TABLE 1. A comparison of behavioral-based authentication approaches.

Ref No.	Utilized information, Authentication Method and Accuracy Measure	Advantage	Limitations
[8]	Calls, SMS, Web browsing history; 35 iPhone users for 100 calls, 1698 SMS and 13 Web browsing history; One-vs-all; Bayesian networks, RBF, KNN, random forest (RF), SVM, Multi-layer Perceptron (MLP); Average TPR=99.3%, Average FPR < 0.7%, Average EER=1.6%.	Provides illegitimate user detection and assigns a score to observed events such as a good or habitual event.	SMS and calling functions are ignored and replaced with apps that achieve the same purpose; This study excludes new apps.
[13]	Time of last viewed email; GPS location for three months; Clustering; Provides an overall score for authentication decision.	Provides an overall score for authentication decisions.	Considers only one app and uses only GPS location, which will not be available in most cases, especially indoors; This study excludes new apps.
[17]	101 unique apps or telephone numbers, calls, SMSs for four weeks; One-vs-all; RBF, MLP EER= 9.8%, FRR=11.45%, FAR= 4.17%.	Continuously verifies mobile users.	Verification will not be performed unless a total of 6 applications have been utilized; Evaluation is based on simulation users; This study excludes new apps.
[18]	Apps usage, location, clock time, gesture, voice, touch; Single model per user; Naïve Bayes; FAR, FRR, TAR, TRR.	Combines several features, resulting in universal and unique modality for users.	SMS count only for the past hour in addition to the time and GPS location; SMS and calling functions are ignored and replaced with apps that achieve the same purpose; This study excludes new apps.
[19]	Unique app usage data; 26 users and 99 users from different datasets; Verification models per user trained only on positive samples; EER= 16:16%, EER= up to 31.82 from unforeseen apps.	Presents a continuous authentication model for smartphones based on app usage data.	The proposed approach needs 2-5 minutes of app usage to detect an intrusion. In addition, it considers apps from different languages which can be easily differentiate between users; For the active authentication problem, the preferred language of the user is a type of behavioral data that can be used to differentiate between users; Considers apps that are used only by individual users.
[7]	Text entered via soft keyboard, apps, websites visited, location; One-vs-all; SVMs; ERR of 5% after one minute of user interaction with the device, and an EER of 1% after 30 minutes	Utilizes both GPS and WiFi based location.	A binary classifier is constructed for each of the 200 users and 4 modalities; Total, of 800 classifiers; This study excludes new apps. Five-minute threshold for what is considered an idle period.
[23]	Power consumption of six popular apps; Uses an on-line power estimation tool to determine system-level power consumption; Average EER of less than 10%.	Requires no external measurement equipment.	Uses built-in battery voltage sensors and knowledge of battery discharge behavior to monitor power consumption; Modeling power consumption only for specific apps is challenging due to other apps running in the background.
[24]	Touchscreen logs; 41 users; SVM, KNN; Misclassification EER in the range of 0% to 4%, Median EER of 0% for intrasession authentication; 2%–3% for intersession authentication.	Provides a proof-of-concept classification framework that extracts different behavioral features from raw touchscreen interaction data.	For the primary study, overall experiment time ranged between 25 to 50 minutes per subject; For huge datasets, the limitation of this method is that not all data can be stored.
[20]	Most used apps and location; EER =9.004% with the first dataset; EER=1.98% with the second dataset.	Presents a continuous authentication for smartphone user based on app usage data.	Evaluation the proposed approach is based on ten consecutive days training dataset; This study excludes new apps.

II. RELATED WORK

User behavior profiling has been considered in many studies for many purposes such as authentication and intrusion as well as fraud detection. Table 1 provides a comparison of relevant related works regarding utilized information, advantages, and limitations. For example, text messages and calling behavior are considered in [5]–[7]. The study in [8] proposes an anomaly-based detection system based on monitoring users' actions, such as sending SMS messages or making calls. The focus of the earliest studies was mainly on detecting misuse behavior during interaction with the mobile network, such as calling and messaging services as presented in [9], [10]. Other studies consider mobile device sensors for user authentication, including an approach in [11] that identifies and authenticates users based on accelerometer data. This approach considers contextual information as user activities, such as walking, climbing stairs and jogging.

Authenticating mobile phone users according to accelerometer-based gait recognition, using the k-neighbor classifier (KNN) algorithm, is proposed in [12]. This approach, which records data as the user is walking, is built on the assumption that different individuals have different walking patterns. This method needs 30 seconds for authentication and requires users to follow a script. However, viewing the device and interacting with the app when the device is located on a static surface will not provide sufficient information to characterize a user based on sensor information. Therefore, it is challenging to establish authentication and identification without proper data availability. An anomaly-based detection system that monitors the actions of users, such as calls, SMSs and Web browsing on mobile phones, is presented in [8]. For performance evaluation of this work, four different machine learning classifiers were applied. Two behavioral features considered in the proposed solution in [13] are the time of the last email viewed by the user and the GPS location. These features are derived from the mobile device that is used.

A user authentication approach that utilizes the access history of app usage events employing only a small amount of information is reported in [14]. The authors in [15] show that authentication accuracy is subject to the day of the week and conclude that access to apps during weekends, when some apps are mostly accessed, should be given more weight. The work in [14], [15] is extended in [16], which presents user authentication models utilizing app access history. Two real-world datasets are used to validate the model using only shared apps during the same daily intervals. In [17], the authors present a behavior profiling framework that rejects a user's access based on the number of consecutive abnormal app usages. The evaluation results of this framework record an EER of 13.5% for basic apps, 5.4% for telephone calling, 2.2% for SMS, and 9.8% for multi-instance. A user behavior profiling, that describes where, when, how and with what the devices were used, is proposed in [18]. The work in [19] presents a continuous authentication model on smartphones based on app usage data. The achieved results in the evaluation of this method include average of EER=16% from first dataset, and 30% based on 50 historic observations sample from second dataset. However, the study considers all apps, including those that are only used by individual users. In addition, it utilizes apps from different languages; however, for the active authentication problem, the preferred language of the user is a type of behavioral data that can be used to differentiate between users.

The work in [20] presents a behavior profiling technique for user authentication on smartphones based on app usage data. For authenticating users, this method considers app names, the day and time, as well as the app use duration. Two datasets are used in the evaluation, and the achieved results include an EER=9.004% from the first dataset and EER=1.98% from the second dataset. In addition, the research reported in [14], [21], [22] shows that app access patterns, as well as the traffic generated during app access, can be applied for user authentication with reasonable accuracy, but does not consider user identification. Furthermore, the evaluation in these studies was based on classifying individual access events to apps. However, to reduce the False Positive Rate (FPR) and False Negative Rate (FAR), a pattern of more implicit features should be considered. A power model construction technique for monitoring the power consumption of each app on an electronic device is presented in [23]. This approach, which utilizes built-in battery voltage sensors and knowledge of battery discharge behavior, achieved an absolute Average Error Rate (AER) of less than 10%. However, it is challenging to model power consumption only for specific apps due to other apps running in the background.

Based on the assumption that users perform predefined repetitive tasks, a study of touch screen behavior, as described in [24], was performed on 41 users to test the applicability of screen touches. In this study, the authors were able to achieve results of Misclassification Error Rates in the range of 0% to 4%. Although it demonstrated the ability to realize a satisfactory performance of matching gestures, the analysis was limited to vertical and horizontal swipes on the used app. In this study, 30 touch features were extracted and, for training the user profile, the KNN classifier and the Gaussian RBF kernel SVM were used. These techniques presented in the related work use previous user access activity to build user usage profiles and then apply these profiles in order to identify legitimate users. This concept has been utilized within different technologies, such as mobile phones, mobile networks, and Web browsing, either for the client-side or for the server-side [25].

Other studies have adopted third-party mechanisms to offload processing capabilities and the required memory storage of mobile devices. As an example, in [26], the SmartThings home platform performs authentication and authorization based on user actions in accessing IoT home devices. The authentication procedure is achieved either at the cloud backend or at the SmartHub controller in the smart home network. As an example for the client-side perspective, the studies in [27]–[29] utilize features while accessing the computer system, such as accessed files and information and how frequently these files are accessed, in order to detect unauthorized access to the computer system. From a server-side perspective, research, such as in [30], [31], has studied the potential to build a user profile based on Web access activities in order to identify users. The utilized features in these studies include the visited website name, start time, total session time, and number of browsed pages.

The presented work in the literature is on single modality in which the built models target single users for the used device. In other words, the focus of most of the listed studies is on the client-side; from activities on the mobile device, the built profile detects illegal usage of the device from the modeled user profile. Additionally, it is clear from the related work that the home network authentication and identification process still mainly relies on knowledge-based authentication approaches that can be shared among household members in addition to other security concerns as mentioned in [32]–[34]. However, none of the related works have utilized app access patterns on mobile devices for user authentication and identification for smart home networks. A review of the relevant literature reveals that access behavior has been used in many technologies, especially at the client-side, for continuous authentication to protect against unauthorized access to mobile phones. Hence, app access patterns can be utilized to

