A Contextual Authentication Framework for Smart Home Environments

by

Yosef Ashibani

A thesis submitted to the
School of Graduate and Postdoctoral Studies in Partial
Fulfillment of the Requirements for the Degree of

Doctor of Philosophy in Electrical and Computer Engineering

Department of Electrical, Computer and Software Engineering

Faculty of Engineering and Applied Science

University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

April 2020

THESIS EXAMINATION INFORMATION

Submitted by: **Yosef Ashibani**

Doctor of Philosophy in Electrical and Computer Engineering

Thesis title: **A Contextual Authentication Framework for Smart Home Environments**

An oral defense of this thesis took place on April 06, 2020 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Shahryar Rahnamayan
Research Supervisor	Dr. Qusay H. Mahmoud
Examining Committee Member	Dr. Khalil El-Khatib
Examining Committee Member	Dr. Masoud Makrehchi
University Examiner	Dr. Patrick Hung
External Examiner	Dr. Abdelkader Ouda, Western University

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

A Contextual Authentication Framework for Smart Home Environments

Yosef Ashibani
Ontario Tech University, 2020

Advisor:
Dr. Qusay H. Mahmoud

A smart home is one equipped with connected Internet-of-Things (IoT) devices that can be remotely accessed and controlled. Access to smart home devices is mostly achieved through smartphones and tablet computers, but this comes with security challenges such as unauthorized access and interception of data transmission. Although smart home devices are critical, many examples of security challenges, such as unauthorized access to home devices and interception of data transmission, are reported. Furthermore, many home IoT devices are still shipped with default credentials even though it is widely known that these settings are used in attacks. A number of cryptographic schemes have been proposed for securing communication among home IoT devices. However, the ability to handle such schemes, especially by devices with constrained computing resources, can be challenging. To address the above issues, this thesis introduces a contextual authentication framework for smart home environments that integrates a context-based user authentication method, a device-to-device message authentication scheme, and an app-based user authentication model. A proof of concept prototype of context-based authentication has been constructed. An identity-based signcryption scheme for securing data transmission between home IoT devices has been designed, and an app-based user authentication model has been developed. The results demonstrate that considerable contextual information can be retrieved and such information can be used in providing seamless, usable, and secure authentication. Furthermore, analysis and evaluation of the proposed signcryption scheme demonstrate that, in addition to providing authentication, it provides integrity and confidentiality as well as the ability to protect communication against possible attacks. The evaluation of the app-based user authentication model is performed on three datasets, and the results show that the model has the ability to authenticate users with high accuracy in terms of low false positive, false negative and equal error rates.

Keywords: Smart home security; context-based user authentication; app-based user authentication; device-to-device message authentication; identity-based signcryption.

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

_____ Yosef Ashibani

STATEMENT OF CONTRIBUTIONS

I hereby certify that I am the sole author of this thesis, and I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others. Furthermore, I hereby certify that I am the sole source of the creative works and/or inventive knowledge described in this thesis.

Results from this thesis research have been disseminated in the following publications:

- Y. Ashibani and Q. H. Mahmoud, "Classification and Feature Extraction for User Identification for Smart Home Networks Based on Apps Access History," in *18th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2019, pp. 376–380.
- Y. Ashibani and Q. H. Mahmoud, "User authentication for smart home networks based on mobile apps usage," in *28th IEEE International Conference on Computer Communications and Networks, ICCCN*, 2019, pp. 1–6.
- Y. Ashibani and Q. H. Mahmoud, "A Machine Learning-Based User Authentication Model Using Mobile App Data," in *International Conference on Intelligent and Fuzzy Systems (INFUS)*, 2019, pp. 408–415, (Best Paper Award).
- Y. Ashibani and Q. H. Mahmoud, "A Behavior-Based Proactive User Authentication Model Utilizing Mobile Application Usage Patterns," in *32nd Canadian Conference on Artificial Intelligence*, 2019, pp. 284–295.
- Y. Ashibani, D. Kauling, and Q. H. Mahmoud, "Design and Implementation of a Contextual-Based Continuous Authentication Framework for Smart Homes," *Applied System Innovation*, vol. 2, no. 1, pp. 1–20, 2019.
- Y. Ashibani and Q. H. Mahmoud, "A User Authentication Model for IoT Networks Based on App Traffic Patterns," in *9th IEEE Annual I Information Technology; Electronics and Mobile Communication Conference (IEMCON)*, 2018, pp. 632–638.
- Y. Ashibani and Q. H. Mahmoud, "A Behavior Profiling Model for User Authentication in IoT Networks based on App Usage Patterns," in *44th IEEE Annual Conference of the Industrial Electronics Society (IECON)*, 2018, pp. 2841–2846.
- Y. Ashibani and Q. H. Mahmoud, "An Efficient and Secure Scheme for Smart Some Communication Using Identity-Based Signcryption," in *36th IIEEE International Performance Computing and Communications Conference, IPCCC*, 2017, pp. 1–7.

- Y. Ashibani, D. Kauling, and Q. H. Mahmoud, "A Context-Aware Authentication Framework for Smart Homes," in *30th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE),* 2017, pp. 1–5.
- Y. Ashibani and Q. H. Mahmoud, "Cyber Physical Systems Security: Analysis, Challenges and Solutions," *Journal of Computers and Security, Elsevier*, vol. 68, pp. 81–97, 2017.
- Y. Ashibani, D. Kauling, and Q. H. Mahmoud, "Poster: A Context-Aware Authentication Service for Smart Homes," in *14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2017, pp. 588–589.

DEDICATION

Dedicated to my family, without their support and encouragement none of this would have been possible.

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor, Dr. Qusay H. Mahmoud, for his generous support, continuous encouragement, and constant efforts to push me out of my comfort zone. Dr. Mahmoud has also helped tremendously in improving my research and communication skills. I cannot possibly quantify the importance of his academic guidance, recommendations, constant feedback, quick responses, and valuable advice during these past five years. He has unfailingly shared with me his time, resources, and judgment. Knowing him as someone who consistently cares about his students, I had the opportunity to discuss with him the obstacles encountered in my study and research. Dr. Mahmoud has not only helped me to develop academic skills, but has also encouraged me to strive for excellence. Dr. Mahmoud has guided and helped me improve several sections of my thesis. Without his valuable comments and sincere suggestions, which have significantly enhanced my work, this thesis would not have been completed. Therefore, many thanks go to him for his immense help in guiding me toward the successful completion of my Ph.D. studies.

In addition, I would like to express deep gratitude to my other supervisory committee members, Dr. Khalil El-Khatib and Dr. Masoud Makrehchi, not only for their insightful comments and encouragement but also for encouraging me to widen my research from various perspectives. I would also like to thank the university examiner Dr. Patrick Hung and the external examiner, Dr. Abdelkader Ouda for their valuable feedback, and Dr. Shahryar Rahnamayan for chairing the thesis examination.

I would also like to thank all my colleagues from the department for their continuous support and cooperation during these five years. I wish to express my appreciation to Mrs. Catherine Lee, Mr. Mark Neville, and Dr. Fernanda Batista as well, for their discussion, editing and proofreading of this thesis. My special thanks, appreciation, and gratitude also go to my parents. Together with all my family, their desire to see me accomplish this goal has sustained me on this colossal project. I thank them for always believing in me and supporting me. Their love and encouragement have been, and always will be, a great source of inspiration in my life.

Lastly, but importantly, I want to thank my wife, my daughters, and my son for their patience throughout the duration of my Ph.D. studies, as well as for their understanding of my need to spend time away from them working and writing up my dissertation. Long nights and busy weekends have been the norm. While the work has seemed never-ending, they have always been there for me. Last but not least, I would also like to thank all those who have been directly and indirectly involved in building this thesis and research work.

Table of Contents

List of Tables

Chapter 2

Chapter 3

Chapter 4

List of Figures

Chapter 1

Chapter 3

Chapter 4

List of Abbreviations

AER	Average Error Rate
API	Application Programming Interface
EER	Equal Error Rate
FAR	False Acceptance Rate
FMR	False Match Rate
FRR	False Rejection Rate
HG	Home Gateway
IBE	Identity-Based Encryption
IBS	Identity-Based Signcryption
IoT	Internet of Things
KNN	K-Nearest Neighbor
TS	Trusted Server
MERR	Median Equal Error Rate
NB	Naïve Bayes
PIN	Personal Identification Number
RF	Random Forest
SVM	Support Vector Machine
TAR	True Acceptance Rate
TNR	True Negative Rate
TPR	True Positive Rate
TRR	True Rejection Rate

Chapter 1. Introduction

This chapter first provides an overview of smart home security and authentication techniques followed by motivation and problem statement, research contributions and the thesis structure.

1.1 Overview

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 providing access, monitoring and control of home devices, smart homes provide other services to home residents, such as entertainment and information storage, thus making users' daily lives more comfortable [1]. SmartThings [2], Wink [3] and HomeKit [2] are examples of smart home platforms. These home platforms are built based on the cloud back-end, where control management and authentication are mostly performed through an installed application on end-user devices, such as smartphones and tablets. Consequently, access to smart home environments, as shown in Figure 1.1, is mostly achieved remotely through users' end devices. These devices have become essential tools for accessing and operating smart home networks. Since the smartphone is susceptible to loss or theft, there is a need for a transparent authentication mechanism that can implicitly authenticate the user without requiring more explicit intervention. Thus, smart home networks should be enhanced with security measures that ensure the access request and control commands come from the authorized user. A solution can be achieved by utilizing the continuous authentication of the current user of the mobile device. Continuous authentication is the process of continuously checking the user's identity at and beyond the login stage. One advantage of

implicit continuous authentication, which is based on user interaction, is that it provides security and usability while reducing explicit user intervention unless required.

Considerable information, such as application (app) usage, can be extracted from mobile devices and used to support authentication either at the entry point or during the access session. 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 [4]. As well as the apps available in stores, the number of Android apps at Google Play are approximately two millions [5], in addition to those new apps that are added daily. The number of apps used is also increasing, with most interactions on mobile phone devices related to foreground apps.

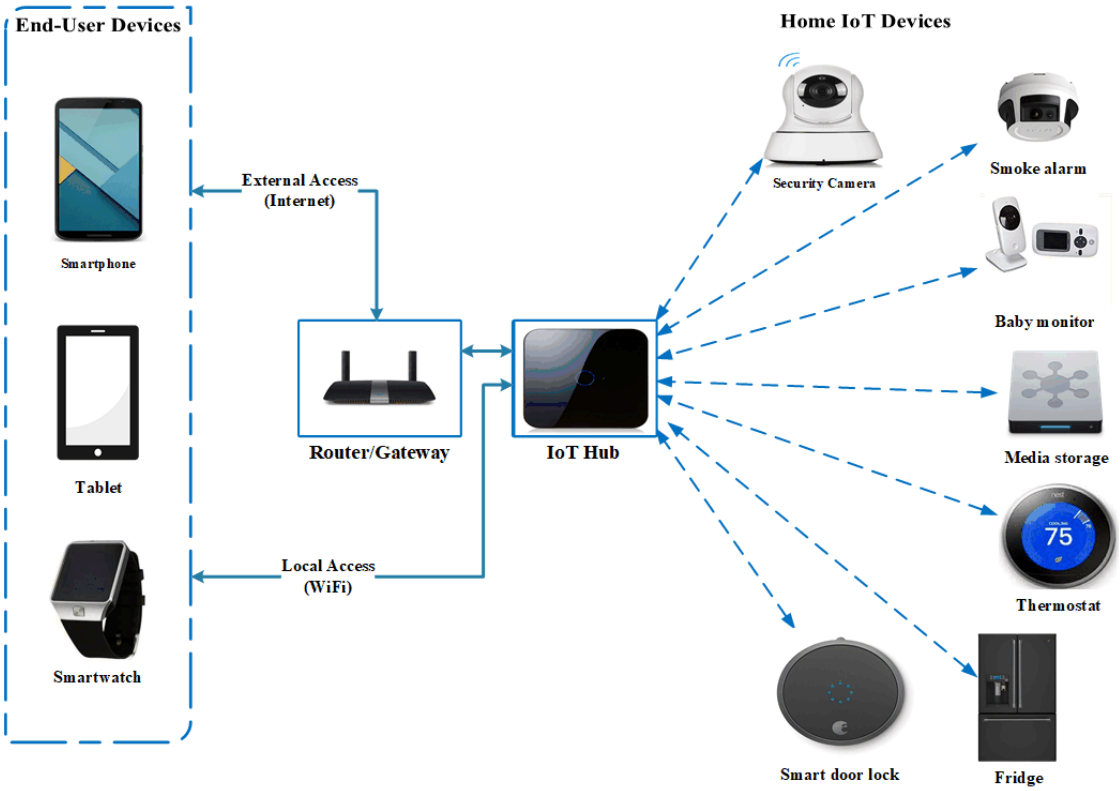

Figure 1.1. An example of a smart home environment

With an increasing number of apps being used on mobile phones, app usage behavior may provide discriminative information about users. Furthermore, app-based access exemplifies information on user patterns while interacting on mobile devices. For instance, the ability to continuously retrieve app usage profile data on mobile devices strengthens the argument for employing behavioral-based mechanisms for continuous user authentication. Behavior profiling, which has many advantages [6], including the ability to be continuously computed without the knowledge of the user, can be obtained without additional hardware. Utilizing user app interaction can be profiled on smartphones/tablets based on functions provided by most of the operating systems on these devices.

Another important issue is that insecure communication from one home IoT device to another may lead to information disclosure which, in turn, could be targeted by attackers. Attacks against such devices, which often have limited capabilities and are usually connected through the Internet by commonly using less secure wireless media, might easily achieve access to sensitive data. Hence, an effective solution that secures communication among these devices is the cryptographic technique.

1.2 Motivation and Problem Statement

Mobile phones and tablets are becoming global end-user devices in smart home networks. Although access to a variety of home IoT devices can be achieved through end-user devices, these devices are susceptible to theft or loss. According to Kensington, 70 million smartphones are lost annually while 4.3% of company-issued smartphones are lost or stolen every year [7]. The use of these devices by unauthorized users could lead to unauthorized access to home IoT devices. Many examples of security attacks have been noted throughout the literature, such as unauthorized access to heating systems, baby monitors, air

conditioning, thermostats, and lighting systems [8][9][10][11], allowing access to home networks as a result of weak authentication methods being employed. Furthermore, regarding IoT security, Hewlett Packard [12] reported that 80% of tested devices had insufficient authentication or authorization, a further 80% showed privacy concerns, and 70% used unencrypted communication channels. It is reported in [13] that some services add a layer of authentication to mobile devices, such as sending an additional password via SMS before logging onto the service's website. However, this solution does not solve the problem when the mobile phone is stolen. For example, as a result of utilizing weak authentication credentials by users, the attacker would have everything necessary to access the service.

According to the MacAfee Report 2019, IoT devices are still shipped with default usernames/passwords even though it is widely known that these credentials have been, and continue to be, used in attacks [14]. As reported in [15], many of the consumer IoT device manufacturers fail to either focus on security or offer security instructions to consumers. Moreover, default usernames and passwords are the most common information for attacking IoT devices [16], with around 57 % of known passwords used in the attacks. Additionally, IoT routers are the most targeted and attacked devices, accounting for 75 % of attacks due to accessibility gained from the internet [17]. Moreover, many smartphone users still adopt weak login credentials, including common or reusable passwords, or no password at all [18].

Although the literature, such as [19], presents a compound password approach, this does not guarantee that users will use them. Moreover, requiring users to enter security credentials on each specific occasion in order to be continuously authenticated is

undesirable. However, offering security mechanisms to users on end-devices with the option of bypassing them, even if they are enabled by default, does not guarantee that users will utilize such options to secure their devices and accessed services. Another important issue is related to acknowledgment that attacks might not only come from outside the system but also from inside.

An additional issue that should be considered when retrieving information related to users is that of securing any sent messages. Therefore, transferring any information between home devices must be protected from disclosure. However, some terminal devices, such as sensors and actuators, do not have adequate data processing capabilities and communication abilities, or adequate storage capacities. Consequently, it is very challenging to implement these methods, especially by devices with limited computation capabilities.

For continuous user authentication, some of the proposed solutions, such as those presented in [20][21][22][23] and [24], require a particular series of actions from the user for authentication, such as gesture recognition approaches which, in turn, eliminate the transparency of continuous authentication. However, only a few studies employ app profiling for continuous user authentication. Although the achieved results in such studies are prominent, the range of apps used in the evaluation was limited. In addition, these studies, as in [25], focus on appl-specific information, such as text messages and calling behavior, for detecting abnormal usage in the mobile environment. It should also be noted that the focus of the literature is on single modality in which the built models target single users. 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. In order to protect transmitted messages between IoT devices, many lightweight security schemes based on symmetric-key cryptography have been proposed. These schemes need a number of shared symmetric keys for each party. If the shared key is compromised, all the communicated messages will be compromised. For providing higher security, public-key cryptography has been used. However, public-key cryptography still needs a certificate for each public key, which is considered to be a complex process [26].

1.3 Research Objectives

The main objective of this thesis is to investigate the design and evaluation of a contextual authentication framework for smart home environments. This framework includes a user authentication model that protects smart home IoT devices from unauthorized access. Information exchange security between home IoT devices is also considered to protect any transferred messages among home IoT devices.

1.4 Research Contributions

The significant contribution of this thesis is a contextual authentication framework for smart home environments, and the individual contributions that embody the framework are:

- **A context-based user authentication method.** This method enhances the knowledge-based authentication with contextual information which ensures the authenticity of a user beyond the point-of-entry. The implementation and evaluation of a proof of concept prototype as well as the utilization of contextual information for context-based

user authentication are presented. Results from this research have been published in [149][154][155].

- **An identity-based signcryption scheme for information exchange among smart home IoT devices.** The presented scheme is based on identity-based cryptography and, in addition to providing authentication, it provides integrity and confidentiality as well as the ability to protect communication among devices against various possible attacks. This scheme is more efficient regarding computational cost and cipher-text length compared to other existing signcryption schemes. Results from this research have been published in [156].
- **An app-based user authentication model.** This model is able to authenticate registered users utilizing app interactions with considerably high accuracy. This model does not require specific action from the user in order to be authenticated; rather, it is based on regular actions while accessing apps, which enhances usability. Results from this research have been published in [86][[93][94]125][145][148].

1.5 Thesis Outline

This thesis considers the aforementioned research objectives and consists of five chapters as follows:

Chapter 1 presents an overview, motivation and problem statement followed by the research objectives and contributions of the thesis.

Chapter 2 reviews the literature regarding user authentication mechanisms, which are organized into three classifications: common authentication, contextual-based authentication, and continuous authentication. Next, it reviews device-to-device message authentication mechanisms and presents a discussion and summary.