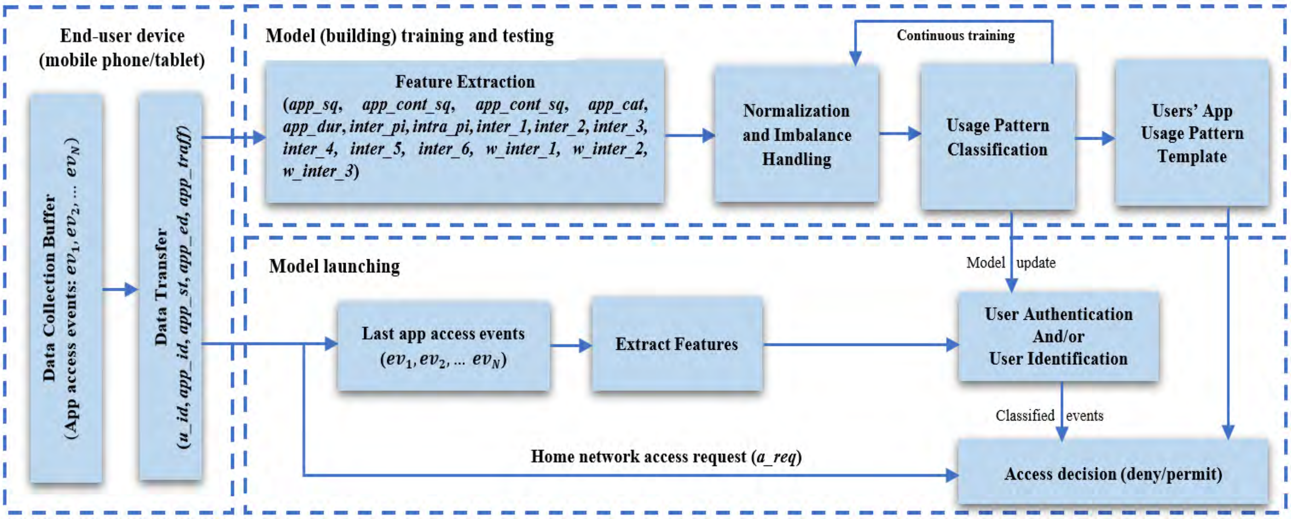
End-user device
(mobile phone/tablet)
Data Collection Buffer
(App access events: ev₁, ev₂, ... ev_N)
Data Transfer
(u_id, app_id, app_st, app_ed, app_traff)
Model (building) training and testing
Feature Extraction
(app_sq, app_cont_sq, app_cont_sq, app_cat, app_dur, inter_pi, intra_pi, inter_1, inter_2, inter_3, inter_4, inter_5, inter_6, w_inter_1, w_inter_2, w_inter_3)
Normalization and Imbalance Handling
Continuous training
Usage Pattern Classification
Users' App Usage Pattern Template
Model launching
Model update
Last app access events
(ev₁, ev₂, ... ev_N)
Extract Features
User Authentication And/or User Identification
Classified events
Home network access request (a_req)
Access decision (deny/permit)

FIGURE 1. Architecture of the proposed user authentication model.

support smart home security in the form of continuous user authentication and identification at the server-side, the smart home central hub. One advantage of this solution is that it will support multi-user behavioral models at the server-side, which reduces the resource consumption on mobile devices. Additionally, this solution could be hosted either locally on the home hub or in the cloud. Also, processing user profiles on the home smart hub offers advantages, including:

- Protecting privacy if the user's device is lost or stolen;
- Avoiding battery drain on the mobile device during pre-processing and training the model;
- Removing the need to rebuild the model in case the user changes mobile devices.

There are a number of weaknesses related to user authentication and identification on smartphones. Many smartphone users still adopt weak login credentials, including common or reusable passwords, or no password at all [35]. As an example, in some cases, users encounter urgent situations where either their devices are left unattended immediately after the login stage or where weak login credentials are applied, causing these devices to be vulnerable to unauthorized access and usage. Hence, all attacks, as mentioned in Section III in the threat model, necessitate the need for an implicit authentication and identification approach that can be utilized with less user intervention and that can quickly detect illegitimate access, consequently increasing user trust and broad adoption. Regardless of the works listed in the literature, when considering the threat scenarios presented in Section III of this paper, we believe that insider user authentication and identification has not received enough attention. Therefore, to increase the trust of homeowners, it is very important to consider these issues in presenting a robust user authentication and identification approach for smart home networks.

III. PROPOSED MODEL

This section presents the architecture of the proposed user authentication and identification model, the design goals, assumptions, and the threat model, followed by a workflow of the presented method. The proposed model includes user authentication and identification based on the user profile built from previous access history to apps on mobile devices, then makes the decision for subsequent access requests regarding legitimate user authentication and identification. This model, as shown in Figure 1, is independent of the main authentication method utilized at the entry-point (e.g. a PIN or password) and will be used at and beyond the login stage on the mobile device. This approach can be integrated as a second layer of authentication and identification in an implemented framework [36]. In this framework, a central controller hub, namely a smart home server that functions as the network controller, is responsible for user registration, authentication, and identification as well as feature collection and extraction. The presented model authenticates and identifies the user who is using the mobile device to access the smart home network. When the user requests access to the home network over the Internet or WiFi, the login page will be loaded from the home hub server via an installed app on the user's mobile device. The user then provides the login credentials, such as username and password, which will be sent back to the home central hub along with the app access history temporarily cached in the data collection buffer on the mobile device. If the authentication credentials are verified, the usage pattern is then processed to check if the request is coming from a registered user. When there is a deviation from the registered user's pattern, the usage pattern of the app access history will be tested against the other registered users' behavior templates. If the pattern is related to one of the registered home users, the access request will be accepted, and access permission will be granted based on the pre-set permission at the registration stage. Otherwise, the request will be declined, and a second-factor authentication will be requested in addition to the incident being reported to the homeowner.

The authentication and identification process is achieved at the access point and continuously during the access session to the home network. Hence, at the entry point, the authentication and identification process will result in three possibilities:

- Registered user from his/her registered device (main user);
- Registered user from another registered device (insider);
- Unregistered user from a registered device (outsider).

For the first two cases, the user will be granted access based on the assigned authorization. For the third case, the access request will be denied, and the following access from the same device will be blocked. The owner of the device will then be required to update the authentication credentials. Additionally, if a *sec_fa* is not provided, the collected app access events will count as an attacker's pattern and be added to the unauthorized users' (attackers) list. Hence, when there is a significant deviation even if the provided credentials are correct, the request will also be declined. In order to improve the security of the home network, updating any login credentials and unblocking mobile devices is achieved by the homeowner via a local channel with the home smart hub. Every time there is an attempt from unauthorized users (probably outsiders), the model will be updated with the new events as unauthorized user patterns. In this way, the proposed approach will be able to recognize legitimate and illegitimate users without the need to notify the homeowner, thus providing home users with increased convenience.

A. DESIGN GOALS

The objective of the proposed method is to produce an authentication and identification approach that builds a user profile based on previous access history in order to make the right decision at the login stage, and at subsequent access requests regarding legitimate user access. The main goals of the presented approach include: authenticating and identifying users with low FPRs, FNRs, and EERs; utilizing implicit features that can be extracted, without requiring user intervention in the identification process; ensuring that the utilized features are generalized, hence can be extracted from most mobile devices regardless of the operating systems on these devices; and protecting the privacy of the collected information on mobile devices during transmission and at the server end. However, there are a number of challenges that need to be overcome in order to achieve the presented goals. These challenges, which are considered in the design process, include: transforming the app access events in the form of observations that include timing transition information; building (training and testing) the model in a way that considers imbalances in the users' observations; utilizing a low number of events, hence reducing the time factor, in the authentication and identification process; and adapting the change in user patterns, including new added apps. By this solution, non-expert users will be able to adopt the proposed solution without the need to be knowledgeable about the technical and programmable issues related to the network.

B. ASSUMPTIONS

The presented approach in this paper is subject to the following assumptions:

- Registered users, who are trusted, are assigned access rights to home appliances through their mobile devices. After registration, there is a model building and training stage, during which authentication and identification will be provided by other means. The interactions performed beyond the enrollment stage as well as during the training and testing stages are related to the main users.
- The home network is protected against outsider unauthorized access, meaning that unregistered devices are unable to communicate with the home network without passing the registration stage.
- There is no mixed (shared) membership for mobile devices among registered users during the model training and testing stage. In addition, smart home services can be accessed and controlled by home members based on the assigned permission at the registration stage.
- All mobile devices/tablets are uniquely identified; the operating systems as well as installed apps, including the smart home user interface, are secure.
- This smart home server, which is a controller device, is assumed to be trusted and capable of securing a connection between home appliances and the smartphone device.

C. THREAT MODEL

Accessing the smart home network and controlling home appliances is mainly achieved through registered mobile devices by known users. However, access can be achieved by other users, who will be able to access the home network using registered users' mobile devices. Accordingly, there are security points where unauthorized access to the home networks could occur:

- The user is logged into the mobile device, but leaves the device unattended, yet unauthorized users insiders) have access to it. An insider, as mentioned in [37], can be a visitor or another registered user.
- The mobile device, on which user's device login credentials are stolen, compromised, or given, is lost or stolen by unknown users (outsider, stranger [6]), causing home devices to be vulnerable to unauthorized access and usage.

Thus, at any access request, the authentication process will result in three possibilities:

- ○ Registered user (main user) requests access from his/her registered mobile device;

- Registered user (insider) requests access from another registered mobile device;
- Unregistered user (outsider) requests access from a registered mobile device.

In general, a non-home member could be accidentally and unintentionally assigned access by a registered user to the home network. This also applies to the case where, for example, a home member gives his/her mobile device for a visitor to access the Internet, and this user consequently tries to access smart home devices without informing the main user.

D. WORKFLOW

The presented work in this paper utilizes the user behavior patterns on users' mobile devices as a second layer of authentication and identification. This approach is employed at the access request and during the access session to the home network. To the best of our knowledge, this is the first attempt to incorporate app access behavior into remotely accessing smart home networks. The architecture of the presented model, as shown in Figure 1, works, after the registration step, by first collecting user access logs, extracting features, and training user behavior during access to apps on mobile devices, then authenticating and identifying users based on the built behavior templates. The following steps show the workflow of the proposed method. After the enrollment stage, the model tracks and collects app access events adopting the event-driven mechanism. Utilizing the event-driven data collection approach minimizes the consumed power and collects only the information that will be employed to build user behavior. All information related to app access history is collected by an installed app on the mobile device and is then sent to the smart home server for the training, testing, authentication and identification phases. The collected logs at the data collection unit will then be sent in an encrypted form to the smart home server which, in turn, will decrypt the received information. After building and training the model, this unit will only send the events of the considered apps for user authentication and identification purposes.

1) APP CATEGORIES

In terms of running on mobile devices, there are two app categories: foreground apps and background apps. Foreground apps require continuous user interaction during the running time. However, background do not need continuous user interaction during the running time. The usage of foreground apps provides real interaction of users with their mobile devices, whereas background apps offer little or no information on user interaction with mobile devices. In addition, since app usage data will be already available as a result of users' usage on their mobile devices and tracking these apps involves minimal or no power consumption even when including the data transfer to the local home hub, real-time user authentication and identification can be achieved with high accuracy. Hence, different from previous related studies, we mainly focus on foreground apps. Additionally, when a foreground app goes into background mode, we neglect it and consider only the session time while the user is interacting with this app, thus presenting the user interaction behavior.

2) DATA COLLECTION

The data collection procedure runs on mobile devices and records the actions (events) whenever a foreground app runs. The app collection procedure occurs during access to apps via mobile network, WiFi networks or local apps that do not need access to the network in order to run. The two modes of data collection procedures are the training and testing mode and the authentication and identification mode.

- Data collection for model training and testing

In this stage, access records (events) on the user's mobile device are collected whenever the user interacts with foreground apps. The collected information includes the user ID (*u_id*), the app identifier (*app_id*), the app interaction timestamp (*app_st*), and the app end interaction timestamp (*app_ed*). The access session is defined as when the app is interactively accessed by the user in the foreground mode. However, when the app situation changes to background mode or if it is closed, this situation is considered as the end of the access session.

- Data collection for user authentication and identification

In this stage, only the information of the last n accessed apps will be collected prior to the home network access request and during access sessions. Hence, every time an app is accessed on the mobile device, the app session information is collected and saved in a first-in-first-out (FIFO) buffer with a limit of n accessed sessions. Thus, whenever a new app is called, the buffer is updated with the new app while the oldest is removed from the queue. Hence, the collected information in the queue includes: *u_id*, *app_id*, *app_st*, and *app_ed*. At this stage, previously accessed apps are also considered. For example, if a new app is launched, but is not the first in the queue (list), this app will not be considered in the access decision for the next request. However, the model will be updated and continuously trained until reaching a good accuracy considering the number of accesses prior to the app being considered for inclusion in the authentication and identification of decision making.

3) DATA PREPROCESSING AND FEATURE EXTRACTION

Features that can be utilized in modeling user behavior can be generally categorized as explicit or implicit. The former includes features that are directly reached while accessing the mobile device, including app name, location and timestamp. In contrast, the implicit features include information that can be derived from statistical operations during smartphone access, such as app usage sequence, distribution, category, and access duration. As reported in [38], implicit features are more effective in distinguishing the access behavior of users. Including more features will help to mitigate the problem of similarity in user behavior, such as having the same access pattern to specific apps. Therefore, the focus of this paper is on the implicit features. Thus, the collected information will subsequently be preprocessed and stored at the home hub.

Feature extraction is an important step where unique features are extracted from the collected information. As a consequence, at this stage, a suitable set of features will be extracted and prepared in order to enhance the classification process. The features that can be retrieved from app access logs on mobile devices include: *app_id; app_st; app_ed;* generated traffic (*app_traff*) while accessing this app; and *u_id*. In order to build a continuous authentication model, the literature presents approaches that use a specific app to differentiate between users, but our goal is to utilize features to continuously authenticate and identify users. However, there are factors that need to be considered in order to generate features that can enhance the classification process. The first factor is that the users' routines in accessing apps usually follow regular intervals, but sometimes deviate due to different circumstances. For example, a user may browse an app at the same time every day; however, due to a change in schedule, the app may be checked late. In this case, duration of access would be similar as it is a routine for the user, but the access time would shift in time slot. As a result, the app access start time might not always be consistent. Thus, the access duration should be given more attention. Hence, we divide the time of day into six time intervals: *inter_1* (>=00:00 & <07:00); *inter_2* (>=07:00 & <10:00); *inter_3* (>=10:00 & <12:00); *inter_4* (>=12:00 & <17:00); *inter_5* (>=17:00 & <21:00); and *inter_6* (>=21:00 & <00:00). In addition, the same might occur with days of the week. Therefore, we divide the days of the week into three weekday intervals: *w_inter_1* (beginning of the week); *w_inter_2* (end of the week); and *w_inter_3* (weekend). Secondly, the time between access sessions is considered to be an important feature, which we believe will enhance the usage behavior of users.

The transition between apps on a mobile device can be in two forms: transition between the same app, and the transition between all apps (the gap between consecutive app access sessions). In this work, we consider both as we include all accessed apps to model user behavior. Hence, the transition between apps is calculated prior to each app access inactivity time prior to the app access session. Thus, we consider two features, named inter-app access time (*inter_pi*), and intra-app access time (*intra_pi*). The first feature, the *inter_pi*, includes the interval between two consecutive accesses (a_i *and* a_{i-1}) to the same app on the same day. This interval is calculated as $b_i - a_i$ for all apps accessed on the same day. The second feature, the *intra_pi*, includes the interval between any two consecutive general accesses (b_i *and* a_i) to the next app on the same day. This interval is calculated as $b_i - a_i$ for all apps accessed on the same day. This feature is individually considered every day as user access behavior may change from day to day. However, there may be a long time gap between the last accessed app and the new access request when, for example, a user does not access apps or at the beginning of the day. This problem is solved by utilizing the time intervals during the same day. Hence, the time transition between intervals denotes the gap between these intervals. The long transition time that occurs in some cases is neglected in order to avoid unknown cases such as sleeping, traveling or being out of power. Both the *inter-pi* ($inter\_pi_i = app\_st(a_i) - app\_st(a_{i-1})$) access time and *intra-pi* ($intra\_pi_i = app\_st(b_i) - app\_st(a_i)$) access time are extracted, and the access events are updated with the new features.

The other important feature that needs to be considered is the sequence order of access to apps. The advantage of considering sequentially accessed apps is that there is no need for a sample time interval, meaning that we do not need to sample the tracked accessed apps for each specific period of time. Rather, the proposed approach requires sequentially accessed apps whenever an app is used, and this access is measured as event-driven access. When the access log is received from a user's mobile device, it is used to generate the required features (at the home central hub), including: session access time (*app_st*, and *app_et*); extracting days of the week from the timestamps (weekday and weekend (*week_day*)); day time (*day_time*); app daily usage sequence order (*app_sq*); app continuous sequence order (*app_cont_sq*); app category (*app_cat*); app access duration (*app_dur*); inter-app access time (*inter_pi*); intra-app access time (*intra_pi*); as well as inactivity time prior to the app access session (*pi*). Hence, the received access logs will be transferred to event information that includes the extracted features. The extracted features will then be stored in raw form in the database for training and testing processes. The number of the required usage sessions mainly depends on the user's interaction, which can be determined in a continuous manner during model training and testing. Including more features will help to mitigate the problem of similarity in user behavior, such as having the same access pattern to specific apps.

4) CLASSIFICATION STRATEGY

An appropriate classifier will be applied to events, with the prepared features from the previous step. In building the complete model, for providing authentication and identification, a binary classification strategy is used. However, many real-life classification scenarios, such as intrusion detection in networks, fraud detection, and health care diseases, have imbalance in the related data, in which the classes are not of the same distribution. There are different approaches to dealing with this problem. However, the type of data in the application should be taken into account when having imbalance in the data. Despite the availability of different techniques that deal with imbalanced data, the suggested solution might not be generalized to other types of applications. Moreover, the variance of the classes' distribution in the same dataset impacts the classification performance. To deal with the class imbalance, the up-sampling (over-sampling) technique is applied to balance the class distribution of the data samples during the training process. Furthermore, as we are targeting multi-user authentication, the *one-vs-rest* classification [39] will be applied for each class, with the result that each access event will be classified as being related or not to one of the enrolled users. Hence, each classifier will be trained with the first class (C_1) as the targeted class (legitimate user), and the second class (C_2) as the illegitimate user. Consequently, the classifier classifies each single event, producing a probability of the related class of this event. Training and testing methodology on each user's information in an incremental usage basis is applied, in which training the model will be applied within a specific time interval and testing the model will be applied on unseen data. Hence, each user's information template is created by training the classifier on the given information of this user as legitimate while considering the rest of the users as illegitimate. Each classifier is trained on the data of a specific user; thus we need to construct N binary classification models ($CM_1, CM_2, CM_3, \ldots CM_N$) based on the number of users.

As a result, the authentication process requires only the computation of one classification model (CM_i) on the information received from the registered device from which the request is issued, and the user claiming to be legitimate. Therefore, each classification model enables the authentication of the related (assigned) user (u_i). In contrast, the identification process requires the computation of the *N-1* classification models to classify the received sample information to one of the previously trained user patterns. We chose to utilize the random forest (*RF*) classifier as it is widely used in many applications such as banking, medicine, the stock market and e-commerce. Furthermore, it has given a higher accuracy in our previous work. Hence, we select the parameter values that minimalize the FPR and FNR as much as possible for all users. In addition, it has evidenced high accuracy in previous studies.

Algorithm 1 App-Based User Authentication and Identification Models Building

```
INPUT: Dataset $D$
OUTPUT: user's classification model $CM_i$, $\forall$ is the extracted and assigned threshold
procedure()
input ← $D$
read features ← $\{f_{x1}, \ldots., f_{xn}\}$: features of access
$App_{ev}^{fgr}$ ← select only foreground-based events
    for user ($u_i$) ← 1 to $N$ do ▷ where $N$ is the number of users
        do {
            $D_i^{sample}$ ← 1 to $n$ do ▷ where $n$ is the number of samples
            extract new features ← $\{f_1, \ldots., f_z\}$ ▷ where $z$ is the number of features to be extracted
            split $D$ into $D_i^{train}$ and $D_i^{test}$
              randomly split $D_i^{train}$ into $k$ subsets $\{D_1, \ldots D_k\}$ ▷ where $k$ is number of folds
              up-sample the $D_{min}^{train}$ of the minority class ($D_{min}^{up\_samp}$)
              build the model ($M_i$) using both $D_{min}^{up\_samp}$ and majority class $D_{maj}$
              test the $M_i$ on $D_i^{test}$
              calculate $FPR$, $FNR$ and $EER$ ▷ where FPR is false positive rate, FNR is the false negative rate, and EER is the equal error rate.
        } while ($FPR$, $FNR$, and $EER<5\%$)
        $\forall$ ← set threshold, number of access events per app
        set the threshold for app access ← $\forall$
        $CM_i$ ← launch
    end for
end procedure
```

5) USER AUTHENTICATION AND IDENTIFICATION

The objective of our proposed method is to build an authentication and identification model of legitimate user access patterns. Algorithm 1 shows the process of building the app usage pattern-based user authentication model. This step involves a training stage in which the user data is collected, and the final classification model is created. After the initial access, the model starts to monitor user behavior while accessing home appliances. The access logs, which are translated events with extracted appropriate features, are then classified as for an enrolled user or not. Two important aspects should be considered while tracking user access to smart home networks, namely user authentication and identification. User authentication is defined as the mechanism that determines whether the provided pattern that is coming from a registered device of the current user belongs to a legitimate, registered user. In contrast, user identification can be defined as the mechanism that determines whether the provided, collected pattern belongs to one of the previously registered, known users. For user authentication, only one classifier will be run, whereas for identification, *N-1* classifiers will be run at the same time and the decision will be based on the output of these classifiers, as shown in Figure 2. Hence, as a first step, the proposed approach performs user authentication on the received access logs, and when this pattern does not belong to the main owner of the end-device, it performs the identification procedure in order to detect whether this user is one of the home members. If the user is classified as one of the home residents, access permission will be given based on the setting established at the registration stage. In most cases, the appearance of many unknown apps during the authentication process will indicate that it is not being accessed by the legitimate user, but from another user, either an insider or outsider.