Chapter 3 introduces the proposed framework, assumptions, threat model, as well as the individual components of this framework including a context-based user authentication

method, a device-to-device message authentication scheme, and an app-based user authentication model.

Chapter 4 presents the implementation of the proof of concept prototype, the experimental evaluation results, and the security analysis.

Chapter 5 concludes the research by summarizing the findings and limitations as well as suggesting future areas for exploration.

Chapter 2. Background and Related Work

This chapter reviews the relevant literature regarding user authentication and divides the presented authentication mechanisms into three classifications: common authentication, context-based authentication, and continuous authentication mechanisms. This chapter also reviews the literature related to proposed public cryptographic mechanisms for smart homes.

2.1 Introduction

Authentication has been utilized within different technologies, such as mobile phones, mobile networks, and web browsing, for either the client-side or the server-side [27]. From the client-side perspective, the studies in [28][29][30] utilize features, such as opened files, including how frequently these files are accessed while accessing a computer system in order to detect unauthorized access. From the server-side perspective, other research studies, such as [31][32], have investigated the potential to build a user profile based on web accessed activities in order to identify users. The utilized features in these studies include the visited website name, start time, and the number of browsed pages. Other studies have adopted third-party mechanisms to offload user behavior model training and authentication processes. As an example, in [33], 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 back-end or at the SmartHub controller in the smart home network. Several mechanisms for user and device authentication, which can be adopted for smart homes, have been identified in the literature. Accordingly, this chapter reviews the literature and divides the presented user authentication mechanisms into three classifications: common authentication, context-

based authentication, and continuous authentication mechanisms. Another issue involves the security of transmitting information among home IoT devices. For example, if an adversary exploits a transmitted message, it will threaten data integrity and confidentiality. Hence, transferring any information that can be utilized for characterizing user behavior must be protected from disclosure. One important aspect of security regarding data transmission is device-to-device message authentication. On the issue of securing communication between smart home IoT devices, this chapter also reviews the literature related to two aspects: proposed public cryptographic mechanisms for smart homes and proposed public cryptographic mechanisms that can be adopted for smart homes.

2.2 Authentication Mechanisms

The most common authentication mechanisms as presented in the literature fall into three categories: knowledge-based authentication, object-based authentication, and biometrics. The first category consists of knowledge-based authentication approaches such as passwords and PINs. This approach needs users to remember authentication credentials, which may be susceptible to attacks in scenarios such as brute-force, dictionary, and phishing attacks. Although, as demonstrated in [34], the password is still the preferred option for authentication by many users and easy to implement [35], many incidents of password attacks have occurred during the last few years [36]. For example, credentials of over 7 million accounts were leaked in 2016 from a community on Minecraft. Additionally, hackers typically publish breached credentials on public online lists such as the Anti Public Combo List, which contains more than 500 million credentials leaked during the past several years [37]. Furthermore, there have been recent attacks resulting in the breach of 4 billion username-password pairs in Google's database [38]. Single-Sign-On (SSO) is a user

authentication technique that reduces the frequent authentication of the user; however, it only identifies the identity not the user [39][40]. Although this method can remove from users the burden of providing login credentials for many apps, it could increase the risk of hacking any other apps that can be accessed under the same authentication if, for example, the login credentials are accessed by an attacker. Moreover, SSO is still a point-of-entry that assumes that the legitimate user is the one who accesses the service throughout the entire access session [40]. Furthermore, it is impractical to require users to continuously provide knowledge-based credentials to prove their presence during the access session.

The second category consists of possession-based authentication approaches, such as using tokens, which have been developed for overcoming the drawbacks that come with knowledge-based approaches. Token-based authentication methods can be classified into two types: hardware tokens and software tokens. Hardware token mechanisms need the user to carry physical tokens. For example, a continuous authentication method for wearable wireless devices is suggested in [41]. As an example, in [42] a transient authentication mechanism is proposed for user authentication through small hardware tokens. Taking into account that such tokens are susceptible to theft or loss, this approach is impractical.

Software tokens utilize the end-user device by, for example, sending the password as a message to the user's registered device, such as a smartphone or tablet, to be used for logging into the service. Software tokens can be in the form of installed software on the end-user device which can issue new passwords in the form of a One Time Password (OTP), such as Google Authenticator [40][43], that changes with every access time. Software token-based authentication has some advantages over both hardware token-based

and knowledge-based authentication techniques as it removes from users the burden of having to carry physical tokens and remember and choose robust passwords. However, it is inconvenient for users to have to continuously input the generated passwords for continuous authentication throughout the entire access session in addition to the drawback of possible access to the end-user device by an unauthorized user.

The third technique for overcoming the limitations of using knowledge-based and object-based authentication approaches is the biometrics technique which is divided into physiological and behavioral. The use of physiological biometric, such as iris and fingerprint recognition, is being used where users are expected to access required services [44][45]. Biometric features, such as fingerprints, can be considered as robust for user authentication. However, for the sake of continuous authentication, it would be inconvenient to require users to re-enter their biometrics every time. As observed from these three authentication mechanisms, there is a need for an implicit authentication mechanism that can provide continuous authentication during the access session.

2.3 Context-Based Authentication

This section first presents the definition of the contextual information in the literature, the taxonomy and quality of the contextual information, and reviews context-based authentication approaches.

2.3.1 Contextual Information

Although there are many contextual information classifications, there is still no unified definition for contextual information. Many researchers consider the following definition as appropriate: “Context is any information that can be used to characterize the situation of

an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications [46][47]". Context includes any information that is related to a user's situation, such as location, device status, and any information related to the environment, such as temperature, loudness, and brightness.

Another classification of security-relevant contextual attributes includes physical environment context, (e.g., time or temperature); service type context (e.g., premium or basic service); user's context (e.g., location); platform context (e.g., trusted state of the platform); and particular transaction (e.g., an electronic token or electronic receipt) [48]. In addition, contextual information includes personal contexts (e.g., preferences); activity contexts (e.g., meeting schedule or shopping list); physical contexts (e.g., time and location) [46]; device contexts (e.g., power and display size); systematic contexts (e.g., network bandwidth); application contexts (e.g., agent and service); and environmental contexts (e.g., light level) [49][50].

There are many contextual information classifications. For example, [51] classifies contextual information based on operational categorization to sensed, static, profiled, and derived contexts. In [52], contextual information is classified based on three common questions regarding where you are, who you are with, and what resources are nearby. However, from a security perspective, especially for the authentication process, contextual information can be classified as follows:

- Direct Contextual Information: This can be achieved directly without performing operations on, or modifying, the contextual values (e.g., activities or historical movement information) as well as social relationships (e.g., friends and family). Direct

contextual information, which is mentioned in [53] as the primary context, includes any retrieved contextual information without relying on already existing contextual information. Direct contextual information includes:

  - User context: any information about the user such as profile, calendar, social networks, and access patterns
  - Device context: any information related to the used end-device which can be retrieved from sensors [54][55], such as location, current and voltage values, Wi-Fi access points, operating systems, and running/installed apps
  - Network data: IP address, media access control address (MAC address), link speed, ping times, and traceroutes
  - Environmental context: any information related to the physical environment, such as temperature, weather, lighting, loudness, or humidity

- Indirect Contextual Information: This can be indirectly achieved by performing operations on or modifying the contextual values: for example, calculating power consumption using voltage and current values or speed from multiple GPS locations. This classification is mentioned in [53] with the term 'secondary contextual information', which refers to any contextual information resulting from performing any operation on existing contextual information.
- Other classifications: Contextual information can also be classified based on the status of the system and its characteristics, as well as the resource of the involved contextual information attributes. Such classifications could include:
  - Static contextual information: contextual information that changes very slowly or does not change at all, and includes the address and name of the user

- Dynamic contextual information: contextual information that changes over time, such as time and location of the user
- Internal contextual information: contextual information that is retrieved from the used devices by the user including battery level, and current and voltage readings
- External contextual information: contextual information that can be retrieved from external resources such as the location of the user, as retrieved from the cellular network

Another important factor regarding retrieved contextual information that is related to security is quality of the contextual information (QoC). The quality is based on the accuracy and trust of the received values. While accuracy is often desirable in apps such as GPS location tracking, [56] mentions that the QoC is only related to the information itself, and does not involve the resource that provides the contextual information or the performed processes. Thus, it is important to consider QoC regarding the accuracy and trust of the received values. Many different methods [57] can be adopted for measuring the quality of contextual information. The popular approaches include:

- Statistical analysis, which is based on mathematical models to exclude unreliable values. For example, retrieving the temperature and pressure values from different resources and discarding any anomalous values. Statistical analysis is an effective approach that can be applied especially to environmental contextual information, and also when having multiple resources of the received contextual information.
- Confidence value, which is based on assigning different confidence values for the involved authentication tools and the used devices. As an example, physical tokens

have higher confidence values than knowledge-based credentials such as username and password. In this thesis, higher confidence values will be assigned to the retrieved contextual information based on the trust of the related resource. For instance, by considering MAC address spoofing, a MAC address will be assigned a lower confidence value. In addition, the confidence level will be assigned based on information resulting from historical analysis of long-term contextual information.

2.3.2 Authentication Methods

Other research studies offer several solutions that utilize contextual information in two ways: either by integrating this information as an authentication adaptation or as additional authentication factors [46]. As previously mentioned, the knowledge-based authentication and object-based approaches only prove the presence of the identity, and not the actual user. Thus, if credentials or tokens are stolen, loaned, misused or forged, there will be no proof of the user's legitimacy. In general, authentication approaches that are based on usernames and passwords or tokens prove only the presence of the identity, and not the user. Consequently, if an illegitimate user is using the end-user's device before the expiry of the access session, all apps and services that are accessed via this device could be accessed by unauthorized persons if there is no technique for verifying if the current user is authorized one. Much research has been conducted to enhance traditional authentication techniques with contextual information. Table 2.1 provides a comparison of relevant related works regarding utilized information, advantages, and limitations. A contextual attribute authentication model for mobile environments is defined in [48]. However, this model only includes location and e-receipt as contextual information. The authors in [58] provide an authentication approach that involves GPS location, time and the tasks

performed on the operating system. However, their approach considers location based on a GPS signal that will not be available when accessing services indoors.

Table 2.1. A comparison of context-based authentication approaches

Ref No.	Utilized Information	Advantages	Limitations
[25]	Calls, SMS and GPS based location.	Enables implicit and continuous authentication.	Only uses GPS location, which will not be available in most cases, especially indoors. In addition, SMS and calling functions are ignored and replaced with apps that achieve the same purpose.
[39]	SMS, calls and geographic location.	Provides implicit continuous authentication and demonstrate the effectiveness of implicit authentication in distinguishing the legitimate user from the adversary.	Only uses GPS coordinates collected, which will not be available in most cases, especially indoors. SMS and calling functions are ignored and replaced with apps that achieve the same purpose.
[42]	Small hardware tokens.	Provides continuous authentication based on the user's presence.	Limits access to the use of location-based contexts.
[43]	SMS	Provides transparent and continuous authentication in addition to implementation.	SMS function is ignored and replaced with apps that achieve the same purpose.
[48]	Location and e-receipt.	Provides authentication model for mobile environments.	Only includes location and e-receipt as contextual information based on the user's presence.
[58]	Location, time and operating system processes.	Provides context-aware authentication and implementation.	Only uses GPS location, which will not be available in most cases, especially indoors.
[59]	Hardware token (RFID tags) location	Perform authentication and access control approach in a very flexible and scalable model	Only limits access to the use of location. Also, it does not provide any implementation or evaluation of the proposed framework.

Based on the assumption that mobile users tend to use their apps in different locations at different times, location-based authentication is described in [25]. Assuming that users perform similar tasks at certain times during weekdays, a user profile approach that gathers behavioral information, such as sent text messages (SMS), calls and geographic location, is proposed in [39]. For observed actions, such as habitual or good events, this approach assigns a score. The study in [60] proposes an anomaly-based detection system based on monitoring users' actions, such as sending SMS messages or making calls. Additionally, other studies, such as [25][39] and [60], consider SMS and calling behavior for abnormal usage detection. As the number of mobile apps increases, SMS and calling functions are being ignored and replaced with apps that achieve the same purpose. In addition, unauthorized users who access mobile devices will, most probably, tend to operate inconspicuously; hence these functions provide insufficient evidence of the intended user.

An authentication method for wearable wireless devices is suggested in [41]. In [42], a transient authentication mechanism is proposed for user authentication through small hardware tokens, limiting access to the use of location-based contexts. The research in [59] provides a context security framework utilizing proximity and does not provide an implementation and evaluation of the proposed framework.

2.4 Device Authentication

In recent years, many schemes have been proposed for securing communication either intended or that could be adopted for smart homes. For communication between any two devices, many available devices in the market are provided with the ability to secure communication with a pre-shared symmetric key which should be exchanged in advance for encrypting and decrypting any transmitted messages. This approach requires a regular

key exchange among devices. If the used key is compromised, especially when perfect forward secrecy protocols (PFS) are not utilized, all communicated messages will be compromised [61]. For example, the authors in [62] and [63] present a lightweight cryptographic technique for resource-constrained devices by combining symmetric and asymmetric cryptography, providing confidentiality, integrity and authentication. However, the proposed technique mainly depends on symmetric cryptography for data encryption.

A lightweight encryption scheme for smart homes that provides confidentiality with less overhead in computation and communication is presented in [64]. The scheme adopts Identity Based Encryption (IBE), which requires less public key management. The authors also provide a security analysis showing the efficiency of the proposed scheme. However, the focus of the proposed scheme is mainly on confidentiality. Moreover, it is based on symmetric encryption for message encryption, which weakens the security of the proposed scheme. A common symmetric key that is automatically generated according to extracted parameters from wireless multi-path channels is presented in [65]. This approach is mainly intended for devices that support 802.11a protocol. The feasibility of this approach being handled by many devices is still debatable as not all smart home devices support 802.11a protocol. A radio frequency for consumer electronics secure key pairing protocol is proposed in [66]. For authentication, the proposed scheme requires each device to communicate with its manufacturer. In this technique, each involved device is required to send authentication information to its manufacturer through a mobile operator in order to be authenticated. This scheme is based on symmetric-key cryptography, and there is always a need for access to the device's manufacturer in order to be authenticated. A security

scheme that provides three levels of authentications among gateway, smart meter, smart appliances and the home area network is proposed in [67]. However, the provided scheme mainly depends on a third party (e.g., the Internet service provider) for providing three authentication levels between the devices.

A secure smart household appliance framework (S2A) has been proposed in [68]. The focus of this framework is on providing safe operations, smart home safety and electricity price control. For reliable security protection, this framework employs machine learning but does not account for device authentication, integrity and data confidentiality, which are the main security goals. A key establishment protocol for smart homes is presented in [69]. This protocol requires a trusted certificate authority for providing public and private keys. Although the authors provide limited security analysis, it needs to be emphasized that involving traditional public-key cryptography requires a third party for certification and, in turn, produces more overhead on constrained devices.

The authors in [70] propose a lightweight mutual message authentication scheme for reducing the number of exchanged messages during authentication. Although the proposed scheme provides two-step mutual authentication, it uses public-key encryption. The authors in [71] propose a framework in which a local home server authenticates a remote user on behalf of smart devices, utilizing fingerprint and location. Communication between a gateway and smart devices is assumed to be secured by other schemes. The research in [72] presents an approach to securing interaction between RFID-tagged consumer items, RFID-reader enabled appliances and RFID-based applications. However, the proposed approach, which is only a conceptual idea, assumes that the home appliances are RFID supported. Kerberos is a network authentication protocol that requires a third party (a key

distribution centre) to achieve temporary keys to access locations on the network. Consequently, Kerberos allows for secure and robust authentication without transmission of passwords [73]. OAuth 2.0 is an open standard framework that enables a third-party app to obtain limited access to a service without providing authentication credentials. OAuth is about authorization since it does not transmit authentication information between clients and service providers [74]. In comparison with OAuth, OpenID is a distributed (decentralized) open identity standard authentication which delegated authentication using a third party. OpenID eliminates the need for providing login credentials and allows users to log into multiple websites without providing separate credentials for each website [75][76].

2.5 Continuous User Authentication Mechanisms

As of result of the difficulty inherent in both knowledge-based and object-based authentication mechanisms for continuous and transparent user authentication, the focus has turned to behavioral-based authentication [40]. In addition, the ability to continuously utilize usage app profiles and sensor data on end-user devices strengthens the argument for employing behavioral-based mechanisms for continuous user authentication. Behavioral-based authentication is centred on the assumption that users have stable and distinct usage patterns while employing end-user devices [77]. Additionally, behavioral-based authentication mechanisms identify users according to unique behaviors, such as the usage pattern of the apps on the device, the way they walk while handling a device, and the way they hold the device during usage.

Moreover, behavioral-based authentication has many advantages such as the ability to provide continuous and implicit authentication of users beyond the point-of-entry.

Although behavioral authentication techniques can be affected by a change in the discriminatory characteristics of the user [78][79], this variation can be minimized by long-term behavior analysis and regular updates. In addition, behavioral authentication techniques do not require explicit user intervention. Furthermore, a study presented in [80] shows that a transparent and continuous authentication is desired by users on mobile devices. Although there are different mechanisms that consider behavioral-based user authentication in the literature, in this chapter we only consider behavioral profiling authentication and combined authentication mechanisms which are described in the following subsections.

2.5.1 Behavioral Profiling-Based Authentication

The behavioral profiling-based authentication technique is mainly established on the hypothesis that users present a unique behavior, such as working on a specific app in a specific period. Much research has been conducted into enhancing user authentication by utilizing behavior profiling. This type of behavior can be modeled by capturing app usage logs from end-user devices. Behavior profiling for authentication, which started in the late 1990s, can be classified as either network-based or host-based. Network-based behavior profiling, such as described in [81], utilizes user calling and migration behavior over the service provider network for building a user behavior profile. In contrast, host-based behavior profiling, as described in [25], is based on the hypothesis that mobile users tend to use their apps at different times in different locations. Table 2.2 provides a comparison of relevant related works regarding utilized information, advantages, and limitations.

Drawing on the assumption that most users are prone to performing similar tasks at a certain time of day, the authors in [39] present a method that creates user profiles by

collecting behavioral information such as geographic location, phone calls and SMSs. Their approach assigns a score to observed events such as a good or habitual event. An experimental result in [25] shows that app-level behavior profiling is able to discriminate between users and detect anomalies in the course of interaction with a device, which in turn enables implicit and continuous authentication for users. The achieved results in the evaluation of this method on data from 30 users for one month include EER=13.5%, 35.1% and 35.7% for telephone calling, device usage and Bluetooth scanning.