In general, unknown apps could appear in two cases: apps that are not part of the training set while training the model and apps that are newly launched by the user. The first case does not have a significant effect on the model because we utilize a *k-fold* based training, hence eliminating the chance of not including all used apps in the training stage. For the second case, when an event contains a new app, the decision unit

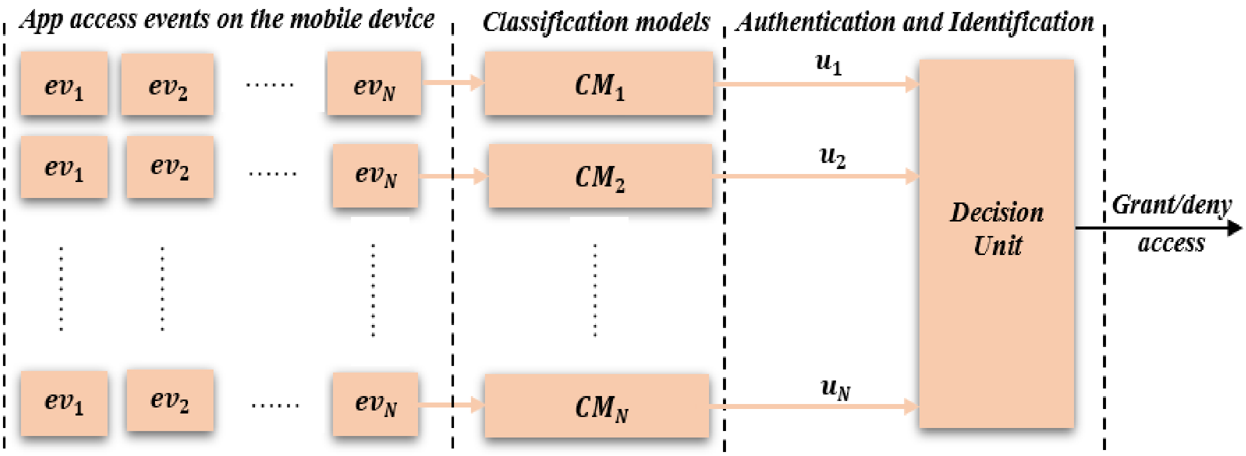

FIGURE 2. User authentication and identification procedure.

handles it as follows: if this app is the last in the last sequence of accessed apps, it invokes the user for a *sec_fa*, and when it is provided by the user, the user will be authenticated and the model will be trained with this new app until reaching a specific number of interactions to this app (*app_cont_sq*). If it is not the last app in the last sequence of the accessed apps, the user can be authenticated if the last received app events meet the set criteria at the decision unit according to the classification probability of the rest of the apps in the sequence.

6) DECISION UNIT

Classifying each access event received from the user's mobile device may increase FPRs. Thus, to eliminate this issue, a number of events (window size [N]) should be considered in determining access decisions. Consequently, we consider applying window size events (number of events) to determine the access decision. Therefore, at the decision unit, the access decision (D_i) of the new request is made based on the classified events of the last two apps accessed, based on the formula 1, immediately before the access request sent from the user.

$$D_\mathrm{i}=\begin{cases} if\ (a_{l-1}\ and\ a_l)\in u_i,\ Permit\ access\\ if\ (a_{l-1}\in u_i)\ and\ (a_l\notin u_i),\ Deny\ access\\ if\ (a_{l-1}\notin u_i)\ and\ (a_l\in u_i),\ Deny\ access\\ if\ (a_{l-1}\notin a_l)\ and\ (a_l\notin u_i),\ Deny\ access\end{cases}\quad(1)$$

At this unit, the decision (D_i) will be made based on the last accessed events (a_{l-1} *and* a_l). If the last accessed events are identified to the current user, the next request will be accepted; otherwise it will be denied, and the user will be requested to undergo a second-factor authentication in order to prove identity. For the identification, the decision will be made based on the majority of the rest of the classification models, based on the formula 2. For example, if the access events were received from a user's (u_i) device and the related classification model classifies such access as not for this user, then these events will be passed along with the access request to the identification decision unit to check if they belong to one of the registered (known) users.

$$D_\mathrm{i}=\begin{cases} if\ (a_{l-1}\ and\ a_l)\in u_{i+1},\ Permit\ access\ to\ u_{i+1}\\ if\ (a_{l-1}\ and\ a_l)\in u_{i+2},\ Permit\ access\ to\ u_{i+2}\\ if\ (a_{l-1}\ and\ a_l)\in u_{i+3},\ Permit\ access\ to\ u_{i+3}\\ .\\ .\\ .\\ if\ (a_{l-1}\ and\ a_l)\in u_n,\ Permit\ access\ to\ u_n\end{cases}\quad(2)$$

If it is recognized as one of the registered home members, the access can be granted based on the permission assigned to this user at the registration stage based on the formula 3. The strategy here comprises the computation of N−1 classification models and the decision (D_i) will then be made based on most of the highest probability score from the models.

$$D^{k_{us}}=\sum\nolimits_{=(j\neq i)}^{n} ide\_Fun(CM_j\,(k_j))\quad(3)$$

where u is the unknown received sample, D is the decision score, n the number of classification models, K is the collection of the events that need to be fed to the classification model CM_i, and *ide_Fun* is the identification function.

The result of the decision will be either classification as one of the known users u_i or unknown user u_{un}. In the third case, when the received sample is not identified to any of the trained users' templates, it is considered to be an unknown (outsider) user and the access request will be declined. In addition, the event misclassification is considered as not classified to the registered user and a second factor authentication (*sec_fa*) will be requested from the users. In the case of the second factor authentication is provided by the user, the model will be trained on this event. Algorithm 2 shows the process of launching the app usage pattern-based user identification and authentication model.

Algorithm 2 App-Based User Identification and Authentication Model Launching

```
INPUT: $ev_{i-n}, \ldots, ev_{i-2}, ev_{i-1}$ are the last app events, $a\_req$ is the access request, ∀ is the extracted and assigned threshold from the model building step, $sec\_fa$ is the second factor authentication
OUTPUT: access request decision (grant/deny) to user ($\boldsymbol{u_i}$, $\mathbf{u_j}$)
procedure ()
receive ← $\{ev_{i-n}, \ldots, ev_{i-2}, ev_{i-1}, a\_req\}$
$\{f_1, \ldots., f_z\}$ ← Generate features set
    for user ($\boldsymbol{u_i}$) ← 1 to $N$ do ▷ where $N$ is the number of users
        while ($a\_req \neq 0$) do
            if threshold >= ∀ then
                $CM_i \leftarrow \{ev_{i-n}, \ldots, ev_{i-2}, ev_{i-1}$
                if $\boldsymbol{u_i} \leftarrow \{ev_{i-n}, \ldots, ev_{i-2}, ev_{i-1}\}$ then
                    access ← grant $\boldsymbol{u_i}$
                end if
            else
                $(CM_{j\neq i}, CM_{i+1}, \ldots CM_N) \leftarrow \{ev_{i-n}, \ldots, ev_{i-2}, ev_{i-1}\}$
                if $\boldsymbol{u_j} \leftarrow CM_{j\neq i}$ then ▷ where $CM_{j\neq i}$ is the classification model of another registered user $\boldsymbol{u_j}$
                    access ← grant $\boldsymbol{u_j}$
                else
                    request ← $sec\_fa$
                    if correct ← $sec\_fa$ then
                        $\boldsymbol{D_i^{train}}$ ← update ($ev_i$) ▷ update the model with the new utilized app
                        access ← grant $\boldsymbol{u_i}$
                    else
                        access ← deny $\boldsymbol{u_i}$
                    end if
                end if
            end if
        end while
    end for
end procedure
```

IV. EXPERIMENTAL EVALUATION AND RESULTS

To evaluate the performance of the presented method, the datasets UbiqLog4UCI [40] and LiveLab [41] collected from real users is utilized, and the identification performance is considered as the accuracy metric when classifying an access session to one of the enrolled users. To make sure that the presented model is not classification algorithm specific, three classification algorithms are used in the training stage. The selected classifiers in this research, which are mostly used in the literature, such as in [42]–[44], including three different classification methodologies. The first classifier is the *RF* classifier, which fits a number of decision tree classifiers on various subsamples of instances (events) and utilizes the average in order to improve accuracy and eliminate overfitting. The second classifier is the gradient boosting classifier that offers several hyperparameter tuning options that provide the function with a very flexible fit. The third classifier used in our evaluation is the *KNN*, which applies the k-nearest neighbors' vote. In addition, the training data has to be saved at the classification time. Even though other classifiers, such as the *SVM*, have been used in the literature, it requires more computation time and produces less accuracy. These three classifiers are then applied, and as a first step, we compared common classification approaches on the training set in terms of FPRs, FNRs and EER. Then, the algorithm that provides the highest recall, precision and high F-measure is implemented, which is the RF classifier.

- UbiqLog4UCI Dataset

This dataset is collected from 35 users in a period of approximately three months. The collected data summary of the *UbiqLog4UCI* dataset is shown in Table 2. The collection procedure includes a background application that collected accessed app events with the timestamp for the time of access. The time is represented in the form of a Unix timestamp. Thus, the collected logs are represented in the form of tuples, including *app_id*, *u_id*, *app_st*, and *app_ed*.

TABLE 2. The utilized datasets.

Dataset	Number of users	Number of events	Number of unique apps	Period (days)
UbiqLog4UCI	35	545,230	1312	90
LiveLab	34	1,041,468	1992	365

TABLE 3. The most accessed apps in the *UbiqLog4UCI* dataset.

App#	App Name	# of events
1	com.android.nfc	28008
2	com.sec.android.provider.logsprovider	24716
3	com.android.settings	22090
4	com.samsung.android.providers.context	19183
5	com.viber.voip	17988
6	com.sec.android.app.twdvfs	17399
7	com.broadcom.bt.app.system	17316
8	com.android.mms	14897
9	com.sec.android.app.launcher	13833
10	com.sec.pcw.device	12734

Although some users have collected data for periods of less than three months, most users have data for three months or more. As not all users have app access information that is adequate for analysis, the information from only 30 users is utilized in the evaluation in this paper. Table 3 shows the statistics for the 10 most used apps, including the number of apps and number of events per app, for the selected users during a period of three months.

- LiveLab Dataset

The second dataset is the *LiveLab* project dataset collected from 34 users in a period of approximately one year.

The collected data summary of the *LiveLab* dataset is shown in Table 2. This dataset includes the data of 24 university students collected for one year, and the data of 10 community college students collected for six months.

The collected information includes categories such as a list of all installed apps among all users, apps run by users, phone calls made/received by users, accelerometer readings, and time that the logger was running. However, the utilized information in this paper is related to app usage and the registered time when apps were accessed. The collection procedure includes the background application that collected accessed app events with the timestamp for the time being accessed. Thus, the collected logs are represented in the form of tuples: *app_id*, *u_id*, *app_st*, and *app_ed*. The time is represented in the form of a Unix timestamp. Table 4 shows the statistics of the 10 most used apps, including the number of apps and the number of events per app, for the selected user during the one-year period.

TABLE 4. The most accessed apps in the *LiveLab* dataset.

App#	App Name	# of events
1	SpringBoard	457867
2	com.apple.MobileSMS	173462
3	com.apple.mobilephone	79016
4	com.apple.mobilemail	46114
5	com.facebook.Facebook	42545
6	com.apple.mobilesafari	25506
7	com.apple.mobileipod-MediaPlayer	17379
8	com.apple.mobiletimer	13557
9	com.apple.Preferences	10803
10	com.apple.Maps	9164

A. STATISTICAL ANALYSIS OF THE EXTRACTED FEATURES

The imbalance in the class representations is important to consider during the training and testing process of the classes. As seen in Figures 3 and 4, the imbalance in the classes is clear in both datasets. Consequently, considering the variance in the classes' representation is necessary. In addition, in Table 3, from the *UbiqLog4UCI* dataset, the 1st, 2nd and 3rd apps are in the top list because they are the most used by users. In Table 4, from the *LiveLab* dataset, the apps from the 1st to the 6th are the most used by users. After the mentioned features at the feature extraction subsection are extracted, user usage patterns can be learned, and a template of this pattern can be built and then utilized for the authentication process. Selecting the most effective features is an important process because it will strengthen the pattern template of users and subsequently affect the performance of the classification process. Before evaluating the proposed approach for user authentication, we study the feasibility of utilizing the extracted features for differentiating users. For testing the extracted features, many feature selection strategies have been considered and applied including forward selection, backward elimination, and stepwise selection. However, the statistical analysis step is an extract step to test our presented model. Hence, in order to test similarities between user patterns and to examine the effect of the extracted features on differentiating between users, a comparison regarding the standard deviation (*std*) and the mean is considered, as shown in Table 5 for the *UbiqLog4UCI* dataset and in Table 6 for the *LiveLab* dataset. The extracted features, in addition to the day of the year (*year_d*) and *week_day (weak_d)*, *week_day_interval* (*w_inter_i*), hour of the day (*h_inter_i*), day interval time (*d_inter_i*) are *app_sq*, *app_cont_sq*, *app_cat*, *app_dur*, *inter_pi* and *intra_pi*.

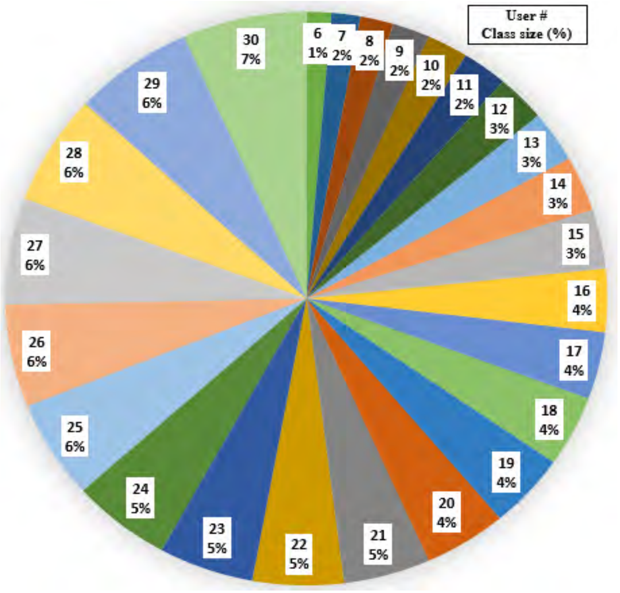

FIGURE 3. User observation representation in the *UbiqLog4UCI* dataset.

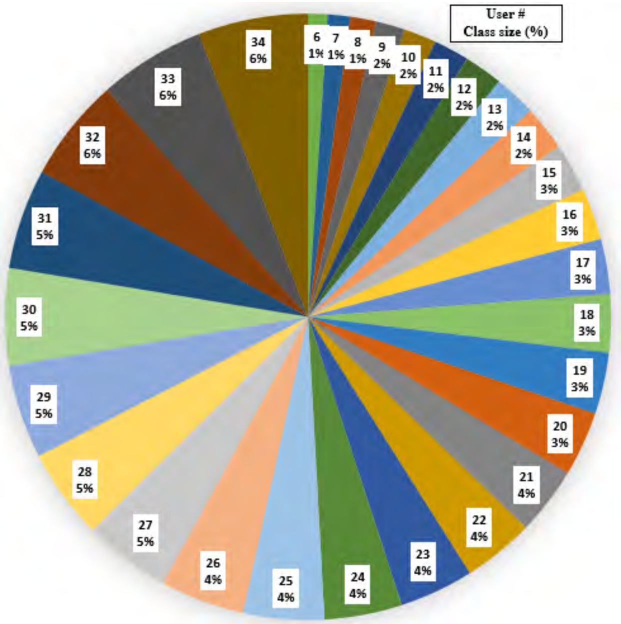

FIGURE 4. User observation representation in the *LiveLab* dataset.

For the analysis, we selected the most similar users from both datasets, according to similarity of access patterns. Due to limited space, we include only the features *app_dur,*

TABLE 5. Statistical analysis of the extracted features in the *UbiqLog4UCI* dataset.

	app_dur		*app_cont_sq*		*app_sq*		*intra_pi*	
User#	*mean*	*std*	*mean*	*std*	*Mean*	*std*	*mean*	*std*
1	17.94	10.02	9.13	5.66	2.05	1.19	90.00	30.78
2	18.91	90.00	10.00	10.00	25.61	31.14	29.22	69.25
3	42.90	10.70	11.59	11.07	15.41	16.40	10.06	10.14
6	52.28	10.85	12.77	15.11	13.01	16.08	10.36	10.29
18	26.22	10.64	28.38	43.78	7.52	10.61	10.22	10.33
19	26.72	10.41	47.97	59.14	38.06	29.40	11.13	19.21
20	42.67	10.60	37.49	50.55	21.68	24.02	36.55	54.57
22	14.70	10.01	34.75	52.81	10.22	10.38	10.73	10.48
27	23.29	10.36	55.40	90.00	23.94	24.29	11.54	14.87
28	60.03	10.89	57.96	55.03	51.38	48.76	11.08	14.12
30	24.65	11.12	35.76	67.07	13.77	13.93	46.24	90.00

TABLE 6. Statistical analysis of the extracted features in the *LiveLab* dataset.

	app_dur		*app_cont_sq*		*app_sq*		*intra_pi*	
User#	*mean*	*std*	*mean*	*std*	*mean*	*std*	*mean*	*std*
2	58.91	88.88	48.92	45.60	13.89	17.37	26.75	57.38
3	26.35	40.93	45.15	43.77	18.20	22.25	15.76	36.32
5	32.93	51.32	37.61	49.28	13.03	16.05	17.75	42.61
6	50.03	66.64	55.64	63.42	14.19	20.25	13.19	11.53
12	35.76	72.17	73.84	87.49	28.23	43.38	23.46	53.68
14	18.90	18.00	77.46	73.50	5.29	5.60	90.00	90.00
21	55.50	67.15	12.11	10.00	17.23	25.22	20.41	48.09
31	40.51	66.08	33.36	26.48	17.67	24.77	14.97	25.07
32	71.16	82.69	29.73	27.69	9.59	12.16	24.07	34.42
33	43.37	49.30	29.59	21.94	27.52	37.49	11.32	21.39

app_cont_sq, app_sq and intra_pi for the analysis, as shown in Tables 5 and 6. For the *UbiqLog4UCI* dataset, within the same features, the *app_dur*, we can see from Table 5 that the mean for users (1 and 2), (3 and 20) and (18 and 19) is similar. However, the std is different for (1 and 2) but still similar for users (3 and 20) and (18 and 19). There is a close similarity between the mean of the *app_cont_sq* feature for users (22 and 30) and (27 and 28) while the std is different for the same users. In addition, in the feature *intra_pi*, there is a close similarity between users (3, 6, 18 and 22) and (27 and 28) in both the mean and the *std*, which makes it difficult to differentiate between these users using these features. However, the similarities become less for other users and in turn will enhance the performance of the classification.

For the *LiveLab* dataset, within the same features, the *app_dur* of apps, we can see from Table 6 that the mean is similar for users (5 and 12) but the std between these users is different, while it is similar for users (6 and 31). There is a close similarity between the mean of the *app_cont_sq* feature for users (32 and 33) but the *std* is different for these users. However, there is a similarity for the *std* between users (31 and 32). In addition, in the feature *app_sq,* there is a close similarity between users (2, 5, and 6) and (21 and 31) in the mean, and similarity between users (2 and 5) and (21 and 31) for the *std*. There is a close similarity between the mean of the *intra_pi* feature for users (3 and 31) but the *std* is different for these users. However, the similarities become less for other users and, in turn, will enhance the performance of the classification.

B. EVALUATION RESULTS

For the evaluation, many evaluation metrics can be utilized to evaluate the presented approach in this paper. An important issue is that the classes are not of the same distribution, and using other evaluation metrics such as accuracy is not useful with imbalanced data. However, our primary focus is to decrease the FPRs and FNRs, which will reflect the real performance of each classified access event, and these metrics are widely utilized in the research area. Hence, our main focus is to decrease the FPRs and FNRs as far as possible, which can be calculated as follows:

$$\textit{True Positive Rate}\,(TPR) = \frac{TP}{TP+FN} \tag{4}$$

$$\textit{True Negative Rate}\,(TNR) = \frac{TN}{TN+FP} \tag{5}$$

$$\textit{False Positive Rate}\,(FPR) = \frac{FP}{FP+TN} \tag{6}$$

$$\textit{False Negative Rate}\,(FNR) = \frac{FN}{FN+TP} \tag{7}$$

After building the normal user behavior model, the second step is to test the model to also authenticate users against anomalies, i.e., unknown users. However, it is impossible to have unknown users' data for training the model. Thus, in order to test the proposed model in differentiating users, the *one-vs-all* method is applied. The *one-vs-all* classification approach means that the target user will be labeled as normal whereas the rest of the users are labeled as anomalies. As seen from Figures 5 and 6, the number of apps and the interactions with these apps change during the weeks, which means that the users' access patterns are not consistent over a long time period. Although the *LiveLab* dataset has a period of one year, we selected 12 weeks from both datasets, for reasons of consistency.