Additionally, it is demonstrated in [82] that user behavior is, in fact, subject to the app being employed by the users. An anomaly-based detection system that monitors the actions of users, such as calls, SMSs and Web browsing on mobile phones, is presented in [60]. For performance evaluation of this work, four different machine learning classifiers were applied. Two behavioral features considered in the proposed solution in [83] 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. The study in [84], which presents a user authentication method based on telephone calls, basic app-level usage, and Bluetooth scanning, achieved an EER of 13.5%, 35.1% and 35.7%, respectively. In [85], the same authors subsequently 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 calls, 2.2% for SMS and 9.8% for multi-instance. The authors in [86] 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.

Table 2.2. A comparison of behavioral profiling-based authentication approaches

Ref No.	Utilized information, Authentication Method and Accuracy Measure	Advantage	Limitations
[21]	Gesture patterns; five basic user movements; 75 users; support vector machine (SVM).	Continuous authentication for smartphones. Re-authentication module deployed in a smartphone and training module is executed on a PC.	Requires a particular series of actions; Utilizes two classification modules for every gesture to deal with each orientation mode; This study excludes new apps.
[22]	Accelerometer-based gait; recognition; 36 users; k-nearest neighbors (KNN); EER= 8.24.	Algorithm performs well in a controlled environment when subjects walk on flat floor.	Requires a particular series of actions; Needs 30 seconds for authentication; Authentication currently is based on 30 seconds' walk data; This study excludes new apps.
[23]	Keystroke; 18 users; EER < 2%, FAR = 11%, FRR =16%.	Two-factor authentication without carrying more hardware.	Requires a particular series of actions; This study excludes new apps.
[60]	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.
[83]	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.
[85]	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.
[87]	Apps usage, location, clock time, gesture, voice, touch; One class per user and single model	Combines several features, resulting in	SMS count only for the past hour in addition to the time and GPS location; SMS and

	per user; Naïve Bayes; FAR, FRR, TAR, TRR.	universal and unique modality for users.	calling functions are ignored and replaced with apps that achieve the same purpose; This study excludes new apps.
[88]	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 discriminate between users; For the active authentication problem, the preferred language of the user is a type of behavioral data that can be used to discriminate between users; Considers apps that are used only by individual users.
[89]	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 four modalities; It requires a total of 800 classifiers; Requires five-minute threshold for what is considered an idle period; This study excludes new apps.
[90]	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.
[91]	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.
[92]	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 of the proposed approach is based on ten consecutive days of training dataset; This study excludes new apps.

A user behavior profiling that describes where, when, how and with what the devices were used, is proposed in [87]. A user authentication approach that utilizes the access history of app usage events employing only a small amount of information is reported in [93]. The work in [88] presents a continuous authentication model on smartphones based on app usage data. The achieved results in the evaluation of this method include an average of EER=16% from the first dataset and 30% based on 50 historic observations sample from the 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 discriminate between users. The work in [93] is extended in [94], 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. The work in [92] 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 EER=9.004% from the first dataset and EER=1.98% from the second dataset.

In addition, the research reported in [93] shows that app access patterns, as well as the traffic generated during app access, can be applied for user authentication with reasonable accuracy. Furthermore, the evaluation in these studies was based on classifying individual access events to apps. 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. Usage pattern profiling has been considered in many studies for many

purposes such as authentication and intrusion as well as fraud detection. For example, SMS and calling behavior are considered in [25][39][60][89]. Mobile device sensors have been considered for identifying users, including an approach in [95] that identifies and authenticates users based on accelerometer data. This approach considers contextual information as user activities, such as walking, climbing stairs and jogging. The authors in [96] devised a method for utilizing accelerometers in television remote controls in order to identify individuals. In [95], an approach that identifies and authenticates users based on accelerometer data is proposed. This approach considers contextual information as user activities, such as walking, climbing stairs and jogging. The achieved accuracy in this work is 72.2%, and the used dataset was generated by users repeating a limited set of pre-defined activities. Authenticating users of a smartphone according to accelerometer-based gait recognition, using the k-NN algorithm, is proposed in [22]. 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. An approach that utilizes mobile sensor data for human activity recognition is suggested in [97]. This approach processes accelerometer data on the cloud using three different classification algorithms: Naïve Bayes (NB) Classifier and KNN.

A further study presents an approach for authenticating a user by gesture recognition while holding a mobile device with an embedded accelerometer sensor [98]. The achieved results in the evaluation of this work include an ERR of 2.01% and 4.82% on a dataset of 100 users. However, these approaches require a particular series of actions to be authenticated. Furthermore, other studies such as [99] and [100], show that employing more sensors can improve authentication performance and accuracy. The main limitation of employing

sensor measurements for authentication, as presented in the previous studies, is that the majority of the related works are intended for user authentication on end-user devices with the assumption of dynamic movement of the user during interaction with the end-device. Therefore, it is difficult to establish authentication without proper data availability.

Another approach to modeling user behavior profiling involves utilizing the power consumption on the used device in order to characterize the user's behavior. For example, the authors in [101] demonstrate that the operation of a particular device driver leads to varying power consumption on the used device. A power estimation model, based on leveraged real user behavior, is presented in [102]. This study describes the high correlation between user behavior patterns and system power consumption patterns. A power model construction technique for monitoring the power consumption of each app on an electronic device is presented in [90]. 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.

Other approaches utilize behavioral traits such as gait, keystroke dynamics [103], gesture recognition and touch interactions for continuous user authentication. A user authentication method for mobile devices based on touchlogger is explained in [104]. This study shows that misuses or intrusions can be detected from touch events by using touchlogger. A different authentication technique with devices that use touch screen patterns is presented in [91]. Such devices are provided with touch screen capabilities to uncover input patterns. Based on the assumption that users perform predefined repetitive tasks, a study of touch screen behavior, as described in [91], 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.

A user authentication method that uses a gesture recognition algorithm is proposed in [20]. This model utilizes user touch screen interactions in addition to device feedback vibrations. Continuous authentication for smartphones based on smartphone gesture patterns is presented in [21]. The authors in [105] determine the possibility of developing new user authentication mechanisms by utilizing multi-touch gestures. A study of utilizing and analysing gait data is made in [106]. This study involved 11 volunteers, with an achieved accuracy of 79.1% and 92.7%, respectively. The gait characteristics of a user, based on six gait signature metrics according to the rate of changes of acceleration data, is presented in [107]. In [108] and [109], the authors described a method, also based on gait recognition, for determining whether the owner is using the device.

2.5.2 Combined Authentication Mechanisms

In order to enhance knowledge-based authentication credentials, many researchers have considered combining multiple methods. A method for user authentication that combines touch traces with pressure features on a mobile device is presented in [110]. A further user authentication method that combines different sensor data, for example, acceleration, keystrokes and touch interactions is proposed in [111]. For continuous identity verification in web services after initial knowledge login, a solution presented in [112] combines

keystroke dynamics with mouse dynamics for continuously identifying users. The results obtained in the evaluation of this work have an overall EER of 8.21%.

A research study reported in [113] shows that adding the features of size and pressure to keystroke dynamics can considerably enhance performance. A continuous user monitoring approach based on touch gestures, physical movement and power consumption is presented in [82]. This method was evaluated for the same device by 73 volunteer participants and obtained an EER with a range of 6.1% to 6.9% for 59 selected users. Combining both a keystroke and a handwriting method for user authentication via a screen sensor is developed in [23]. For authentication on a smartphone, using combinations of user behavioral pattern features of five basic user movements, such as sliding up, down, right, and left, is explained in [21]. A non-intrusive authentication method that employs orientation sensor data using the KNN classification is proposed in [114]. The authors further mention that combining multiple sensor inputs would improve accuracy. A new mobile system framework (SenSec) is introduced in [24]. In this framework, an accelerometer and gyroscope, along with magnetometer data, are continuously gathered during usage of the device in order to build a gesture model.

A lightweight, temporally and spatially aware user behavior model for authentication that is based on both hard and soft sensors is introduced in [115]. Four different attacks, including insider attacks, with the ability of detection in 717 seconds, are considered in this work. However, the authors did not quantitatively corroborate the accuracy of the model. The authors in [99] describe a continuous and transparent motion-based user authentication for smartphones, which utilizes orientation, accelerometer and magnetometer embedded sensors in the device. This proposed approach is constructed on user movement profiling,

using SVM as a binary class classifier for authentication. Although this study claimed a 90% accuracy, the used datasets were for the duration of five days and three weeks.

A continuous and implicit user authentication service is proposed in [116]. This approach is based on four sensors on a mobile system: voice, location, locomotion and multitouch. This method only visually shows that different users have different touch traces, without showing how to authenticate users using these traces. Research studies have widely considered that the combination of behavioral biometrics with knowledge-based and object-based authentication methods, such as passwords and tokens, can enhance authentication accuracy. In order to verify users, the authors of [117] propose combining both biometric techniques with gestural input on a multi-touch surface. This approach was able to achieve an accuracy of 90%. TouchIn, a two-factor authentication system that combines behavioral profiling with a password on multi-touch mobile devices, is presented in [118].

In order to improve the performance of keystroke dynamics-based authentication, a mechanism of combining the user's normal typing pattern with a password is proposed in [119]. The aim of this method is to increase the strength of the used password in comparison with passwords. For user authentication on mobile devices, the authors in [120] combine speaker identification and face recognition. A two-factor authentication system that combines touch gestures and graphical passwords on mobile phones is presented in [121].

A method that continuously authenticates the user's identity on a smartphone, based on finger movement and touch movement, is proposed in [21]. Seventy-five users participated in this experiment and, for verification, an SVM training model was created. In the evaluation results, the best FAR of 4% and FRR of 4% could be approximately achieved.

The authors in [71] integrate device fingerprint into a smart home for user authentication. However, this research does not consider continuous user authentication or behavior-based user authentication. The paper [122] presents a cloud-based monitoring framework to remotely monitoring smart homes through a surveillance camera; however, the solution does not consider continuous user authentication. The research in [123] proposes a security framework for smart devices in a smart home environment. The focus in this framework is only on securing communication between home devices; however, the presented solution does not consider user authentication. As can be observed from the potential solutions put forward by previous studies, most of the presented solutions are mainly designed for authenticating users on a dedicated device rather than for a remote accessing service. However, few studies proposed remote user authentication by utilizing a behavioral technique. For example, the authors in [124] suggested a new authentication framework that dissociates the intrusion detection system on the hosted device and enables the transparent movement of mechanisms between the host to the trusted server and vice versa. The proposed architecture can enable anomaly-based detection to be directly applied to the hosted device, or simultaneously to both the host and the cloud.

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 support smart home security in the form of continuous user authentication and identification at the server-side, the smart home central hub [125]. This concept is based on minimizing the load of the host device and offloading computation on the cloud. One advantage of this solution is that it will support multi-user behavioral models at the server-side, which

reduces resource consumption on mobile devices. This solution could be hosted either locally on the home hub or in the cloud; however, processing user profiles on the home smart hub offers advantages, including:

- Protecting user's information if the user's device is lost or stolen [126];
- Avoiding battery drain on the mobile device during preprocessing and training the model;
- Removing the need to rebuild the model in case the user changes mobile devices.

2.6 Machine Learning Techniques for Behavioral-Based User Authentication

Behavioral-based authentication encompasses data collection, feature extraction and classification. This process of utilizing machine learning techniques aims to extract/generate features that are related to the user and, in turn, utilize these features for authentication measures. An important step involves choosing a suitable classification approach to achieve the best accuracy [127]. Generally, machine learning techniques are classified into three types: supervised learning, unsupervised learning, and semi-supervised learning.

2.6.1 Supervised Learning

Supervised learning, which is one of the most popular machine learning techniques, is used when that dataset is labelled. Moreover, the supervised learning classification function maps the input to output labels or, in the regression function, maps the input to continuous output [127].

- Classification: Predictive models can be built using classification algorithms based on a method of approximating the mapping function from input features to discrete output

features. It is important to note that classification algorithms will be used when the desired output is a discrete label [128].

- Regression: Predictive models can be built using regression algorithms based on a method of approximating the mapping function from input features to a continuous output feature. A regression task is used to predict continuous outputs [128].

The following points summarize the widely used supervised learning classifiers:

- Random Forest (RF): The RF is an ensemble learning method that operates by constructing a multitude of decision trees at the training stage. Consequently, a Random Forest for each Decision Tree is built by randomly sampling a featured subset. The correlation between trees is reduced by randomly selecting features that improve the prediction power, resulting in higher efficiency. Random Forest classification algorithm offers advantages over other algorithms such as KNN, Naïve Bayes, Logistic Regression, and SVM. Some of these advantages include: overcoming the problem of overfitting; being less sensitive to outliers in training data; measuring the importance of every feature during model training [129]; rapidly performing out-of-sample predictions; providing accurate predictions on many types of applications [130]. Utilizing the Random Forest most often prevents overfitting, by creating random subsets of the features and building smaller trees using these subsets.
- Support Vector Machine (SVM): The SVM, which is a classification algorithm that can be utilized for regression and classification tasks, has been increasingly used to solve classification problems. The SVM classifier has been defined by separating a hyperplane from training samples [131]. The SVM is applied to the generated extraction vectors to produce an optimal hyperplane that categorizes new examples

based on given training data that will help separate two classes. Based on the maximum size of the margin, an optimized hyperplane can distinguish the class from other classes [128].

- Naïve Bayes (NB): The NB is a probabilistic classifier that has been implemented based on the Bayes' Theorem rules with independence assumptions between predictors. An NB algorithm assumes that the presence of a specific attribute of a class is unrelated to the presence of any other attribute. It is important to mention that all of these attributes independently contribute to the classification prediction, even if these attributes depend on each other or upon the existence of the other attributes. The NB classifier is easy to compute and build, as well as highly useful for handling very large datasets [128].
- K-Nearest Neighbor (KNN): As one of the classification algorithms, the KNN can be used as a supervised learning approach. The input of the KNN is a set of label showcases, whereby the labels are utilized for training the supervised classifier. The nearest neighbor computes the minimum Euclidean distance between the points. The KNN can label a new point by finding the nearest neighbors to that new point, hence giving greater weight to closer neighbors were the K is the number of neighbors. Based on past data, and labelling all observations, the KNN algorithm can identify a new data entry using the majority class according to the nearest neighbor [132].

2.7 Discussion

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. As discussed in [104], a touch screen authentication mechanism needs constant user interaction to be continuously authenticated. Many of the proposed solutions, such as those

presented in [20][21][22][23] and [24], require a particular series of actions from the user for authentication, including gesture recognition approaches which, in turn, eliminates the transparency of continuous authentication. Even though the achieved accuracy in other presented mechanisms is reasonable, such as demonstrated in [95], the used datasets were generated by users repeating a limited set of pre-defined activities for training, and usually only for short term usage. Additionally, it is clear from the related work that the home network authentication process still mainly relies on knowledge-based authentication approaches, as mentioned in [133][134][135]. The increasing number of apps that become available in the market at an almost daily rate provides users with the opportunity to install a variety of additional apps on their end-devices. Accordingly, it can be concluded that behavior profiling, especially app-based profiling, for the purpose of verifying the current user of an end-device, should be considered for continuous authentication.

For protecting user information during transmission among home IoT devices, many lightweight security schemes, such as in [62][63][66], have been proposed based on symmetric cryptography. These schemes involve sharing a number of symmetric keys for each party and, if the shared key is compromised, especially when PFS is not utilized, all communicated messages will be realized. For providing a higher level of security, public-key cryptography has been used. However, public-key cryptography still needs a certificate for each public key [26], which is considered a complex process, in addition to the involvement of a third party. Hence, any new proposed security scheme should satisfy the following requirements:

- any cryptographic solution should be less computationally intensive so as to be able to run on devices with limited capabilities;

- any involved device should be able to authenticate any transmitted message without involving a third party in each transaction and have the ability to protect against possible attacks.

There are many proposed public cryptographic mechanisms that can be adopted to secure communication between smart home IoT devices; however, not all of them efficiently satisfy all the security goals. Signcryption is a mechanism that provides digital signatures and encryption, at the same time satisfying both authentication and integrity which, in turn, enables its adoption for smart home devices. In addition, the IBE [136] can satisfy the security requirements in IoT device communication.

The consideration of continuous user authentication beyond the point-of-entry is crucial and should be considered as part of authentication. However, according to the literature, only a few studies employ app profiling for continuous user authentication. Although the achieved results in such studies are prominent, the range of apps used in the evaluation was restricted due to the limitations of the datasets. In addition, these studies, as in [25], focus on app-specific information, such as SMSs and calling behavior, for detecting abnormal usage in the mobile environment. With the increased number of apps replacing text messages and calling functions, these two features will no longer provide sufficient evidence of a legitimate user. Furthermore, considering that unauthorized users tend to operate inconspicuously, these features will provide insufficient evidence of the user. The focus of the earliest studies was mainly on detecting misuse behavior during interaction with the mobile network, such as calling and messaging services. However, none of the related works have utilized app access patterns on mobile devices for user authentication for smart home networks.

2.8 Summary

As seen in the literature, access behavior has been used in many technologies, especially at the client-side, for implicit user authentication to protect against unauthorized access to mobile phones. Hence, app access patterns can be utilized to support smart home security in the form of background user authentication at the server-side, namely the smart home central hub. Furthermore, a detailed review of the works listed in the related literature suggests that multi-user authentication 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 approach for smart home networks. Moreover, it is clear from the literature that the point-of-entry authentication approach has been developed in order to determine permission to access the device itself and that it provides no further protection during the usage session. However, in reality, the need for security will vary depending upon what the user is doing, and different services and data should require different levels of security. As a result, an advanced authentication approach, which is capable of continuously monitoring and authenticating a user based upon the user's legitimacy, is needed. This can be achieved by using an implicit authentication mechanism. Consequently, users will not be aware that authentication is taking place, avoiding the need to stop operation, for example, in order to re-enter a PIN.

Many proposed signcryption schemes that are based on RSA cannot efficiently provide all the security goals. However, a suitable cryptographic technique that could solve security issues in device communication for smart homes is the IBS. A detailed review of the current literature suggests that no signcryption scheme has yet been proposed for securing smart home communications, which is the primary focus of this thesis. Additionally, none of the

related works have utilized app access patterns on mobile devices for user authentication for smart home networks. This thesis solves the aforementioned issues by proposing a contextual authentication framework that considers device authentication utilizing IBE as well as user authentication based on app interactions.