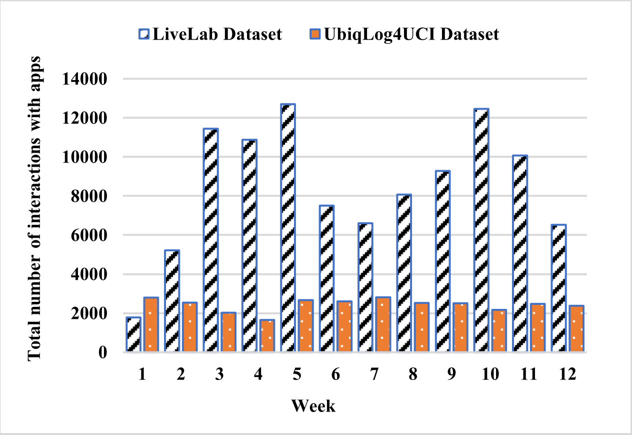

FIGURE 5. Number of interactions with apps per week for 12 weeks.

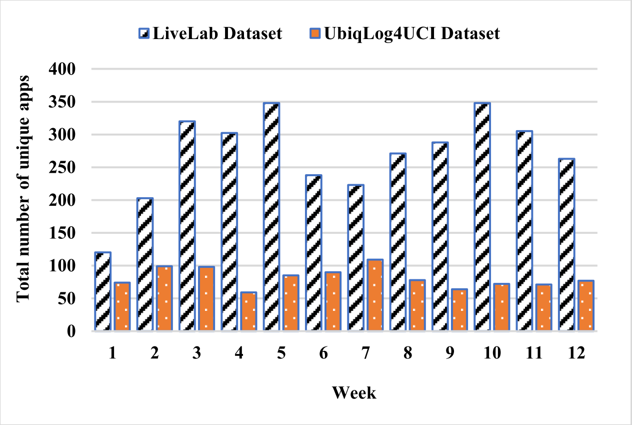

FIGURE 6. Number of utilized apps per week for 12 weeks.

Another important feature that we considered during feature extraction is the *intra_pi*. Figures 7 and 8 show the average *inter_pi* between apps for a 12 week period for five selected users from both datasets. As seen from Figures 7 and 8, in addition to the difference in the average access time to apps, it is noticeable that not all users have continuous access to apps during the full time period. This

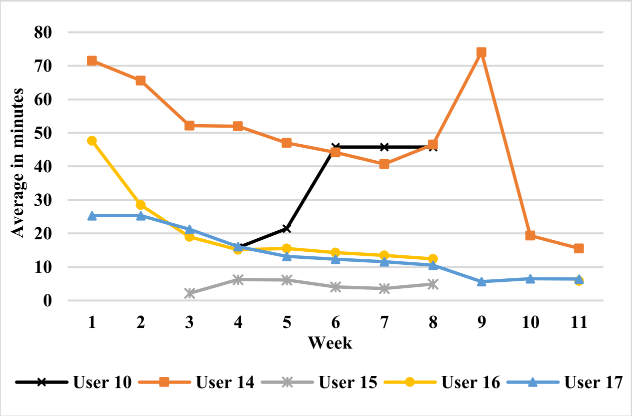

FIGURE 7. The average *inter_pi* for selected users for the *UbiqLog4UCI* dataset.

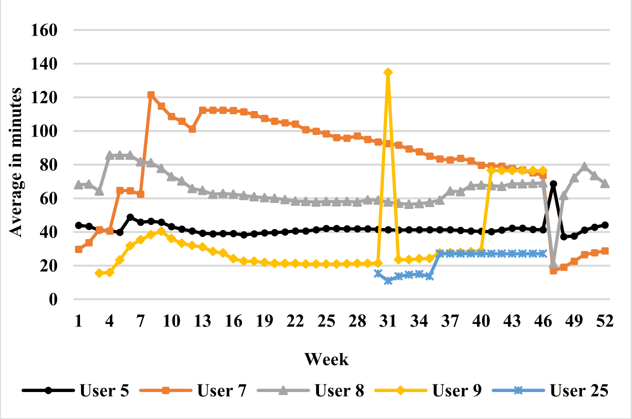

FIGURE 8. The average *inter_pi* for selected users for the *LiveLab* dataset.

might affect the classification procedure and increase the FPRs and FNRs.

To solve this issue, we extracted a new feature, the *app_cont_sq*, for each app. Therefore, the new added apps, either for a new user or a previously known user, will not be included by the model until reaching a specific access sequence threshold, the value of *app_cont_sq*. As the target is to discover any access anomaly in the network, the number of required events by the decision unit should be as small as possible. This will also reduce the processing time for both user authentication and identification. To find the best set of access sequence of events, we tested the datasets, and the results are shown in Figures 9 and 10. As seen from Figures 9 and 10, the FPRs, FNRs and EERs are improved as the number of *app_cont_sq* of apps increases. Therefore, to reduce the FPRs, FNRs and EERs, the presented approach in this paper requires a specific number of *app_cont_sq* to each app to be considered in the access decision in addition to requiring a specific number of last events to make the decisions at the time of the request. In other words, the app will be considered only when reaching a specific number of interactions, and the access decision will be made based on the number of last two events and their classification by the model. For example, if the model receives an event and classifies

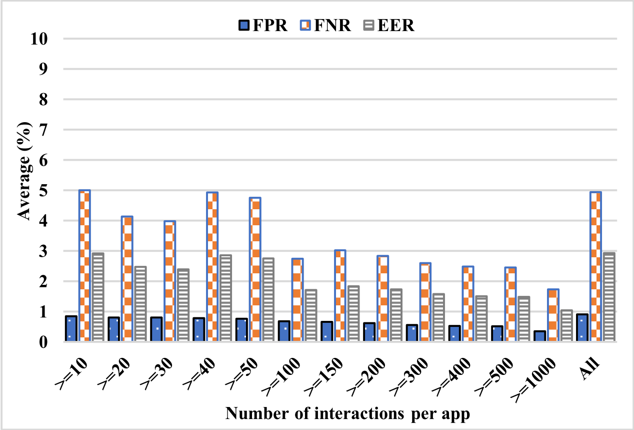

FIGURE 9. Model performance based on the number of interactions with apps for the *UbiqLog4UCI* dataset.

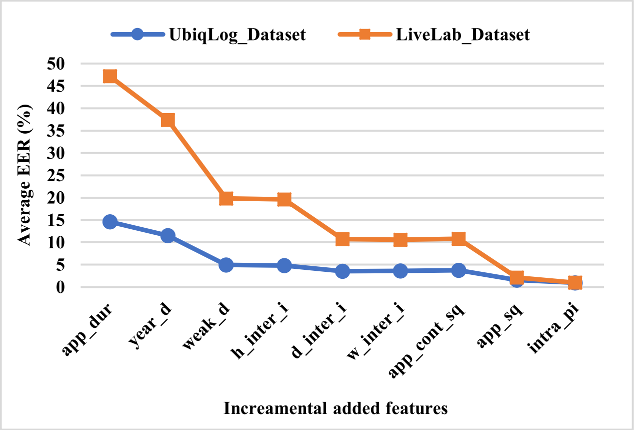

FIGURE 11. Performance evaluation based on the incremental addition of extracted features for both datasets.

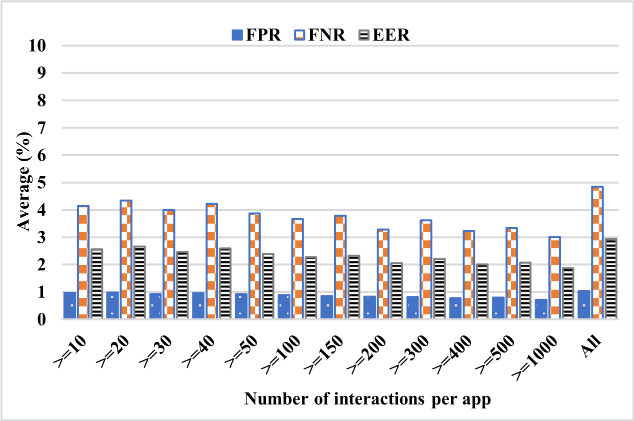

FIGURE 10. Model performance based on the number of interactions with apps for the *LiveLab* dataset.

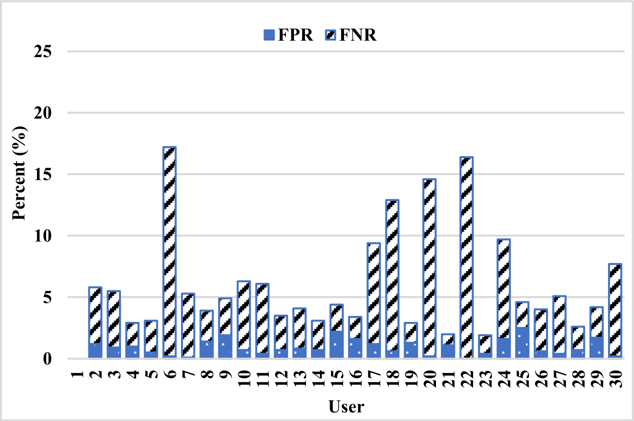

FIGURE 12. Model performance on unseen data for the *UbiqLog4UCI* dataset.

it as an anomaly or, in other words, not from the registered user, it classifies the previous events and based on a pre-set rule, the decision is made. In the next evaluation, features are added incrementally, and the model performance is shown in Figure 11. From the evaluation, it is clear that the added extracted features lead to improve the performance of the presented model. In addition, All the extracted features have an improvement on the performance of the model. Although that for the first dataset, it appears that the performance is improved slowly as compared to the second dataset, this issue because of the short-term data of users. However, for the second dataset, the effect of the added features is higher than in the first dataset. This test is applied with app interactions of fifty or greater, and the presented results indicate that the EER is high at the start when including the first feature for both datasets.

For the second dataset, the percentage of first measured EER is high as compared with the first dataset because of the longer time period of the collected data. However, as adding the other extracted features, EER decreases, and the best result is achieved when all features are included. Classifiers can then be trained on data from the owner and others, without assuming known data from the attacker. The methodology used in this paper is based on dividing the data into three parts: training set to train the model, testing set to tune the hyperparameters of the algorithm, and the validation set (unseen data), to validate the model. Figures 12 and 13 show the evaluation results of this approach and, from the results, it is clear that the model achieves good accuracy with low FPR and FNR.