Chapter 3. Proposed Solution

This chapter presents a contextual authentication framework for smart home environments. This framework consists of three main parts: the context-based user authentication method, device-to-device message authentication, and app-based user authentication model. The first part is a context-based authentication method that utilizes contextual information to verify the current user who is trying to access smart home IoT devices from his/her registered end-device (smartphone/tablet). The second part is a device-to-device message authentication scheme to protect transmitted messages among smart home IoT devices. The third part is a user authentication model that builds a user profile based on previous apps' access history in order to make the right decision at the login stage, at subsequent access requests regarding authorized user access and during the access sessions. This framework can be adopted in different IoT applications, such as the smart home, smart medical care [137], smart agriculture, smart grid and smart environmental protection [138][139]. This chapter presents a detailed description of the proposed framework, a contextual information taxonomy and collection, a context-based user authentication method, a device-to-device message authentication scheme, and an app-based user authentication model.

3.1 Assumptions and Threat Model

The scope of this research is restricted to the user and device-to-device message authentication for smart homes. The main goal of the proposed framework is to protect against unauthorized access while permitting implicit authentication to authorized users. In addition, it provides secure communication between smart home IoT devices. The presented framework in this thesis is subject to the following assumptions:

- Registered users, mobile devices, and the added trusted server (TS) are assumed to be trusted. After registration, there is an app-based authentication model building and training stage, during which and authentication will be provided by other means. Additionally, every user has one profile and one assigned account in the home network.
- 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.
- All mobile devices/tablets are uniquely identified; the operating systems as well as installed apps, including the smart home user interface, are secure. In addition, the registration of the device, including end-user device and home IoT devices, is achieved through a local channel directly between the home devices, the mobile device and the trusted server.
- The home gateway (HG), the main connection point between the smart home IoT devices, the TS and the Internet, is assumed to be vulnerable to attacks. Thus, any transmitted message could be exploited by an adversary residing in the HG, which could disclose user information.

The proposed framework aims to protect against threat scenarios for both user authentication and device authentication. Accessing the smart home network and controlling home appliances is mainly achieved through registered mobile devices by known users. However, access permissions can be given to other users, such as visitors and friends, who will be able to access the home network using their mobile devices. Accordingly, for user authentication, there are security points where unauthorized access to the home networks could occur:

- The user is logged into the mobile device, and the device is left unattended immediately after the login stage, yet it is being used by unauthorized users (insiders). An insider, as mentioned in [140], can be a visitor or another registered user and familiar with other users.
- The mobile device, on which user's device login credentials are stolen, compromised, or given, is lost or stolen by unknown users (outsiders, strangers [39]), causing these devices to be vulnerable to unauthorized access and usage.
- Thus, at any access request, the authentication process will result in three possibilities:
    - Registered user request access from his/her registered mobile device;
    - Registered user request access from another registered mobile device (insider);
    - Unregistered user request access from a registered mobile device (outsider).

Transmitted messages between home IoT devices could be exploited by an adversary residing in the HG. Consequently, message authentication should be performed by each entity involved in a smart home network to verify the source of any issued message. Hence, the following scenarios are examples of possible threats include:

- An adversary residing in a HG might launch an active attack [141] to target data integrity, such as modifying or altering a command during its transmission.
- An adversary might also, as a result of being able to eavesdrop on a message transmission, discover a command message. For example, if an attacker is able to reach a message command sent to a lightening system to turn the light on or, an adversary would know whether the user is at home, which could result in the home being broken into, impacting confidentiality. Thus, any important information involved in the transmitted messages among devices must be inaccessible to unauthorized parties.

3.2 Framework Architecture

The architecture of the proposed framework consists of user authentication mechanisms as well as data protection mechanism, as depicted in Figure 3.1. In this thesis, we consider a smart home system model which consists of four main parties: a number of home devices; an end-user device; a home gateway; and a trusted server (TS). The end-user device, which could be any smart device such as a smartphone or a tablet, is used as a node in the smart home model. Furthermore, we consider the communication process as any message exchanged between home devices.

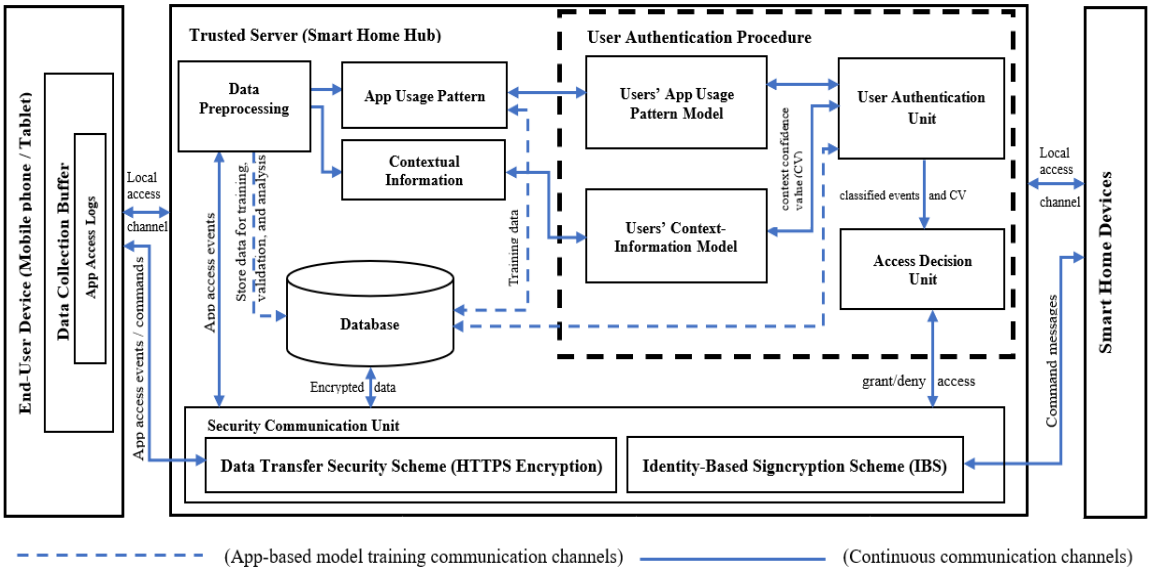

Figure 3.1. A contextual authentication framework for smart home environments

In the proposed framework, different contextual attributes and properties can be utilized to enhance the determination of a legitimate user. This can be achieved by capturing long-term contextual information (e.g., app access patterns, power consumption); short-term contextual information (e.g., location, profile); and sensor data (e.g., pressure, temperature). Additionally, it is important that the collected contextual information is transmitted to and stored securely on the TS, and has backup storage, an external hard disk

drive, in case of data loss, so models and profiles do not need to be retrained. The data is protected in transit between the end-user devices and the TS through the use of HTTPS encryption. Also, data transit between the TS and the home IoT devices is protected through the IBS scheme and the data is stored in an encrypted form. The terminology used in the rest of this thesis as well as the major components upon which the framework is built are as follows:

- Home Devices: Home devices are any devices (e.g., surveillance camera, media server, smart lock, refrigerator, or baby monitor) that can be remotely accessed. Most of these devices communicate locally over wireless channels through the local home gateway.
- End-User Devices: End-user devices are any devices (e.g., smartphone or tablet) that can be used to access home devices, through an application programming interface (API) installed during the registration stage. This API can collect usage data and send it to the home TS for user behavior analysis. Every access request or command will be accompanied with context information measurements for user pattern analysis and evaluation.
- Home Gateway (HG): The home gateway is a network entity (router) that acts as an intermediary between home devices and end-user devices. The primary role of the HG is to help in exchanging messages locally among devices through either the WiFi or the Internet.
- Trusted Server (TS): We integrate a trusted server that is responsible for initializing the system, registering user, building usage pattern profiles, and registering new devices and assigning required secret communication parameters. The TS contains a database that records all the registered devices and the access log history of users as well as

performing user access patterns modeling. It is responsible for the authentication process and protects access to these devices. The TS collects the required contextual information and determines whether the access request satisfies the predefined requirement.

3.2.1 Contextual Information

Table 3.1 shows what contextual information would be available and how it would be collected. The vast majority of these contextual attributes could be used when the user is accessing the system remotely, with some environmental information able to be collected with trusted sensors installed at a remote location including, for example, by using a Bluetooth module to check if the user's device is nearby. Some of the contextual information can be collected and maintained solely by the TS and normal responses from user devices, while others would require extra data from the user's device itself. Lastly, while most of this data can be collected in the background, some other contextual information, such as the user's profile information, security questions [142], and calendar, would require specific interaction with the system.

3.2.2 Device-to-Device Message Authentication

From the initial connection of devices in a smart home network, the following security requirements should be satisfied in order to prevent attacks:

- Authentication: Authentication should take place for any transmitted message to verify its source. The system can then prevent any unauthorized access to devices.
- Integrity: Integrity ensures that a transmitted message is not altered or generated by an attacker during its transmission.

- Confidentiality: Confidentiality should be satisfied to protect any disclosure of the transmitted messages among devices which may contain sensitive information that could be exploited by an adversary. Protecting any disclosure of the information related to home devices would also prevent users' information being revealed.
- The proposed scheme should be lightweight, meaning lower cost in regard to computation time and cipher-text length.

3.2.3 User Authentication

The presented approach aims authenticating users with low FPRs and FNRs; utilizing implicit features that can be extracted, without requiring user intervention in the authentication process; ensuring that the utilized features are generalized, hence can be extracted from most mobile devices regardless of the operating systems on these devices or the type of hardware; and protecting collected information on mobile devices during transmission and at the TS 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 of users; and adapting the change in user patterns, including new added apps. This study examines the available contextual information in regard to the permission requirements and retrieval time to be used for authentication. Finally, it provides user authentication by checking contextual information in real-time during the access session without user intervention unless the situation requires it.

Table 3.1. Contextual information

Context Type	Feature	Data Collected by	Available Remotely	Requires App or External API	Requires User Intervention
User Context	Location	Device	Yes	No	No
	Access patterns (logs)	TS	Yes	No	No
	Profile	Device	Yes	No	Yes
	Calendar	Device	Yes	Yes	Yes
Device Context	Location (Bluetooth)	TS	Possible	No	No
	Operating System	TS	Yes	No	No
	Browser	TS	Yes	No	No
	Voltage value	Device	Yes	Yes	No
	Wi-Fi access points	Device	Yes	Yes	No
	Used applications	Device	Yes	Yes	No
	Battery level	Device	Yes	Yes	No
	MAC address	TS	Yes	No	No
	Motion detection	Device	Yes	Yes	No
	Rotation detection	Device	Yes	Yes	No
	Compass (environment detection)	Device	Yes	Yes	No
Network Context	IP address	TS	Yes	No	No
	Connection type	Device	Yes	Yes	No
	Ping	TS	Yes	No	No
	Speed	Device	Yes	Yes	No
	Traceroute	TS	Yes	No	No
Environmental Context	Lighting	Device	Possible	Yes	No
	Temperature	Both	Possible	Yes	No
	Pressure	Both	Possible	Yes	No
	Humidity	Both	Possible	Yes	No
	Loudness	Device	Possible	Yes	No

3.3 Device-to-Device Message Authentication Scheme

This section presents an Identity-Based Signcryption scheme (IBS) for device-to-device message authentication in smart homes. This section first presents the preliminaries that are used in this thesis, and then, it gives an overview of the proposed scheme, which consists of five phases: system initialization, registration, signcryption, unsigncryption and correctness.

3.3.1 Preliminaries

This section presents some preliminaries that are used in this thesis. Table 3.2 shows the used notations throughout the thesis.

3.3.1.1 Elliptic Curve Cryptography

This subsection presents elementary concepts, including the group theory and elliptic curve groups.

An *abelian group* (G1,$*$) consists of a set G1 with a binary operation $* : G1 \times G1 \rightarrow G1$ satisfying the following properties [143]:

- Associativity: $a * (b * c) = (a * b) * c$ for all $a, b, c \in G1$.
- *Existence of an identity:* There exists an element $e \in G1$ such that $a * e = e * a = a$ for all $a \in G1$.
- Existence of inverses*:* For each $a \in G1$, there exists an element $b \in G1$, called the *inverse* of a, such that $a * b = b * a = e$.
- Commutativity: $a * b = b * a$ for all $a, b \in G1$.

The *additive group*, the (additive) identity element, is usually denoted by 0, and the (*additive*) inverse of *a* is denoted by (-a). The *point addition* operation is performed by

drawing a line through two points (e.g., *P*, *Q*), and this line will intersect the curve *E* at one more point *R*. Then by drawing a line parallel to the y-axis through point *R*, the line will intersect the curve E at the new point R=P+Q. The *inverse* of a point *P*= (*x*, *y*) on the curve *E* is the point mirrored at the x-axis in -*P*= (-*x*, *y*). If the point is the same P at the additive operation, the tangent on the curve at the point *P* (*P*+*P*=*2P*) is known as point doubling. The *multiplicative group*, the (*multiplicative*) identity element, is usually denoted by 1, and the (*multiplicative*) inverse of *a* is denoted by (a^{-1}). The group is *finite* if G1 is a finite set, in which case the number of elements in G1 is called the *order* of G1. If *p* is a prime number, and Z_p ={0,1,2, . . ., *p*-1} denotes the set of integers modulo *p*, then $(Z_p,+)$, where the operation + is defined as the addition of integers modulo *p*, is a finite additive group of order *p* with (*additive*) identity element 0. Additionally, $(Z_p^*, \cdot)$, where Z_p^*= {1,2, . . ., *p*-1} *denotes* the nonzero elements in Z_p and the operation (·) is defined as multiplication of integers modulo *p*. Z_p^* is a finite multiplicative group of order *p*-1 with (*multiplicative*) identity element 1 [143]. For example, *p is* a prime number and Z_p denotes the field of integers modulo *p*. An *elliptic curve E* over Z_p is defined by an equation of the form

$$y^2 = x^3 + ax + b \qquad (3.1)$$

where *a*, $b \in Z_p$ satisfy $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$. A pair (*x, y*), where *x, y* $\in Z_p$, is a *point* on the curve if (*x, y*) satisfies the equation (3.1) defining the curve *E*. The *point at infinity*, denoted by ∞, is also said to be on the curve. The set of all the points on *E* is denoted by $E(Z_p)$ [143]. For example, if *E* is an elliptic curve over Z_7 with defining equation

$$y^2 = x^3 + 2x + 4b \qquad (3.2)$$

then the points on *E* are *E* (Z_7) = {∞, (0,2), (0,5), (1,0), (2,3), (2,4), (3,3), (3,4), (6,1), (6,6)}. The nonzero elements of a finite field Z_p , denoted Z_p^* , form a cyclic group under multiplication. Hence there exist elements $b \in Z_p^*$ called generators such that

$$Z_p^* = \{b^i : 0 \leq i \leq p-2\} \qquad (3.3)$$

The order of a $\in Z_p^*$ is the smallest positive integer *i* such that $a^i = 1$. As Z_p^* is a cyclic group, it follows that *i* is a divisor of *p* −1.

3.3.1.2 Bilinear Pairing on Elliptic Curves

Let G_1 and G_2 be two cyclic groups of order *p* for some large prime *p*, and $P_1 \in G_1$ be the generator of G_1. The IBE makes use of a bilinear pairing map ê: G_1 x $G_1 \rightarrow G_2$ between these two cyclic groups. A bilinear parameter generator gen is a probabilistic algorithm that takes a security parameter *sp* as input, and outputs the parameters (*p*, P_1, G_1, G_2, ê) where *p* is a k-bit prime number. The map must satisfy the following properties [144]:

- *Bilinear*: ê (*aP*1, *b*Q) = ê$(P_1, Q)^{ab}$ for all P_1, Q ∈ G_1 and all *a*, *b* ∈ Z_p^*.
- *Nondegenerate*: If P_1 is a generator of G_1, then ê (P_1,P_1) is a generator of G_2, thus ê (P_1,P_1) ≠ 1.
- *Computable*: There is an efficient algorithm to compute ê (P_1, Q) for all P_1, Q ∈ G_1.

A bilinear map that satisfies the properties above is said to be an admissible bilinear pairing map ê: G1 x G1 → G2. The Weil pairing or Tate pairing can be used to construct an admissible bilinear map between these two groups over elliptic curves [144].

Table 3.2. Definitions of notations

Notation	Definition
TS	trusted server
HG	Home Gateway (router)
ID_{TS}	identity of the *TS*
Q_{TS}	public identity of the *TS*
S_{TS}	private key of the T*S*
ID_{di}	identity of device *i*
Q_{di}	public identity of device *i*
S_{di}	private key of a device *i*
sp	security parameter
s	master secret key of the *TS*: $s \in Z_p^*$
p	k bit prime number
ê	admissible bilinear parameter
$H_1,\ H_2$	secure cryptographic hash functions
Z_p^*	set of elements {1, ..., $p-1$}
P_1	the generator of *G*1
G_1	a subgroup of additive group of points
G_2	a subgroup of multiplicative group of points
r_i	a random integer: $r_i \in Z_p^*$
m	(message) plaintext
n	plaintext length
C_{enc}	cipher-text
σ	the signcrypted message
m’	the unsigncrypted message

3.3.1.3 Complexity Assumptions

The proposed scheme is based on two Diffie-Hellman problems as presented below.

- *Assumption* 1: (Computational Diffie-Hellman problem (*CDH*)). Given the elements $(P_1, aP_1, bP_1) \in G_1$ for unknown $a, b \in Z_p^*$, there is no polynomial time to compute $abP_1 \in G_1$.
- *Assumption* 2: (Bilinear Diffie-Hellman problem (*BDH*)). Given the elements $(P_1, aP_1, bP_1, cP_1) \in G_1$ for unknown $a, b, c \in Z_p^*$, it is difficult to compute $\hat{e}(P_1, P_1)^{abc} \in G_2$.

3.3.1.4 Identity-Based Signcryption (IBS)

IBS includes four steps: system initialization, registration, signcryption and unsigncryption. The function of each step is outlined below.

- System initialization: The T*S* uses a security parameter *sp* to generate the public system parameters while keeping *sp* parameter secret.
- Registration and private key generation: Given an identity of a device ID_{id}, the *TS* computes the corresponding private key S_{di} and sends it back to the device.
- Signcryption: To send a message *m* to a receiver with identity ID_{id}, the sender encrypts the message *m* and then signs it consecutively using the public parameters resulting from the system initialization stage, and the private key S_{di} of the sender and ID_{id} of the receiver, and the message *m* producing a cipher-text message.
- Unsigncryption: When the receiver receives the cipher-text message from the sender device ID_{id}, it unsigncrypts it (using the sender's ID_{id}, the receiver's S_{di}) to obtain the corresponding plaintext *m* after checking the correctness of the sender.

3.3.2 Proposed IBS Scheme

This section presents an overview of the proposed IBS scheme, which consists of four phases: system initialization, registration, signcryption, unsigncryption. The steps are described below. Algorithm 3.1 shows the process of scheme initialization and registration procedure, whereas algorithm 3.2 shows the scheme signcryption and unsigncryption procedure.

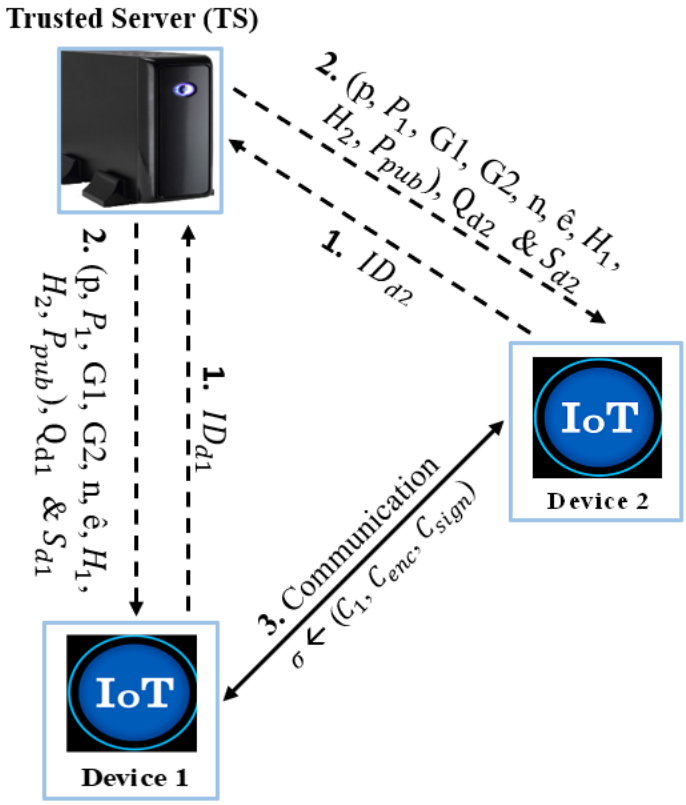

Figure 3.2. Operations of the proposed IBS scheme

3.3.2.1 System Initialization

In this stage, a T*S* is responsible for configuring system parameters. In particular, using the security parameter *sp* as input, the *TS* generates the bilinear parameters (p, P_1, G_1, G_2, ê) by running gen(k), and chooses two secure cryptographic hash functions: H_1: $\{0,1\}^* \rightarrow G_1$ and H_2: $G_2 \rightarrow \{0,1\}^n$ (n : *plain text length*). The *TS* chooses a random secret number $s \in$

Z_p^*, which is kept as a master secret, and then calculates $P_{pub} = sP_1$. The *public parameters*: (p, P_1, G_1, G_2, n, ê, H_1, H_2, P_{pub}) are then published by the T*S*.

3.3.2.2 Registration Stage

At this stage, as shown in Figure 3.2, each device submits its chosen identity ID_{di} to the *TS* which, in turn, generates the public identity Q_{di} and calculates the private key S_{di} using the master secret key *s*. For the sender device, the *TS* calculates $Q_{d1} = H_1(ID_{d1})$ and $S_{d1} = s(Q_{d1})$, then sends the public identity and the private key to the sender device over a secure channel. Similarly, the second (receiver) device sends the chosen identity ID_{d2} to the *TS* and receives the public identity and the private key consecutively: $Q_{d2} = H_1\ (ID_{d2})$ and the private key is $S_{d2} = sQ_{d2}$. The *TS*'s identity is $Q_{LS}. = H_1\ (ID_{TS})$ and the private key is $S_{TS} = sQ_{TS}$.

After the registration stage, there is no need for communication with the *TS* during the authentication process of any transmitted messages. Access to the *TS* is only required at the time of registration or for updating secret keys.

3.3.2.3 Signcryption

We assume that the sender device is d_1 and the receiver device is d_2. Thus, the sender device now has the following parameters: (S_{d1}, Q_{d2}, m). In order to signcrypt a message $m \in \{0,1\}^n$, the proposed signcryption technique works as follows: it encrypts the message and then signs it consecutively using the public parameters resulting from the initialization stage as well as (S_{d1}, Q_{d2}, m). The actions of the sender device are described below.

Generates a random integer $r_i \in Z_p^*$ and then calculates the following:

$C_1 = r_i P_1$; Q_{d1} & Q_{d2} are already initialized in the registration stage.

k = $H_2(\hat{e}\ (r_i Q_{d2}, P_{pub}))$

C_{enc} = (m $\oplus$ k), where C_{enc} is the cipher-text.

H= $H_1\ (C_{enc})$

$C_{sign} = r_i H + S_{d1}$, where S_{d1} is the sender's private key.

Result σ = (C_1, C_{enc}, C_{sign}) is the signcrypted message *m* by the sender device S_{d1}. The σ is then sent to the targeted receiver, device d_2.

Algorithm 3.1: Scheme initialization and registration procedure

---

INPUT: ID_{di} *and* k ▷ where ID_{di} is the identity of the device *i* and *sp* is the security parameter

OUTPUT: the public parameters **(***p*, P_1, G_1, G_2, *n,* ê, H_1, H_2, P_{pub}, Q_{di}, S_{di}, Q_{TS}, S_{TS}, *s*)

▷ these parameters include bilinear parameters, two secure cryptographic hash functions, the public identity, the private key, The *TS*'s identity and private key, and a master secret key s.

1. **procedure** ()
2. **if (**initialization stage) **then**
3. (*p, P_1, G_1, G_2, ê*) ← run gen(*sp*)
   H_1: $\{0,1\}^* \rightarrow G_1$ ▷ where H_1 *and* H_2 are cryptographic hash functions
4. H_2: $G_2 \rightarrow \{0,1\}^n$ ▷ where n is the plain text length
   $s \leftarrow Z_p^*$ ▷ where s is a chosen the number (master secret key) $\in Z_p^*$
5. sP_1← calculates (P_{pub})
6. return (*p, P_1, G_1, G_2, n, ê, H_1, H_2, P_{pub}*) ▷ where (*p, P_1, G_1, G_2, n, ê, H_1, H_2, P_{pub}*) are the public parameters to be published by the *TS*
7. **else (**registration stage) **then**
8. *TS*← ID_{di}
9. $Q_{di}, S_{di}, s \leftarrow$ *TS* generates
10. Q_{di}, S_{di} ← *TS* calculates $H_1(ID_{di})$, $s(Q_{di})$
11. Q_{TS}, S_{TS}← *TS*'s public and private key
12. return ($Q_{di}, S_{di}, Q_{TS}, S_{TS}$)
13. **end if**
14. **end procedure**

Algorithm 3.2: Scheme signcryption and unsigncryption procedure

Sender device is d_1 and the receiver device is d_2
INPUT: $m \in \{0,1\}^n$ the message to be signcrypted
OUTPUT: σ the signcrypted message, m' the unsigncrypted message

1. **procedure** ()
2. **if (**signcryption) **then**
3. **(**signcrypt a message $m \in \{0,1\}^n$**)**
4. $r_i \leftarrow Z_p^*$ ▷ the sender device d_1 generates a random integer $r_i \in Z_p^*$
5. $C_1 \leftarrow r_i P_1$ ▷ Q_{d1} & Q_{d2} are already initialized in the registration stage.
   $k \leftarrow H_2(\hat{e}\,(r_i Q_{d2}, P_{pub}))$
6. $C_{enc} \leftarrow (m \oplus k)$ ▷ where $C_{enc}$is the cipher-text
7. $H \leftarrow H_1\,(C_{enc})$
8. $C_{sign} \leftarrow r_i H + S_{d1}$ ▷ where S_{d1} is the sender's private key.
9. $\sigma \leftarrow (C_1, C_{enc}, C_{sign})$ ▷ where σ is the signcrypted message m by the d_1
10. return (σ) ▷ σ is then sent to the targeted receiver, device d_2.
11. **else (**unsigncryption) **then**
12. **if** (ê $(P_1, C_{sign}) =$ ê(h, C_1) ê(Q_{d1}, P_{pub})) holds **then**
    ▷ the received message is verified
13. $k' \leftarrow H_2\,(\hat{e}\,(S_{d2}, C_1))$
14. $m' \leftarrow C_{enc} \oplus k'$
15. return (m') ▷ where m' is the unsigncrypted message
16. **else**
17. return(reject the message)
18. **end if**
19. **end if**
20. **end procedure**

3.3.2.4 Unsigncryption

After receiving the signcrypted message, d_2 first verifies the received message by running the following equations:

$$\hat{e}\,(P_1, C_{sign}) = \hat{e}(h, C_1)\,\hat{e}(Q_{d1}, P_{pub}) \tag{3.4}$$

If equation (3.4) holds, the next step is unsigncryption, meaning that the received message m is authenticated. The unsigncryption algorithm works as follows:

$k' = H_2\,(\hat{e}\,(S_{d2},\, C_1))$

$m' = C_{enc} \oplus k'$

Otherwise, it is invalid, and the message will be rejected.

3.4 App-Based User Authentication Model

This section presents the app-based user authentication model. The proposed model, as shown in Figure 3.3, authenticates users based on the user profile built from previous app access history, then makes the decision for subsequent access requests regarding legitimate user authentication and identification. The presented model utilizes the app access patterns on users' mobile devices as a second layer of authentication. The architecture of the presented model, as shown in Figure 3.3, works, after the registration step, by first collecting user access logs, extracting features, and training user access pattern to apps on mobile devices, then authenticating users based on the built pattern templates. After the user's registration stage, the model tracks and collects app access events adopting the event-driven mechanism.

Utilizing the event-driven data collection approach as compared with continuous sensor measurements 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 TS for the training, testing, authentication phases. After building and training the model, installed app on the mobile device will only send the events of the considered apps for user authentication and identification purposes [145]. Algorithm 3.3 shows the process of building the app usage pattern-based user authentication model.

3.4.1 App Categories

In terms of running on mobile devices, there are two app categories: foreground apps and background apps. Foreground apps are those that run and are actively used by the user. In other words, foreground apps require continuous user interaction during the running time. Background apps are those that run without the necessity of user interaction. In other words, background apps 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. Thus, the possible solution is to use those apps that continuously reflect usage behavior, namely foreground apps. In addition, since app usage data will be already available as a result of users' usage on their mobile devices and tracking these apps, real-time user authentication and identification can be achieved with high accuracy. Hence, different from previous related studies, we mainly focus on foreground apps.

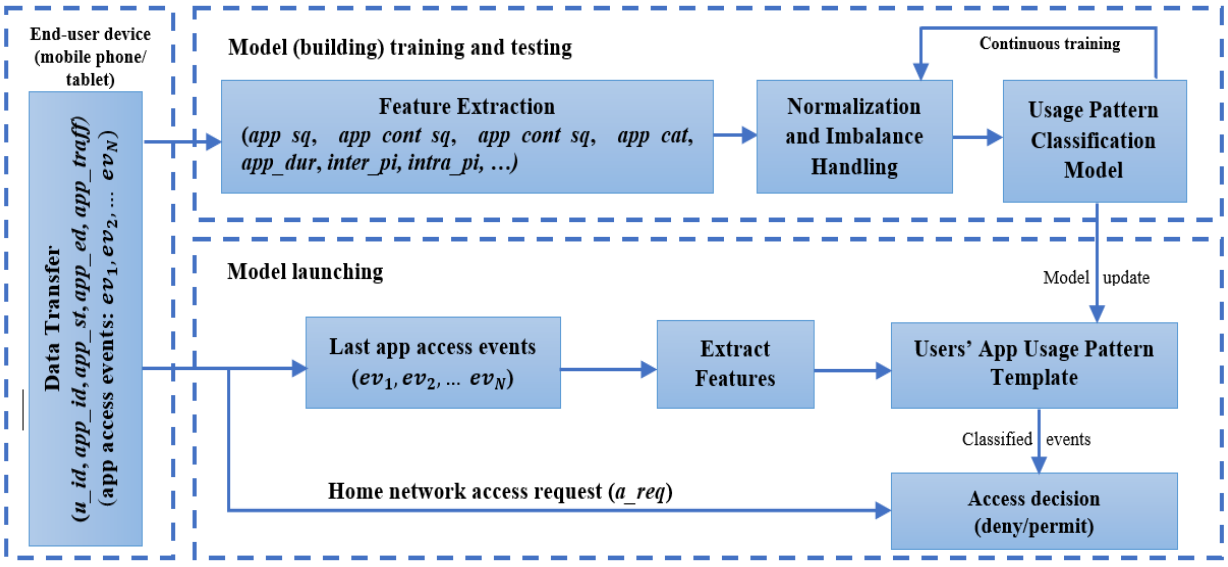

Figure 3.3. Architecture of the proposed app-based user authentication model

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.

Algorithm 3.3: App-Based User Authentication Model Building

INPUT: Dataset $\boldsymbol{D}$
OUTPUT: user's classification model $\boldsymbol{CM_i}$, $\boldsymbol{V}$ is the extracted and assigned threshold

1. **procedure** ()
2. input ← $\boldsymbol{D}$
3. read features ← $\{\boldsymbol{f_{x1}}, \ldots, \boldsymbol{f_{xn}}\}$: features of access time ($\boldsymbol{D^{time}}$) & network traffic ($\boldsymbol{D^{traff}}$)
4. $\boldsymbol{App_{ev}^{fgr}}$ ← select only foreground-based events & normalize features
5. **for user** ($\boldsymbol{u_i}$) ← 1 to *N* **do** ▷ where *N* is the number of users
6. do {
7. $\boldsymbol{D_i^{sample}}$ ← **1 to** $\boldsymbol{n}$ **do** ▷ where *n* is the number of samples
8. extract new features ← $\{\boldsymbol{f_1}, \ldots, \boldsymbol{f_z}\}$ ▷ where *z* is the number of features to be extracted
9. split $\boldsymbol{D}$ into $\boldsymbol{D_i^{train}}$ and $\boldsymbol{D_i^{test}}$
10. randomly split $\boldsymbol{D_i^{train}}$ into $\boldsymbol{k}$ subsets $\{\boldsymbol{D_1}, \ldots \boldsymbol{D_k}\}$ ▷ where *k* is number of folds
11. up-sample the $\boldsymbol{D_{min}^{train}}$ of the minority class ($\boldsymbol{D_{min}^{up\_samp}}$)
12. build the model ($\boldsymbol{M_i}$) using both $\boldsymbol{D_{min}^{up\_samp}}$ and majority Class $\boldsymbol{D_{maj}}$
13. test the $\boldsymbol{M_i}$ on $\boldsymbol{D_i^{test}}$
14. calculate ***FPR***, ***FNR*** and ***EER*** ▷ where FPR is false positive rate, FNR is the false negative rate, and EER is the equal error rate
15. } **while** (***FPR***, ***FNR***, *and* ***EER<5%)***
16. $\boldsymbol{V}$ ← set threshold, number of access events per app
17. set the threshold for app access ← $\boldsymbol{V}$
18. $\boldsymbol{CM_i}$ ← launch
19. **end for**
20. **end procedure**

3.4.2 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 networks, WiFi networks or local apps that do not need access to networks

in order to run. 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. The two modes of data collection procedures are the training and testing mode and the authentication mode.

1) 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*).

2) Data Collection for User Authentication

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, newly accessed (added) apps are also considered. For example, if a new app is launched, 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 low FPT and FNT.

3.4.3 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 [146], 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 thesis is on the implicit features. Thus, the collected information will subsequently be preprocessed and stored at the *TS*.

Feature extraction is an important step where unique features are extracted from the collected information. The features that can be retrieved from app access logs on mobile devices include *app_id, u_id; app_st; app_ed;* generated traffic (*app_traff*) while accessing this app. In order to build an authentication model, the literature presents approaches that use a specific app to discriminate between users, but our goal is to utilize features to authenticate 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 building the usage pattern 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 thesis, we consider both as we include all accessed apps to model user behavior. 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.

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-1} and a_i) to the same app on the same day. This interval is calculated as $a_i - a_{i-1}$ for all apps accessed on the same day. The second feature, the *intra_pi*, includes the interval between any two consecutive general accesses (a_i and b_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. 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 generated, 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 such as every five minutes or even more often. Rather, the proposed approach requires sequentially accessed apps whenever an app is used, and this access is measured as event-driven access. In addition, we consider the order of access to apps in everyday use as well as app continuous sequence order, which will enhance discrimination between users. When the access log is received from a user's mobile device, it is used to generate the required features at the home TS, including: session access time (*app_st*, and *app_et*); extracting days of the week from the time stamps (weekday and weekend names (*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 log, which includes the *u_id*, the *app_id*, the *app_st*, and the *app_ed,* will be transferred to event information that includes the new generated features. Additionally, the timestamp is mapped into one of the six-time intervals (*inter_1*, *inter_2*, *inter_3*, *inter_4*, *inter_5*, and

inter_6) as well as one of the three weekday intervals as follows: (*app_st* -> *inter_i*, *w_inter_i*). The generated 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.

3.4.4 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, a binary classification strategy is adopted. 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 values. 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 [147] 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 known 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, the next usage samples. 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 classifiers (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, when the events are not classified for the targeted registered user, the authentication 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 RF classifier as it is widely used in many applications such as banking, medicine, the stock market and e-commerce. In addition, it has evidenced high accuracy in previous studies [94][125][148]. The RF is a supervised classification algorithm which creates a forest with a number of trees. Even though other classifiers, such as the SVM, have been used in the literature, it requires more computation time and produces less accuracy. In addition, we select the parameter values for the RF that minimalize the FPR and FNR as much as possible for all users.

3.4.5 User Authentication Unit

Our proposed method aims is to build an authentication model based on legitimate user access patterns. This step requires 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. Three important aspects, as mentioned at the threat model, should be considered while tracking user access to smart home networks: registered user from his/her registered device (legitimate user); registered user from another registered device (insider), and unregistered user from a registered device (outsider). Hence, 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.

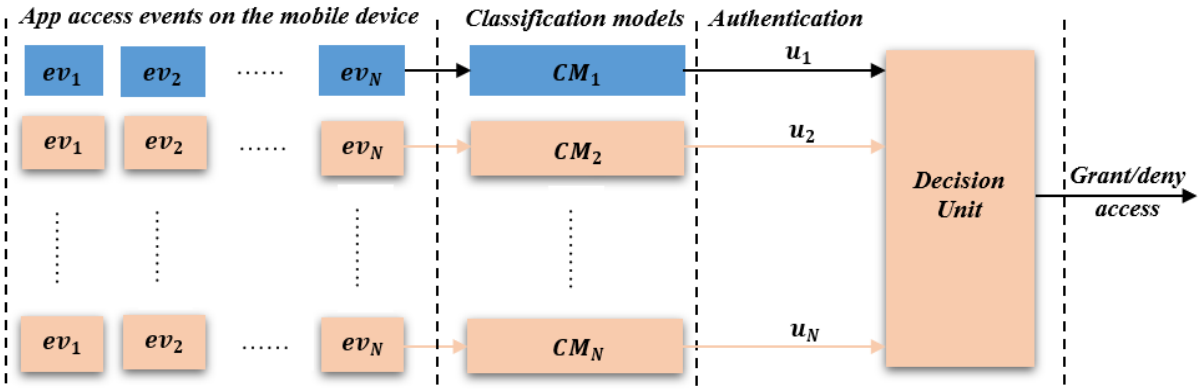

Figure 3.4. User authentication procedure

In contrast, for insiders, user authentication 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 if the pattern is not classified for the registered user, *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

3.4. 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 other authentication procedures in order to detect whether this user is one of the home members (insider) or not (outsider). If the user is classified as one of the known users, access permission will be given based on the permission that was set at the registration stage by the administrator.