The following evaluation is based on the number of users involved. The average EER is shown in Figures 14 and 15. It can be observed from the results that there is little change in the average EER when the number of users is increased. The performance of the presented approach consistently remains below 3.13%; however, as the number of users increases, the performance slightly decreases. This decrease is mainly a result of similarity in the users' access sessions, as usage may change and similarity among users may be present. Ultimately, the results indicate that the model produces a low EER even when the number of users increases.

For the last evaluation step, we simulate access requests on the unseen data, 30% of the dataset, and the proposed approach is tested based on Equation 1, while the results are

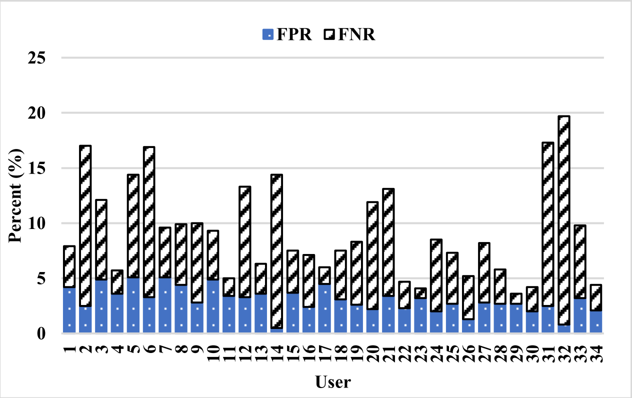

FIGURE 13. Model performance on unseen data for the *LiveLab* dataset.

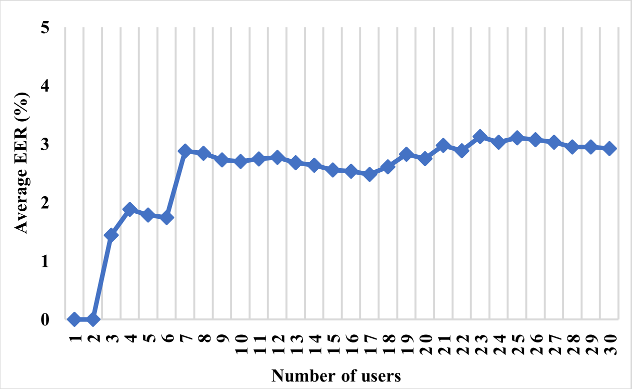

FIGURE 14. Model performance based on the number of enrolled users for the UbiqLog4UCI dataset.

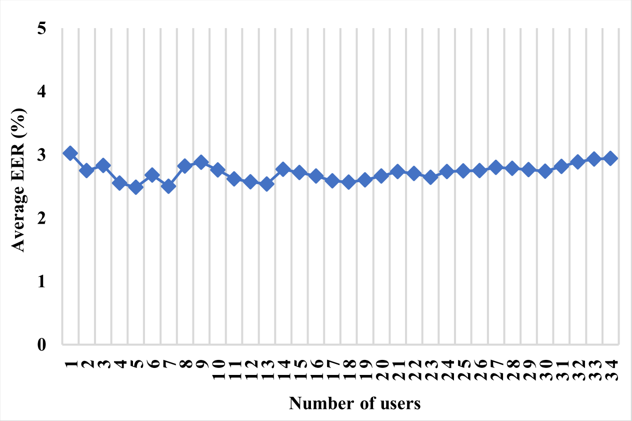

FIGURE 15. Model performance based on the number of enrolled users for the LiveLab dataset.

shown in Figures 16 and 17. From these figures, we can see that the minimum percent of average access decisions made is 95.20 % in the *UbiqLog4UCI* dataset for user 25 while the maximum percent of average access decisions made is 99.98 for user 1. The low 95.20 % for user 1 is because of the similarity with user 22. For the *LiveLab* dataset, we can see that the minimum percent of average access decisions made is 91.53 % in the *LiveLab* dataset for user 3 while the maximum percent of average access decisions made is 99.62 for user 14.

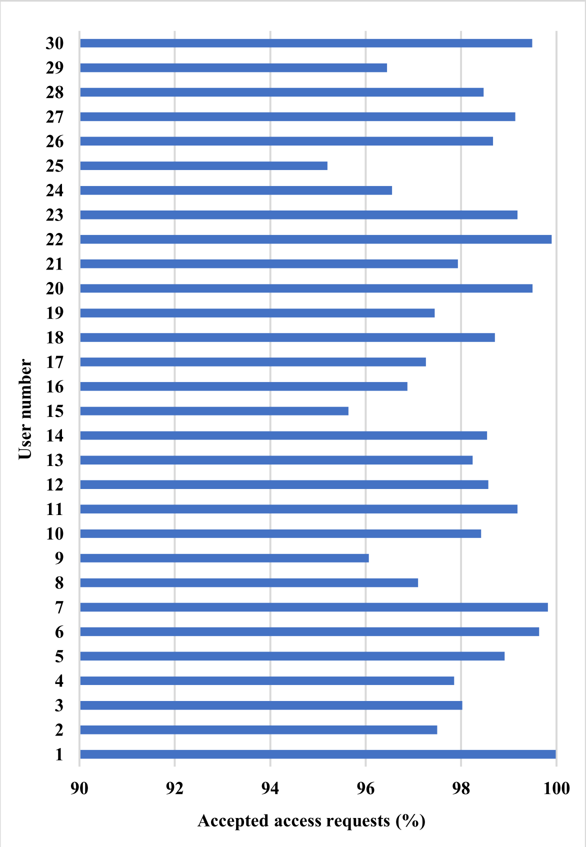

FIGURE 16. Model performance based on simulated access requests for the *UbiqLog4UCI* dataset.

The low 91.53 % for user 1 is because of the similarity with users 5 and 6, as discussed in the statistical analysis of the generated features shown in Tables 5 and 6.

C. DISCUSSION

In this paper, we provided a user authentication and identification approach for smart home networks based on user access behavior with apps on mobile devices. However, user behavior in accessing apps (in terms of access time and continuity accessing the same apps) might change over time, as seen in Figures 7 and 8. In other words, user access patterns may change over time and these changes, such as adding new apps or stopping the use of others, should be considered. Consequently, we overcome this issue by extracting new multi-instance features, including *app_cont_sq*, *app_sq*, *inter_pi*, and *intra_pi*, as discussed in the subsection on statistical analysis of the extracted features.

Thus, by including the *app_cont_sq* feature of newly launched apps, we guarantee that the apps will not be considered in the authentication and identification process until having enough training data. Furthermore, utilizing multi-events (last accessed app patterns) and the *intra_pi* feature in the access decision will increase security by reducing the chance of access by an illegitimate user. Additionally, as the identification takes place, other registered users will be allowed access from registered devices, which increases the usability of the proposed method.

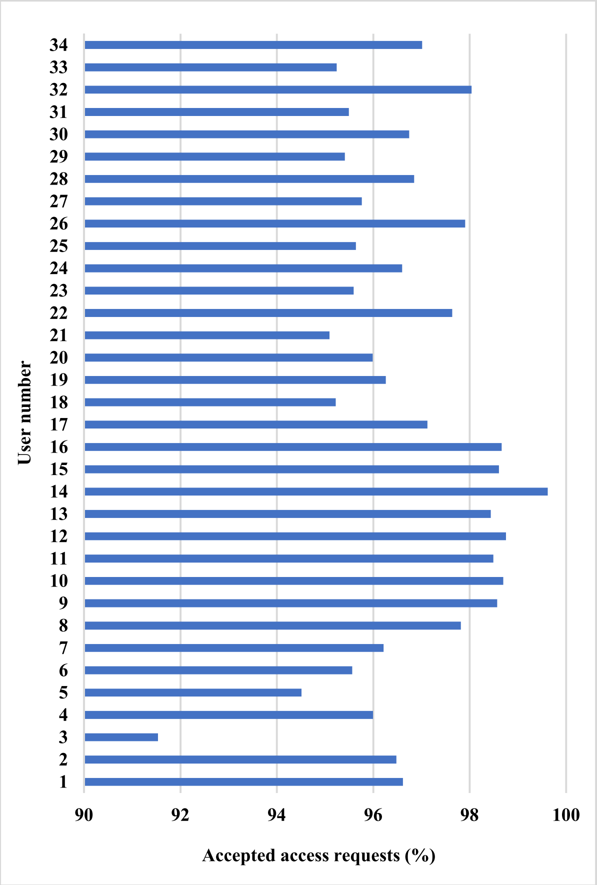

FIGURE 17. Model performance based on simulated access requests for the LiveLab dataset.

Although we considered traffic generated during app access sessions in a previous work [21], traffic generated information is not considered in this work due to the nature of the utilized datasets. We believe that the availability of this feature will considerably improve differentiation between users and eliminate similarities. Additionally, most datasets utilized in the previous studies cited in this paper are not publicly available or do not include a sufficient number of apps required to test our model. Hence, in our research we have utilized two publicly available datasets (UbiqLog, from the University of California-Irvine, and LiveLab, from Rice University). These datasets provide smartphone app logging that allowed for extraction of features needed to evaluate our model in this paper. Our work represents the first utilization of these datasets for user authentication and, hence, will provide a base for future user authentication model development and comparison. Although other related work studies have utilized different datasets, we have compared the results of our work against the above studies in terms of FPR, FNR, and EER for performance evaluation.

V. CONCLUSION AND FUTURE WORK

This paper presented a user authentication and identification model that verifies user access requests to smart home server, which improves security when a home network is remotely accessed. The objective of the proposed approach is to increase security in addition to delivering usability and protecting user information. This method does not require specific action from the user in order to be continuously authenticated and identified, but it is based on regular actions while accessing apps. Privacy is also considered in the proposed solution as no user data is kept on the smartphone and all the data, as well as the built models, are in the smart home server. Hence, in the case of the smartphone being lost or stolen, no user data will be accessed. In addition, as the presented method is performed in the background, based on the user's general access routine, it is difficult for a shoulder-surfing attack to continuously perform legitimate user interactions during different sessions.

In order to improve the efficiency of the proposed approach, we utilize only minimal features. In addition, the model evaluation, which is performed on two datasets based on common evaluation metrics, determines that it provides high accuracy in terms of low FPRs, FNRs, and EERs. Overall, the results obtained in this paper, in addition to the statistical analysis of the extracted features, show that the adoption of decision-making based on last app events leads to good accuracy. The results also show that considering the *inter_pi* and *intra_pi* features reduces the chance of the network being accessed by unauthorized users, therefore, increasing the level of security.

For future work, we plan to evaluate our model on a larger number of users with the inclusion of traffic generated information. We also suggest that developers provide mobile devices with anonymous app access data to be used for model training against the main user during the model building. In order to be more accurate, an alternative approach would be to download anonymous app access data.

ACKNOWLEDGMENT

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] H. Thapliyal, R. Kumar Nath, and S. P. Mohanty, "Smart home environment for mild cognitive impairment population: Solutions to improve care and quality of life," *IEEE Consum. Electron. Mag.*, vol. 7, no. 1, pp. 68–76, Jan. 2018.

[2] E. Fernandes, J. Jung, and A. Prakash, "Security analysis of emerging smart home applications," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 636–654.

[3] S. Faisal, N. Anani, J. Leiper, and M. Gupta, "The application of everything: Canada's apps economy value chain," in *Proc. Inf. Commun. Technol. Council (ICTC)*, Canada, 2014.

[4] A. Sharma and S. K. Sahay, "Group-wise classification approach to improve Android malicious apps detection accuracy," 2019, *arXiv:1904.02122*. [Online]. Available: http://arxiv.org/abs/1904.02122

[5] F. Li, N. Clarke, M. Papadaki, and P. Dowland, "Behaviour profiling for transparent authentication for mobile devices," in *Proc. Eur. Conf. Cyber Warfare Secur., Academic Conf. Int. Ltd.*, 2011, pp. 307–315.

[6] E. Shi, Y. Niu, M. Jakobsson, and R. Chow, *Implicit Authentication Through Learning User Behavior*. Berlin, Germany: Springer, 2011, pp. 99–113.

[7] L. Fridman, S. Weber, R. Greenstadt, and M. Kam, "Active authentication on mobile devices via stylometry, application usage, Web browsing, and GPS location," *IEEE Syst. J.*, vol. 11, no. 2, pp. 513–521, Jun. 2017.

[8] D. Damopoulos, S. A. Menesidou, G. Kambourakis, M. Papadaki, N. Clarke, and S. Gritzalis, "Evaluation of anomaly-based IDS for mobile devices using machine learning classifiers," *Secur. Commun. Netw.*, vol. 5, no. 1, pp. 3–14, Jan. 2012.

[9] J. Hall, M. Barbeau, and E. Kranakis, "Anomaly-based intrusion detection using mobility profiles of public transportation users," in *Proc. IEEE Int. Conf. Wireless Mobile Comput., Netw. Commun. WiMob*, Aug. 2005, pp. 17–24.

[10] S. Subudhi and S. Panigrahi, "Quarter-sphere support vector machine for fraud detection in mobile telecommunication networks," *Procedia Comput. Sci.*, vol. 48, pp. 353–359, Jan. 2015.

[11] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Cell phone-based biometric identification," in *Proc. 4th IEEE Int. Conf. Biometrics, Theory, Appl. Syst. (BTAS)*, Sep. 2010, pp. 1–7.

[12] C. Nickel, T. Wirtl, and C. Busch, "Authentication of smartphone users based on the way they walk using k-NN algorithm," in *Proc. 8th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Jul. 2012, pp. 16–20.

[13] M. Jakobsson, E. Shi, P. Golle, and R. Chow, "Implicit authentication for mobile devices," in *Proc. 4th USENIX Conf. Hot Topics Secur.*, Aug. 2011, pp. 1–6.

[14] Y. Ashibani and Q. H. Mahmoud, "A Behavior-based proactive user authentication model utilizing mobile application usage patterns," in *Proc. 32nd Can. Conf. Artif. Intell.*, May 2019, pp. 284–295.

[15] Y. Ashibani and Q. H. Mahmoud, "A machine learning-based user authentication model using mobile App data," in *Proc. Int. Conf. Intell. Fuzzy Syst. (INFUS)*, Jul. 2019, pp. 408–415.

[16] Y. Ashibani and Q. H. Mahmoud, "User authentication for smart home networks based on mobile apps usage," in *Proc. 28th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2019, pp. 1–6.

[17] F. Li, N. Clarke, M. Papadaki, and P. Dowland, "Active authentication for mobile devices utilising Behaviour profiling," *Int. J. Inf. Secur.*, vol. 13, no. 3, pp. 229–244, Jun. 2014.

[18] D. Bassu, M. Cochinwala, and A. Jain, "A new mobile biometric based upon usage context," in *Proc. IEEE Int. Conf. Technol. Homeland Secur. (HST)*, Nov. 2013, pp. 441–446.

[19] U. Mahbub, J. Komulainen, D. Ferreira, and R. Chellappa, "Continuous authentication of smartphones based on application usage," *IEEE Trans. Biometrics, Behav., Identity Sci.*, vol. 1, no. 3, pp. 165–180, Jul. 2019.

[20] A. A. Alzubaidi, *Continuous Authentication of Smartphone Owners Based on App Access Behavior*. Colorado Springs, CO, USA: Univ. Colorado, 2018.

[21] Y. Ashibani and Q. H. Mahmoud, "A user authentication model for IoT networks based on app traffic patterns," in *Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Nov. 2018, pp. 632–638.

[22] Y. Ashibani and Q. H. Mahmoud, "A Behavior profiling model for user authentication in IoT networks based on app usage patterns," in *Proc. 44th IEEE Annu. Conf. Ind. Electron. Soc. (IECON)*, Oct. 2018, pp. 2841–2846.

[23] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proc. 8th IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth. CODES/ISSS*, 2010, pp. 105–114.

[24] 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 Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 136–148, Jan. 2013.

[25] B. Al-Bayati, N. Clarke, and P. Dowland, "Adaptive behavioral profiling for identity verification in cloud computing: A model and preliminary analysis," *GSTF J. Comput.*, vol. 5, no. 1, pp. 21–28, 2016.

[26] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: A survey on the security of IoT frameworks," *J. Inf. Secur. Appl.*, vol. 38, pp. 8–27, Feb. 2018.

[27] A. Aupy and N. Clarke, "User authentication by service utilisation profiling," *Adv. Netw. Commun. Eng.*, vol. 2, pp. 18–26, 2005.

[28] S. Yazji, X. Chen, R. P. Dick, and P. Scheuermann, "Implicit user re-authentication for mobile devices," in *Proc. Int. Conf. Ubiquitous Intell. Comput.* Berlin, Germany: Springer, 2009, pp. 325–339.

[29] M. Ben Salem and S. J. Stolfo, "Modeling user search behavior for masquerade detection," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*. Berlin, Germany: Springer, 2011, pp. 181–200.

[30] Y. C. Yang, "Web user behavioral profiling for user identification," *Decis. Support Syst.*, vol. 49, no. 3, pp. 261–271, Jun. 2010.

[31] M. Abramson and D. W. Aha, "User authentication from Web browsing behavior," in *Proc. 26th Int. FLAIRS Conf.*, May 2013, pp. 268–273.

[32] D. Naboulsi, M. Fiore, S. Ribot, and R. Stanica, "Large-scale mobile traffic analysis: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 124–161, 1st Quart., 2016.

[33] A. Parate, M. Böhmer, D. Chu, D. Ganesan, and B. M. Marlin, "Practical prediction and prefetch for faster access to applications on mobile phones," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. UbiComp*, 2013, pp. 275–284.

[34] V. Srinivasan, S. Moghaddam, A. Mukherji, K. K. Rachuri, C. Xu, and E. M. Tapia, "MobileMiner: Mining your frequent patterns on your phone," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput. Ubi-Comp Adjunct*, 2014, pp. 389–400.

[35] N. Alkaldi and K. Renaud, "Why do people adopt, or reject, smartphone password managers?" in *Proc. 1st Eur. Workshop Usable Secur.*, 2016, pp. 1–14.

[36] Y. Ashibani, D. Kauling, and Q. H. Mahmoud, "Design and Implementation of a Contextual-Based Continuous Authentication Framework for Smart Homes," *Appl. Syst. Innov.*, vol. 2, no. 1, pp. 1–20, 2019.

[37] I. Muslukhov, Y. Boshmaf, C. Kuo, J. Lester, and K. Beznosov, "Know your enemy: The risk of unauthorized access in smartphones by insiders," in *Proc. 15th Int. Conf. Hum.-Comput. Interact. With Mobile Devices Services MobileHCI*, 2013, pp. 271–280.

[38] A. Girardello and F. Michahelles, "Explicit and implicit ratings for mobile applications," in *Proc. Informatik Gesellschaft Inform.*, 2010, pp. 606–612.

[39] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognit.*, vol. 44, no. 8, pp. 1761–1776, Aug. 2011.

[40] R. Rawassizadeh, E. Momeni, C. Dobbins, J. Gharibshah, and M. Pazzani, "Scalable daily human behavioral pattern mining from multivariate temporal data," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 11, pp. 3098–3112, Nov. 2016.

[41] A. Rahmati, C. Shepard, C. C. Tossell, L. Zhong, P. Kortum, A. Nicoara, and J. Singh, "Seamless TCP migration on smartphones without network support," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 678–692, Mar. 2014.

[42] H. Cao and M. Lin, "Mining smartphone data for app usage prediction and recommendations: A survey," *Pervas. Mobile Comput.*, vol. 37, pp. 1–22, Jun. 2017.

[43] F. A. Narudin, A. Feizollah, N. B. Anuar, and A. Gani, "Evaluation of machine learning classifiers for mobile malware detection," *Soft Comput.*, vol. 20, no. 1, pp. 343–357, Jan. 2016.

[44] M. Rawat, N. Goyal, and S. Singh, "Advancement of recommender system based on clickstream data using gradient boosting and random forest classifiers," in *Proc. 8th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2017, pp. 1–6.

YOSEF ASHIBANI (Member, IEEE) received the B.Sc. degree in computer engineering from the College of Electronic Technology, Bani Walid, Libya, the M.Sc. degree in computer engineering from the Libyan Academy, Tripoli, Libya, and the Ph.D. degree in electrical and computer engineering from Ontario Tech University in Canada. His research interests include cyber physical systems (CPSs), the Internet of Things (IoT), and smart home security.

QUSAY H. MAHMOUD (Senior Member, IEEE) is currently a Professor of software engineering with the Department of Electrical, Computer and Software Engineering, Ontario Tech University. He was the Founding Chair of the Department and served as the Chair between January 2013 and June 2015, and more recently has served as an Associate Dean of the Faculty of Engineering and Applied Science, Ontario Tech University. His research interests include intelligent software systems and cybersecurity.