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 handles it as follows: if this app is the last in the last sequence of accessed apps, it invokes the user for a second-factor authentication, 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.

3.4.6 Decision Unit

Classifying each access event received from the user's mobile device may include 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 3.5, immediately before the access request sent from the user.

$$D_{\mathrm{i}} = \begin{cases} if\ (a_{l-1} and\ a_l) \in\ u_i,\ Permit\ access \\ if\ (a_{l-1} \in\ u_i)\ and\ (a_l \notin\ u_i),\ Deny\ access \\ if\ (a_{l-1} \notin\ u_i)\ and\ (a_l \in\ u_i),\ Deny\ access \\ if\ (a_{l-1} \notin\ u_i)\ and\ (a_l \notin\ u_i),\ Deny\ access \end{cases} \tag{3.5}$$

At this unit, the decision (D_i) will be made based on the last two accessed events ($a_{l-1} and\ a_l$). Consequently, when the last two classified events are identified for a specific user (u_i), the next access request will be accepted. 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. If the last two accessed events are not classified to the legitimate user, the decision will be made based on the majority of the rest of the classification models, based on the formula 3.6.

$$D_{\mathrm{i}}=\begin{cases} if\ (a_{l-1} and\ a_l) \in\ u_{i+1},\ Permit\ access\ to\ u_{i+1} \\ if\ (a_{l-1} and\ a_l) \in\ u_{i+2},\ Permit\ access\ to\ u_{i+2} \\ if\ (a_{l-1} and\ a_l) \in\ u_{i+3},\ Permit\ access\ to\ u_{i+3} \\ \qquad\qquad\qquad\qquad\qquad\qquad . \\ \qquad\qquad\qquad\qquad\qquad\qquad . \\ \qquad\qquad\qquad\qquad\qquad\qquad . \\ if\ (a_{l-1} and\ a_l) \in\ u_n,\ Permit\ access\ to\ u_n \end{cases} \tag{3.6}$$

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 to the other classification models to check if they belong to one of the registered

(known) users. If it is recognized as one of the registered users, the access can be granted based on the permission assigned to this user at the registration stage, based on the formula 3.7.

Algorithm 3.4: App-Based User Authentication Model Launching

INPUT: $ev_{i-n}, \dots, ev_{i-2}, ev_{i-1}, a\_req$ are the last app events, *a_req* is the access request, V 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 $(\boldsymbol{u_i}, \boldsymbol{u_j})$

1. **procedure** ()
2. receive ← $\{ev_{i-n}, \dots, ev_{i-2}, ev_{i-1}, a\_req\}$
3. $\{\boldsymbol{f_1}, \dots, \boldsymbol{f_z}\}$← Generate features set
4. **for user** $(\boldsymbol{u_i})$ ← **1** to $\boldsymbol{N}$ **do** ▷ where *N* is the number of users
5. **while** $(a\_req \neq 0)$ **do**
6. **if** threshold >= V **then**
7. CM_i← $\{ev_{i-n}, \dots, ev_{i-2}, ev_{i-1}\}$
8. **if** $\boldsymbol{u_i}$ ← $\{ev_{i-n}, \dots, ev_{i-2}, ev_{i-1}\}$ **then**
9. access ← grant $\boldsymbol{u_i}$
10. **else**
11. $(CM_{j\neq i}, CM_{i+1}, \dots CM_N)$← $\{ev_{i-n}, \dots, ev_{i-2}, ev_{i-1}\}$
12. **if** $\boldsymbol{u_j}$ ← $CM_{j\neq i}$ **then** ▷ where $CM_{j\neq i}$ is the classification model of another registered user $\boldsymbol{u_j}$
13. access← grant $\boldsymbol{u_j}$
14. **else**
15. request ← *sec_fa*
16. **if** correct← *sec_fa* **then**
17. $\boldsymbol{D_i^{train}}$← update (ev_i) ▷ update the model with the new utilized app
18. access ← grant $\boldsymbol{u_i}$
19. **else**
20. access ← deny $\boldsymbol{u_i}$
21. **end if**
22. **end if**
23. **end if**
24. **end while**
25. **end for**
26. **end procedure**

The strategy here comprises the computation of *N-1* models and the decision (D_i) will then be made based on most of the highest probability score from the models.

$$D^{k_u} = \sum_{=(j\neq i)}^{n} ide\_Fun(\, M_j(k_j)) \tag{3.7}$$

where *u* is the unknown received sample, *D* is the decision score, *n* is the number of classification models, *K* is the collection of the events that need to be fed to the classification model, and *ide_Fun* is the authentication 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, in the case of false classification of the event user and a second-factor authentication 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 3.4 shows the process of launching the app usage pattern-based user authentication model.

3.5 Summary

In this chapter, we present a contextual authentication framework for smart homes. This framework considers both user authentication and device-to-device message authentication. In this framework, there are characteristics and features which are considered as fundamental properties of the framework. For example, users are not required to set up any security configuration but are required to provide some related information and preferences that will be enhanced with contextual information collected by the system itself. Preliminarily, at the app app-based authentication model training, the framework

utilizes other information in order to reduce unauthorized access until building the model. During this stage, users are provided access privileges that expire without the need for manual revocation in the case of not manually terminating the access session, especially during the model training and testing stage.

App usage data is utilized to characterize user access patterns in accordance with the app activities of end-devices. In this technique, historical user interaction activities with the apps is utilized to characterize user behavior utilizing classification. Additionally, the presented app-based authentication model does not require specific action from the user in order to be authenticated, but it is based on regular actions while accessing apps, which enhances usability to users.

Chapter 4. Experimental Evaluation and Results

This chapter describes the implementation of the proof of concept prototype, experimental evaluation results and a discussion. According to the categorization of the IoT applications as presented in [139], the implementation and evaluation of the proposed framework is based on the first category, which is the smart home scenario. Following this approach, an authentication mechanism for smart homes, which integrates retrieved contextual information in a real-time manner, is introduced. This chapter then presents a performance evaluation of a device-to-device message authentication scheme, followed by app-based user authentication model evaluation metrics and results.

4.1 Prototype of Context-Based Authentication

The proposed prototype architecture was implemented at the Devices, Networks and Architecture (DNA) Lab, as Figure 4.1 shows [149]. This section describes the contextual information retrieved and the authentication process based on the collected contextual information.

The prototype implementation utilizes a Linksys E1200 [150] router (HG) flashed with DD-WRT [151]. The application then runs on a Raspberry Pi, the TS, which can control the router using the SSH functionality provided by the DD-WRT firmware. The HG uses two wireless networks: one for users and one for home IoT devices, to allow for access control between the two networks. The firewall blocks all access to the home IoT network by default, allowing only access to the Raspberry Pi (TS) (secure flask server) application, which controls further access. A simple port forward on the HG to the TS would allow for external access to the application and remote control of devices if permitted.

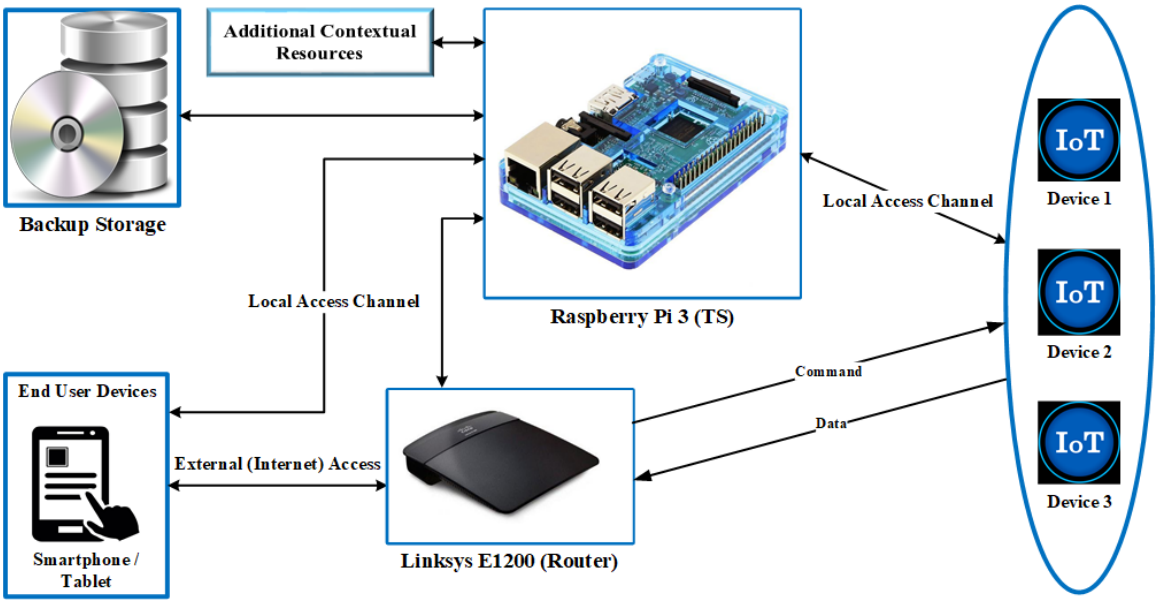

Figure 4.1. Architecture of the implemented prototype

The application is a Python program that runs a Flask [152] web server and Paramiko [153] SSH session along with a MySQL database. The Flask server handles user authentication and provides the controls necessary to interact with the home IoT devices on the network, along with the administrator's tools to add or configure new users or devices. The Paramiko library establishes an SSH connection to the HG, which allows the application to both searches for connected devices and controls their interactions. Finally, the MySQL database stores user's data and devices data along with their individual configurations, rule sets, and logs. Table 4.1 shows the detailed specifications of some of the used devices for the implementation. To access any of the home IoT devices on the network, the user would first browse to the page where the application is hosted on the TS. This would either be a statically assigned IP address or hostname that could be easily bookmarked by frequent users or shared with other users. The user accesses the home devices via a homepage that allows the control of less sensitive devices and hide or deny control of more sensitive devices. Each user would also be able to have an associated calendar to determine when

access to devices should be allowed. For this prototype [154], integration with a user's Google Calendar was used, along with a cached copy in the database. This allows both users to determine times when they will be away from home and therefore not accessing certain devices, as well as enables the homeowner to determine times to explicitly permit or deny access to certain devices by a user.

Table 4.1. Specifications of devices used in the prototype implementation

Brand	Device Type	Device Name	Network Interfaces	Revision	Power
Linksys	Wireless Router	N300 Wi-Fi Router	4x 10/100 Ethernet and 802.11n Wi-Fi	E1200-V2	AC to DC
Raspberry Pi Foundation	Single-Board Computer	Raspberry Pi 3	10/100 Ethernet, 802.11n Wi-Fi and Bluetooth 4.1	Model B	USB to DC

New devices can be added to the system by connecting them to the wireless network intended for the device, and then going to the respective page. Devices are listed on the page based on the router's ARP table, acquired through SSH, and cross-referenced against the database to determine if they have already been added and configured. In addition to being able to set their name, description, and static IP, used devices can be set to only allow access from certain user accounts or be set to ignore location authentication if they are in a fixed location. Home IoT devices can also be configured to only permit access to certain users, along with many options, such as allowing external access, anonymous access, requiring location authentication, or restricting usage to certain devices or schedules. All actions by users are logged in the database against the device origin and the user credentials

if logged in. This includes actions such as arriving or departing a location, accessing the site or devices, and logging in or out of the site. Some of the contextual information can be collected and maintained solely by the TS and normal responses from the user end-devices, while others would require extra data from the user's end-device itself. This is achieved through the use of an app installed on the end-users' devices as a background service, such as an Android app on the Google Nexus tablet or Samsung Galaxy Note 4 phone used in the DNA lab [155]. Because of susceptibility to loss or theft, no information related to users is stored on the devices. This ensures that if the device is compromised, the adversary cannot learn the user profile and simulate the user's behavior to gain system access. All data is replicated to a backup location to avoid both data loss and the need to retrain user profiles, thus preventing a malicious user from rebuilding a user's profile to match their own contexts.

4.1.1 Context-Based User Authentication

In evaluating the effectiveness of the implemented prototype in regard to context-based user authentication, we have used the following criteria:

- The overhead (time/ms) imposed on the system by each added attribute used in the authentication process

- The authentication-assigned weights and thresholds set by the administrator and their effects on access decision-making

- The ability for the TS to handle multiple simultaneous requests without bottlenecking access to smart devices

4.1.2 Performance

This part of the evaluation examines the time (ms) imposed on the system by each added authentication parameter in the authentication processes shown in Tables 4.2 and 4.3.

Table 4.2. Performance of individual authentication methods

Used Parameter	Local Access Time (ms)	Internet Access Time (ms)
No authentication	7	90
IP address-based location (network)	8	90
Bluetooth-based location (proximity)	14	97
knowledge-based credentials (username, password)	15	96
Calendar access	13	96

Table 4.3. Performance of authentication methods combined

Used Parameter	Local Access Time (ms)	Internet Access Time (ms)
No authentication	7	90
Location based on both IP address and Bluetooth	14	90
Location based on both IP address and Bluetooth, and knowledge-based credentials (username, password)	16	98
Location based on both IP address and Bluetooth, knowledge-based credentials (username, password), and calendar access	20	98

The above tables demonstrate our performance tests on all the authentication methods, both individually selected contextual attributes and combined attributes. Both the location based on the IP address and the user's session cookie, granted by user credentials, are retrieved

from the user's request itself. Nearby Bluetooth devices are retrieved from the cache, and the current time is compared against events in the user's calendar in the database. As expected, no authentication yields the fastest results, with a combination of all methods being the slowest. However, as can be seen in Tables 4.2 and 4.3, there is very little overhead on the request-level associated with the different authentication methods. While location, calendar access, and even knowledge-based credentials most affect the response times, the difference is almost negligible, especially when accessing the system over the Internet.

4.1.3 Authentication Weights and Device Thresholds

For the second experiment, the system framework is evaluated with regards to the calculated confidence levels, based on the assigned weights and the subsequent access levels given to the various services. As mentioned in the design goals, the framework will utilize contextual information parameters to enhance the knowledge-based credentials to authenticate the user. Hence, as a preliminary step, users will be authenticated based on the overall available parameters and the average re-check interval. The re-check interval is set to one minute as an initial value, and this time is then updated based on the average of the previous access sessions of the user. For example, if the average of the previous access sessions is 10 minutes, then the frequency of re-checking the context information would be one-third of this time. When any of the authentication requirements do not meet predefined roles (the predefined confidence levels, as shown in Table 4.4, are the minimum threshold for accessing the required device), the access session will be automatically revoked. This quality is calculated with assigned weights to each included authentication parameter, for example: location weight = 20%, calendar schedule data weight = 20%, time = 10%,

username and password = 30%, and profile data (such as preferences) = 20%. Smart devices are set up with a required confidence level needed to access their services. If in the event that confidence level is too low to access a service, the user will be requested to enter a second-factor authentication, such as security questions/preferences, or try again when scheduled or no longer attempting to access remotely, depending on what attributes are lacking.

Table 4.4 shows some example weights for contexts that are assigned by default, and security levels that are assigned by the administrator which will be applied to smart devices of varying concern. The confidence level is considered as the total of the assigned weight of the available contextual information. As seen in Table 4.4 (A), when providing both knowledge-based credentials of username and password, the assigned weight is 40 whereas, when achieving access to only the calendar, the assigned weight is only 10. In contrast, as seen in Table 4.4 (B), the highest security level is 4 with threshold 100, whereas the lowest security level is 1 with threshold 30.

Table 4.4. Example of authentication weights for context parameters and threshold for accessing devices

(A)

Available Parameters	Assigned Weight/100
Username & Password	40
Location (proximity)	30
Location (network)	20
Calendar	10

(B)

User Security Level	Access Threshold/100
4	100
3	70
2	50
1	30

Table 4.5 reflects some examples of calculating a confidence level by adding together the assigned weights and determining if the system should grant the user access to the requested service. As can be seen from the same table, should the confidence level (calculated by combining the available parameter weights) be sufficient to meet the security level/access threshold of the requested service, access is granted.

Table 4.5. Calculated confidence levels and authentication scenarios

Available Parameters	Weights	Confidence Level	User Security level	Service Security level	Access Decision
Username, password, and Bluetooth	40, 30	70	3	2	Granted
Bluetooth and on local network	30, 20	50	2	2	Granted
Scheduled and on local network	10, 20	30	1	3	Denied
Username, Password	40	40	1	4	Denied
Bluetooth	20	20	1	2	Denied
Scheduled, Bluetooth and on local network	10, 30, 20	60	2	2	Granted
Scheduled, Bluetooth and on local network	10, 30, 20	60	2	1	Granted

If the calculated confidence level is insufficient to meet the security level/access threshold of the requested service, access will be denied. As an example, as seen in Table 4.5, with the ability to access calendar information as well as the location based on the network, the total calculated weight will be 30. Thus, if the user is requesting access to a service with an assigned security level of 3 or higher, the request will be denied, since the achieved security level is lower than that required. In addition, having only a username and password would allow login and access to only those services that do not require a high threshold. As an example, weight 40 will not allow access to important services by setting a higher access threshold.

4.1.4 Scalability

In this section, the TS is tested for its ability to handle multiple simultaneous requests to access to the home network. This evaluation demonstrates that the implemented prototype is able to handle several simultaneous requests simulated by Apache JMeter, without significantly affecting response times. As shown in Figure 4.2, the implemented prototype is able to handle several simultaneous requests without significantly affecting the response times from our previous trials.

	Start Time	Thread Name	Sample Time(ms)	Latency
1	23:41:05.285	Thread Group 1-4	45	45
2	23:41:05.363	Thread Group 1-2	63	63
3	23:41:05.410	Thread Group 1-3	48	48
4	23:41:05.507	Thread Group 1-1	64	64
5	23:41:05.763	Thread Group 1-4	43	43
6	23:41:05.837	Thread Group 1-2	69	69
7	23:41:05.891	Thread Group 1-3	47	47
8	23:41:05.984	Thread Group 1-1	63	63
9	23:41:06.228	Thread Group 1-4	42	42
10	23:41:06.301	Thread Group 1-2	64	63

of Samples 211 **Latest Sample** 40 Average 55 Deviation 11

Figure 4.2. Snapshot of Apache JMeter multiple simultaneous requests

log_item	log_user	log_ip	log_device	log_event	log_description
6	1	192.168.1.5	NULL	3	User Login
8	1	192.168.1.5	NULL	3	User Login
17	NULL	192.168.1.5	NULL	4	Failed Login - Bad Password - Get user
19	1	192.168.1.5	NULL	3	User Login
21	NULL	192.168.1.5	NULL	4	Failed Login - Bad Login
22	NULL	192.168.1.5	NULL	2	User Failed to Register - Fields Missing
23	NULL	192.168.1.5	NULL	2	User Failed to Register - Database Error

Figure 4.3. Historical data captured by the TS

As shown in Figure 4.3, the implemented prototype stores all historical information that is related to users interacting in any way with the system or various services. This includes, but is not limited to, user logins (both successful and failed), accessed services (both permitted and denied), and new registrations.

4.1.5 Comparison with Related Work

Although we did not find a similar prototype at the time to compare the results with, we compare the presented approach with the related work regarding the utilized contextual parameters. Table 2.1 provides a comparison of some relevant related works regarding utilized information, advantages, and limitations. From the comparison, the presented context-based authentication in this thesis differs from the aforementioned solutions in several ways. It not only depends on one parameter but also on available contextual information such as location based on both IP address and Bluetooth, calendar and profile. Additionally, it provides authentication by checking contextual information in real-time during the access session without user intervention.

4.2 Evaluation of Device-to-Device Message Authentication Scheme

This section provides an evaluation of the proposed scheme regarding computation time and cipher-text message size [156].

4.2.1 Correctness

The correctness of the retrieved message is derived as follows:

- Correctness of Keys:

$$K' = H_2(\hat{e}(sQ_{d2}, C_1))$$

$$= H_2(\hat{e}(sQ_{d2}, r_i P_1))$$

$$= H_2(\hat{e}(r_i Q_{d2}, sP_1))$$

$$= H_2(\hat{e}(r_i Q_{d2}, P_{pub}))$$

$$= K$$

- Correctness of Equation 3.1:

$$\hat{e}(C_{sign}, P_1) = \hat{e}(r_i h + sQ_{d1}, P_1)$$

$$= \hat{e}(r_i h, P_1)\, \hat{e}(sQ_{d1}, P_1)$$

$$= \hat{e}(h, r_i P_1)\, \hat{e}(Q_{d1}, sP_1)$$

$$= \hat{e}(h, C_1)\, \hat{e}(Q_{d1}, P_{pub})$$

- Correctness of Unsigncryption:

$$m' = C_{enc}, \oplus K'$$

$$= m \oplus K' \oplus K$$

$$= m$$

4.2.2 Results

For evaluation purposes, the scheme is assessed according to the computation time that is taken by both the signcryption and unsigncryption processes. Although at the time, we could not identify any signcryption scheme that is designed mainly for providing authentication in smart home environments, we evaluated the presented scheme by comparing it with two other IBS schemes [157][158] that are implemented according to

the authors' descriptions. Table 4.6 provides a comparison of our scheme with these two schemes regarding the computational overhead and cipher-text length. As the comparison shows, our scheme outperforms the other two schemes regarding computational costs and cipher-text size.

For this evaluation, we consider the implementation of Tate Pairing on an MNT curve with an embedding degree of $k = 6$ where G_1 is represented by 161 bits, and order p represented by 160 bits on a machine with Intel Pentium IV 3.0 *GHz*. Since pairing and point multiplication computations are the main computations in the proposed scheme, they have been considered in calculating the execution time in comparison with the related work. We adopt the measured processing time [159][160] as follows: $T_{\hat{e}} = 4.5\ (ms)$ is the time of a pairing operation whereas $T_{mul} = 0.6\ (ms)$ is the time of a one point multiplication operation in G_1. As shown in Table 4.6, in the proposed scheme in this thesis, there are three multiplication operations and five pairing operations for the whole scheme.

Table 4.6. Computational overhead and cipher-text length

Scheme	Performed Operations			Cipher-text Length
	T_{mul}	ê	$T_{total}\ (ms)$	
The Proposed Scheme	3	5	24.3	$\|m\| + 2\| G_1\|$
[157]	5	5	25.5	$\|m\| + 2\| G_1\|$
[158]	4	5	24.9	$\|m\| + 2\| G_1\| + \|Z_p^*\|$

In the signcryption stage, there are three multiplication operations and one pairing operation and in the unsigncryption stage there are four pairing operations. Thus,

signcryption and unsigncryption of the proposed scheme are calculated according to the following formula:

$T_{total}\ (ms)$ = (number of the pairing operations) x $T_{\hat{e}}$ + (number of the multiplication operations) x T_{mul} (4.1)

Thus, the execution time of the whole scheme would be: $T_{total}\ (ms) = 5\ x\ T_{\hat{e}} + 3\ x\ T_{\text{mul}} = 5\ x\ 4.5 + 3\ x\ 0.6 = 24.3\ ms$

4.3 Evaluation of App-Based User Authentication Model

As a first step to considering the app usage patterns of end-user devices (e.g., smartphones or tablets) for user access authentication in smart homes in IoT networks, the primary objective of this work is to investigate the following research questions:

- Can users be authenticated based on the time usage patterns of apps? If there is a change, how much time (e.g., days or weeks) is needed to build a model to authenticate users? This question investigates the change in app usage patterns over a long period (e.g., months), and its effect on accuracy, as well as determines the time needed to build a model to authenticate users with a minimum threshold regarding training data.

- Do the overall app usage patterns change over time (e.g., a week or a month)? If so, does any change affect accuracy?

Additionally, several studies have examined mobile app usage in the broader population but failed to consider network traffic patterns during app access for user authentication.

Thus, another step is to investigate the following research questions related to using app access patterns and network traffic patterns on Wi-Fi or cellular to authenticate users:

- Can users be authenticated based on time usage patterns and generated network traffic patterns while accessing apps? This question investigates whether the total access time and the traffic generated while accessing apps can be utilized to accurately authenticate end-users.
- Do the overall app usage patterns and network traffic patterns change over time (e.g., a week or a month)?
- Does either a Wi-Fi or a cellular access pattern more accurately authenticate users?

4.3.1 Evaluation Metrics

Many evaluation metrics can be utilized, such as true positive (TP); false positive (FP); true negative (TN); and false negative (FN).

$$True\ Positive\ Rate\ (TPR) = \frac{TP}{TP + FN} \tag{4.2}$$

$$True\ Negative\ Rate\ (TNR) = \frac{TN}{TN + FP} \tag{4.3}$$

$$Fals\ Positive\ Rate\ (FPR) = \frac{FP}{FP + TN} \tag{4.4}$$

$$Fals\ Negative\ Rate\ (FNR) = \frac{FN}{FN + TP} \tag{4.5}$$

The classification accuracy and error rates of the utilized algorithm can then be raised in the form of a confusion matrix, as shown in Table 4.7.

Table 4.7. A Two-class confusion matrix

	Predicted positive	Predicted negative
Actual positive	True positive (TP)	False positive (FP)
Actual negative	False negative (FN)	True negative (TN)

The classification accuracy and the error rate can be calculated as follows:

$$Classification\ Accuracy\ = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.6}$$

$$Error\ Rate\ =\ 1 - (classification\ accuracy) \tag{4.7}$$

In contrast, most classifiers are more sensitive in detecting majority classes than the minority examples, which are the most important in our case, and in this case the classifier may be biased. The classification accuracy metric may not reflect the real performance of the proposed model; hence the literature recommends other advanced metrics, such as recall and precision, to evaluate the accuracy of the classification algorithms [161][162]. In this thesis, the classification performance of the positive class is more significant, thus the two measures. The TPR, also called recall (R), references the percentage of retrieved objects that are relevant to the targeted class. The other classification performance metric which is known as precision (P), references the percentage of relevant objects which are identified for retrieval. These two measures, R and P, are then used in calculating the F-measure to give a single accuracy measurement of a class, ensuring that both measures the R and the P are reasonably high.

$$Recall\ (R) = \frac{TP}{TP + FN} \tag{4.8}$$

$$Precision\ (P) = \frac{TP}{TP+FP} \tag{4.9}$$

$$F - measure = \frac{2 * P * R}{P + R} \quad (4.10)$$

In the presented model, we need to reduce the FPR as much possible in order to improve security. Consequently, we consider FPR as a high priority in our model. The FNR, which also has an impact on the performance of the model, is important as it will impact the usability of the model. As a consequence, it is taken into consideration in the proposed model. In order to reduce the cost of misclassification, we need the model to reduce the FPR, making it close to 0, thus increasing the precision and maximizing the R. In other words, we can afford the expense of classifying positive classes as negative, an action which has less impact than classifying negative classes as positive. Thus, our main objective is to decrease the FPR as far as possible, followed by eliminating the FNR. The one-vs-all [147] is utilized, in which one class is trained with all N other classes with an average accuracy (Ave) as exhibited in Equation (4.11). As an example of this approach, if there are five different classes (a_5) of data, one class will be trained against the other four.

$$\textit{Average (Ave)} = \frac{1}{n}\sum_{1}^{n} a_i \quad (\text{where } i = 1,2, \ldots, n) \quad (4.11)$$

4.3.2 Datasets

For the evaluation procedure, three datasets are utilized. The first one, the Android App-Usage dataset [163], is extracted from users' access profiles while using Android apps on end-user devices. This dataset contains two categories, the first of which is management activity data, including app installation, uninstallation and updates. The second category is the network trace data, which collects any network activity, such as whether the access is made via cellular or Wi-Fi networks, and provides the daily total access time and traffic generated from foregrounds and backgrounds. This information is calculated according to

the flows at the TCP level and is counted separately for cellular and Wi-Fi networks. Distinguishing between the background and foreground is accomplished by examining the Android system stack every two seconds. The app is considered as running in the foreground if it was active in the last two seconds. The features used in the evaluation include the user's identifier, the date of this data entry, time, the time of app access, the app identifier, foreground (Cellular and Wi-Fi network connection) time, and foreground (Cellular and Wi-Fi) generated traffic. The second dataset is the *UbiqLog4UCI* [164] that is collected from 35 users in a period of approximately three months. As not all users have app access information that is adequate for analysis, the information from only 30 users is utilized in the evaluation. The third dataset is the *LiveLab* [165] project dataset collected from 35 users in a period of approximately one year. 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 evaluation is related to apps usage and the registered time when apps were accessed.

4.3.3 App Access Time Patterns and Network Traffic Patterns

This subsection provides evaluation results of the classification model that uses usage patterns and network traffic patterns for the purpose of authenticating users. The authentication model works by classifying each single event, namely the usage patterns and the generated traffic of the accessed apps, received from the user’s device and identifies the users [148]. The model selects continuously accessed apps shared among users and predicts and classifies potentially unauthorized actions in the user’s access behavior. The focus in this thesis is on classifying multi-users rather than one user. To select the most suitable

classifier as well as ensuring that the model is not classifier specific, we applied various classification strategies, including Random Forest (RF) [166], Decision Tree (DT) [167], K-Neighbors Classifier (KNN) [168], Gaussian Process Classifier (GPC) [169], GaussianNB (NB) [170], Multi-layer Perception Classifier (MLP) [171], Quadratic Discriminant Analysis (QDA) [168], Support Vector Machine (SVM) [172], Nearest Centroid (NC) [173], Ridge Classifier (RC) [174], and BernoulliNB (BNB) [175]. These algorithms are applied to each part of the dataset, time access events and network traffic events. We divided the dataset into 70% for training the model and 30% for validating the model [176][177]. As a first step, we compared common classification approaches on the training set in terms of recall, precision and F-measure. Then, the algorithm that provides the highest recall, precision and high F-measure is implemented, which is the RF classifier.

Despite some apps in the utilized dataset with minor usage time, with access time close to zero, being tested for the purpose of identifying users, we found the effect on accuracy was minor, and such apps were removed from the datasets. Additionally, the dataset is cleaned from unnecessary rows when both Wi-Fi network traffic data and foreground cellular network traffic were equal to zero. From the visualization of the dataset, there is a variance of users' daily access time during weekdays. Thus, extracting the days of the week and converting them to a numerical representation is an important step for the classifier. However, mapping weekdays into numbers may lead the algorithm to give more importance to larger numbers than others because of their higher numerical values. Thus, to avoid the problem, these numbers are converted into seven binary columns.

For the purpose of user authentication, our goal is to use apps that are most employed by users. However, the main target here is to test our model on differentiating between users

who utilize same apps in approximately at the same daily intervals. Therefore, we removed the less frequently used apps and selected the most used apps used by all users during the 20 weeks. To avoid the classifier being biased to the majority samples, an up-sampling mechanism is applied to both parts of the dataset, access time and network traffic events. Furthermore, the oversampling is achieved after dividing the dataset and applied only on the training data. This step avoids the validation and training sets from having the same samples, thus eliminating overfitting.

The next stage is to examine the difference between including all the accessed apps and the most accessed apps selected during the preprocessing stage, as well as to guarantee that these apps come from the same distribution by users in the dataset, student t-test (*t*) statistical significance with the degree of freedom (*df*) [178] is used, based on the formulas 4.12 to 4.15 with an assumption of unequal variances:

$$t = \frac{\bar{z}_1 - \bar{z}_2}{\sqrt{(q_1^2/r_1)+(q_2^2/r_2)}} \tag{4.12}$$

$$df = \frac{[(q_1^2/r_1)+(q_2^2/r_2)]^2}{\frac{(q_1^2/r_1)^2}{r_1-1}+\frac{(q_2^2/r_2)^2}{r_2-1}} \tag{4.13}$$

$$q_1^2 = \frac{\sum_{i=1}^{n_1}(z_i-\bar{z}_1)^2}{r_1-1} \tag{4.14}$$

$$q_2^2 = \frac{\sum_{j=1}^{n_2}(z_j-\bar{z}_2)^2}{r_2-1} \tag{4.15}$$

Where $\bar{z}_1$and $\bar{z}_2$ are the sample means, q^2 is the sample variance, r_1 and r_2 are the sample sizes with a degree of freedom df, using Satterthwaite's approximation. The standard deviation resulted from including all apps is 2.75, and 2.9 from including the most used apps. The calculated *t* value is 0.83, which is less than 2.06, with a degree of freedom of 25 at p=0.05. Consequently, there is no significant difference between both data samples, all apps and most used apps in the dataset.

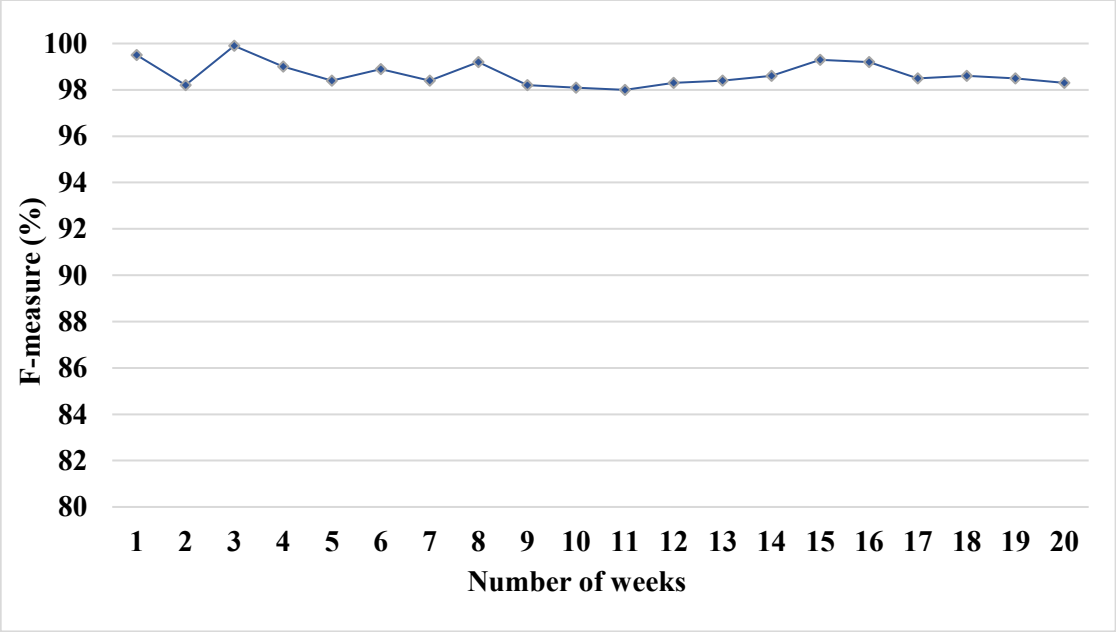

Figure 4.4. F-measure performance of both access time and network traffic patterns

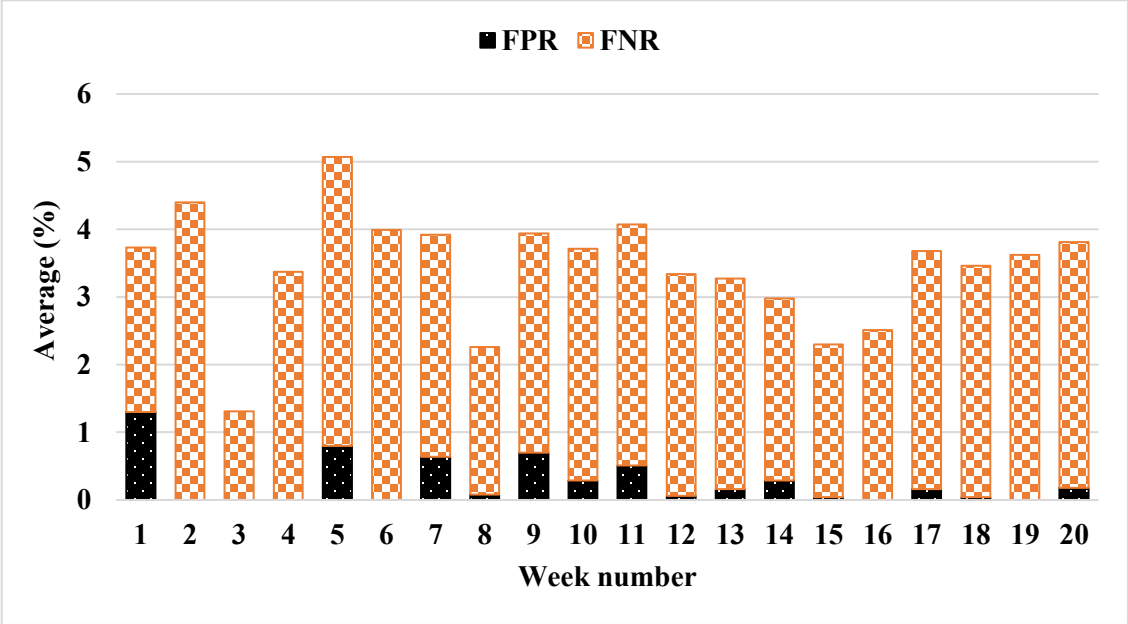

Figure 4.5. FPR and FNR performance of both access time and network traffic patterns

The performance of the proposed method is then examined with regard to the false positive rate (FPR) and false negative rate (FNR) and F-measure. The results obtained are based on a dataset from users who accessed their devices for five months of the same year. Within

this timeframe, a period of 152 days, 530 different apps were accessed at least once by one user, generating 66,704 different events. In spite of the large number of users included in the dataset collection, only the data of 10 users is publicly available. The next step is to test the model to authenticate users against 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. The model is tested to authenticate users in an incremental way in terms of the number of weeks for a period of 20 consecutive weeks, as shown in Figure 4.4. Training the model is achieved data of the prior week(s) and then the model is tested on unseen data of the following week. The performance of the model continues to be high with a minimum F-measure of 98%. The model is then tested on the dataset with regard to the FPR and FNR for the same period. The average FPR and FNR utilizing the one-vs-all classification approach is presented in Figure. 4.5. As seen in Figure. 4.5, the model achieves a maximum FPR of 1.3%, which is considered low, when combining both the access time patterns and the network traffic patterns.

The results demonstrate that users can be authenticated based on their total app usage time and generated network traffic, which addresses the research questions set at the beginning. As the most used apps on smartphones are heavily reliant on networks, the primary focus of this test is on user authentication based on the daily app usage and network traffic generated while accessing these apps. The results indicate that users can be authenticated with a minimum F-measure of 98%. In this test, the most frequently used apps are utilized, which reflect the actual patterns of users while accessing the apps, as well as combine both the

total app usage time and generated network traffic. Additionally, the chosen apps are either network-based or the type that rely on networks to function. Furthermore, this test considers only the usage patterns and the network traffic of foreground online-based accessed apps which demonstrates user interaction patterns with these apps, thus reflecting patterns of interaction. Since the utilized apps are continuously accessed, traffic from such apps is continually generated.

From the dataset analysis, most of the traffic is produced via Wi-Fi networks, which are the most utilized user networks. In addition, Wi-Fi traffic patterns provide more authentication accuracy than Cellular networks. However, the accuracy level is improved when combining both Wi-Fi and Cellular network traffic data. Based on the results, there is a minimal change in the usage patterns of users when utilizing the generated traffic data of the most foreground accessed apps by users. Additionally, Wi-Fi network traffic patterns provide more authentication accuracy than cellular networks. However, the accuracy level is improved when combining both Wi-Fi and cellular network traffic data, as shown in results. Although there are some changes in the app access time and in the total traffic during app access time, these changes do not significantly affect the accuracy level as a result of using the most frequently used apps. Moreover, by combining both access time and network traffic patterns, the accuracy level is improved, as shown in Figure 4.5, where the average of FPR and FNR do not change much from week 1 to week 20.

4.3.4 App Access Events-Based User Authentication

This subsection provides evaluation results of the app-based user authentication model that uses apps access event patterns to authenticate users. To evaluate the performance of the app-based user authentication model, the two datasets *UbiqLog4UCI* and *LiveLab* are

utilized, and the authentication performance is considered as the accuracy metric when classifying an access event to one of the enrolled users. The collection procedure for both datasets includes the foreground accessed apps 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. To make sure that the presented model is not classification algorithm specific, three classification algorithms are used in training. The selected classifiers in this research, which are mostly used in the literature, such as in [179][180][181], 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 and utilizes the average in order to improve accuracy and eliminate over-fitting. 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 is the KNN which applies the k-nearest neighbors' vote. Although it is easy to implement, the training data has to be saved at the classification time.

- **Statistical Analysis of the Extracted Features**

The imbalance in the class representations is important to consider during the training and testing process of the model. From the analysis, the imbalance in the classes is clear in both datasets. Consequently, considering the variance in the classes' representation is necessary. 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.

Hence, in order to test similarities between user patterns and to examine the effect of the extracted features on discriminating between users, a comparison regarding the standard deviation (*std*) and the mean is considered, as shown in Table 4.8 for the *UbiqLog4UCI* dataset and in Table 4.9 for the *LiveLab* dataset. The extracted features, in addition to the day of the year and *week_day*, are *app_sq*, *app_cont_sq*, *app_cat*, *app_dur*, *inter_pi* and *intra_pi*. For the analysis, we selected the most similar users from both datasets, according to similarity of access patterns. For the *UbiqLog4UCI* dataset, within the same features, the *app_dur*, we can see from Table 4.8 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).

Table 4.8. 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

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 4.9 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).

Table 4.9. 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

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.

- **Results**

For the evaluation, many evaluation metrics can be utilized to evaluate the presented approach in this thesis. However, our main focus is to decrease the FPRs and FNRs as far as possible. 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 4.6 and 4.7, the number of interactions with apps changes 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.

Another important feature that we considered during feature extraction is *intra_pi*. Figures 4.8 and 4.9 show the average *intrer_pi* between apps for a 12 week period for five selected users from both datasets. As seen from these figures, 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 might affect the classification procedure and increase the FPRs and FNRs. To solve this issue, we utilized a new feature, the *app_cont_sq*, for each app. Therefore, for either a new user or a previously known user, the new added apps will not be included by the model until a specific access sequence threshold is reached, namely the value of *app_cont_sq*. In addition, testing the model against unknown users' data will increase the FPR, especially if limited information is applied during the training phase.

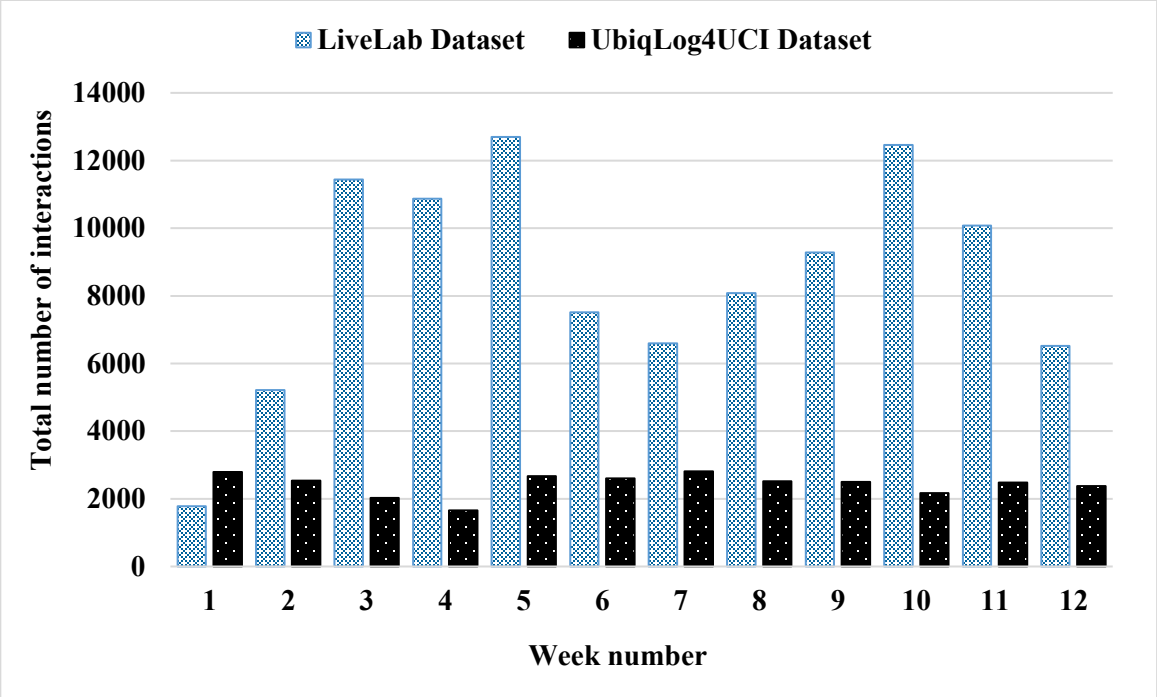

Figure 4.6. Number of interactions with apps per week for 12 weeks

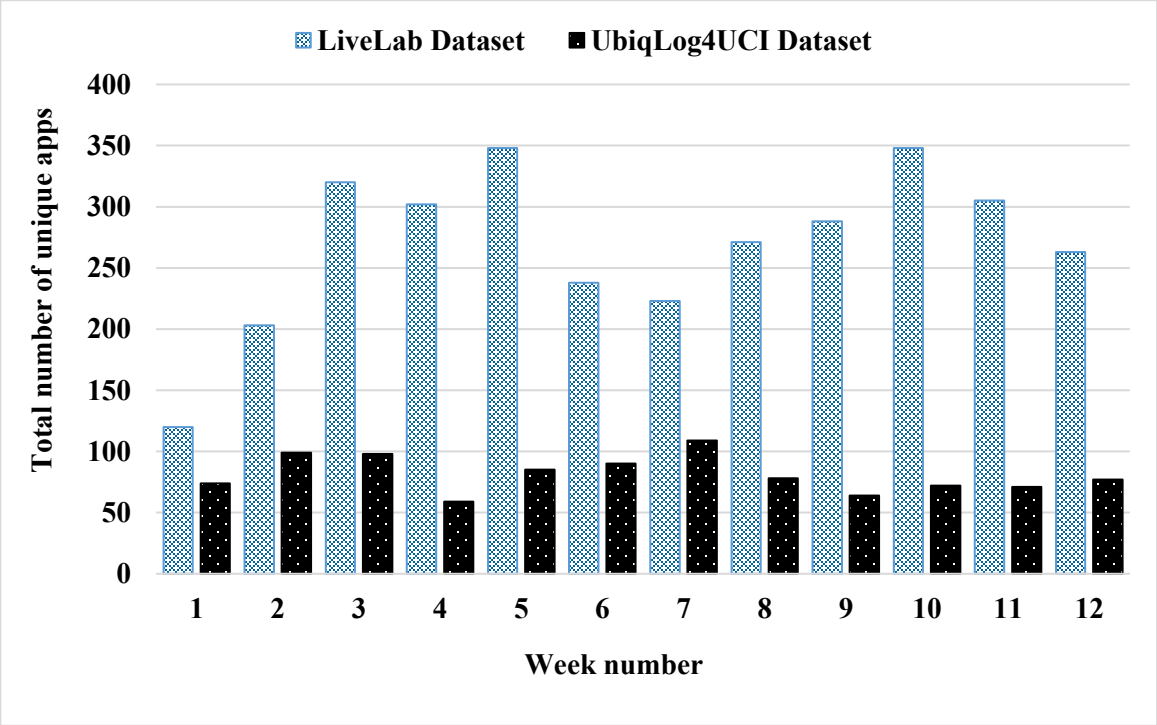

Figure 4.7. Number of utilized apps per week for 12 weeks

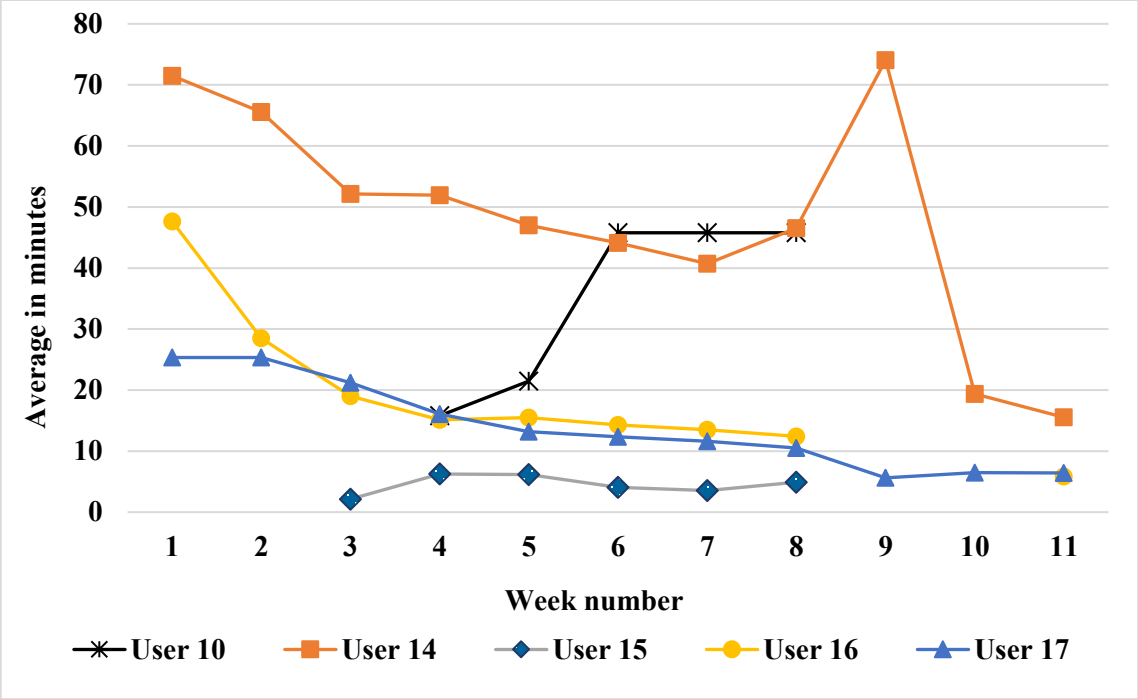

Figure 4.8. The average *intrer_pi* for selected users for the *UbiqLog4UCI* dataset

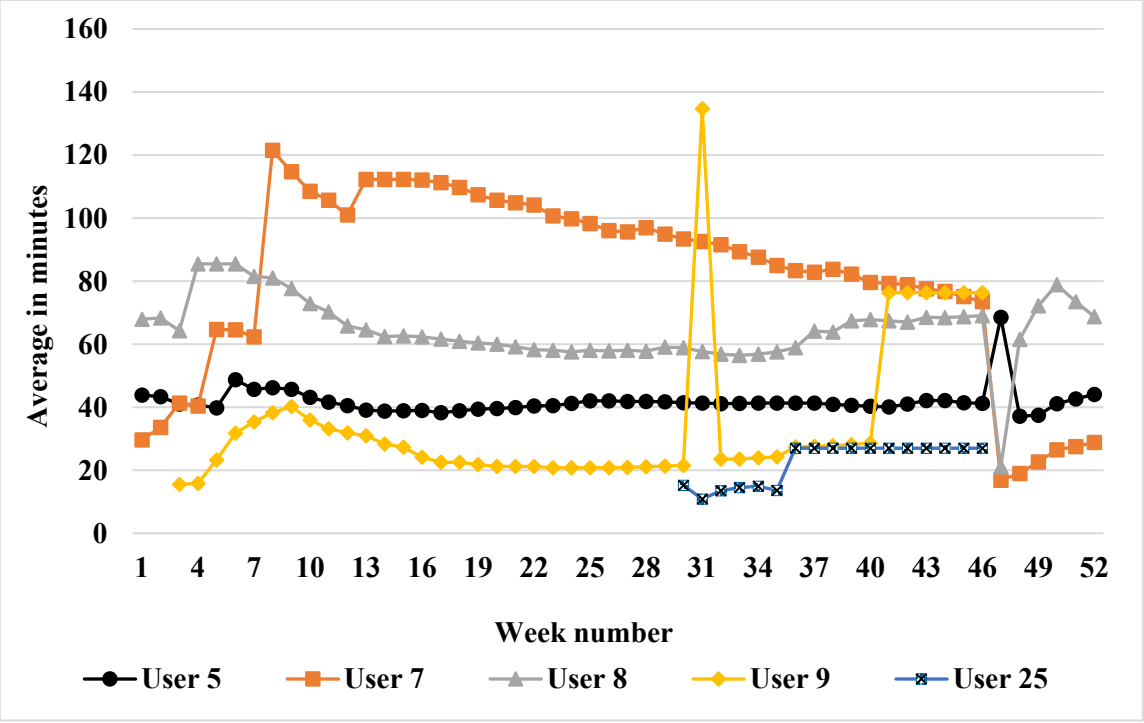

Figure 4.9. The average *intrer_pi* for selected users for the *LiveLab* dataset

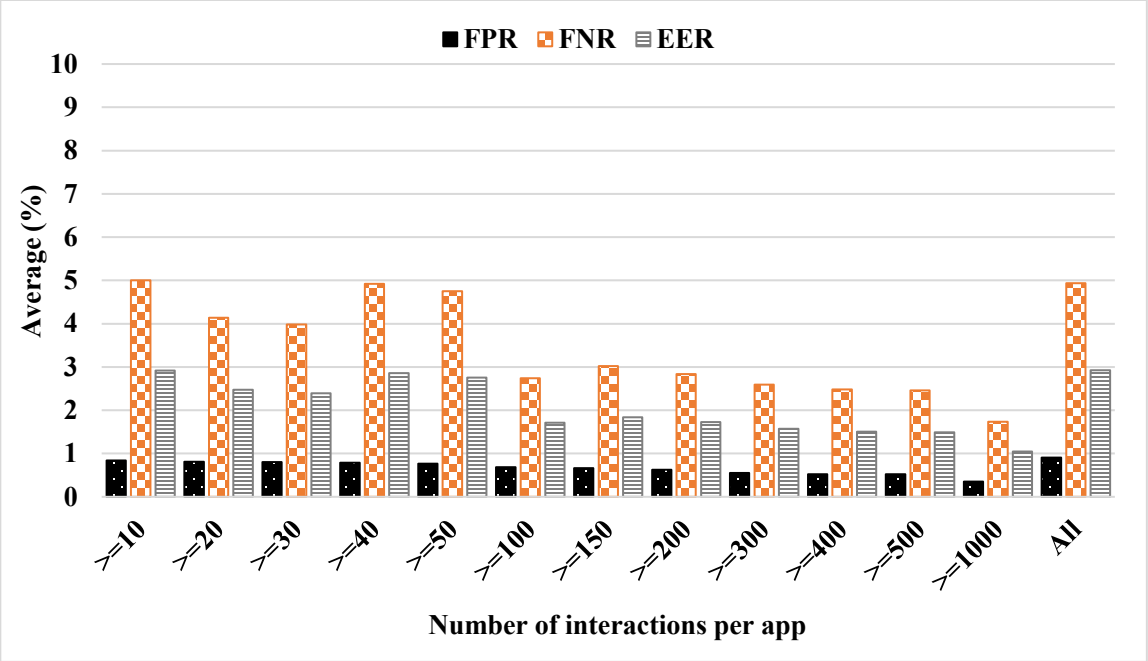

Figure 4.10. Model performance based on the number of interactions with apps for the *UbiqLog4UCI* dataset

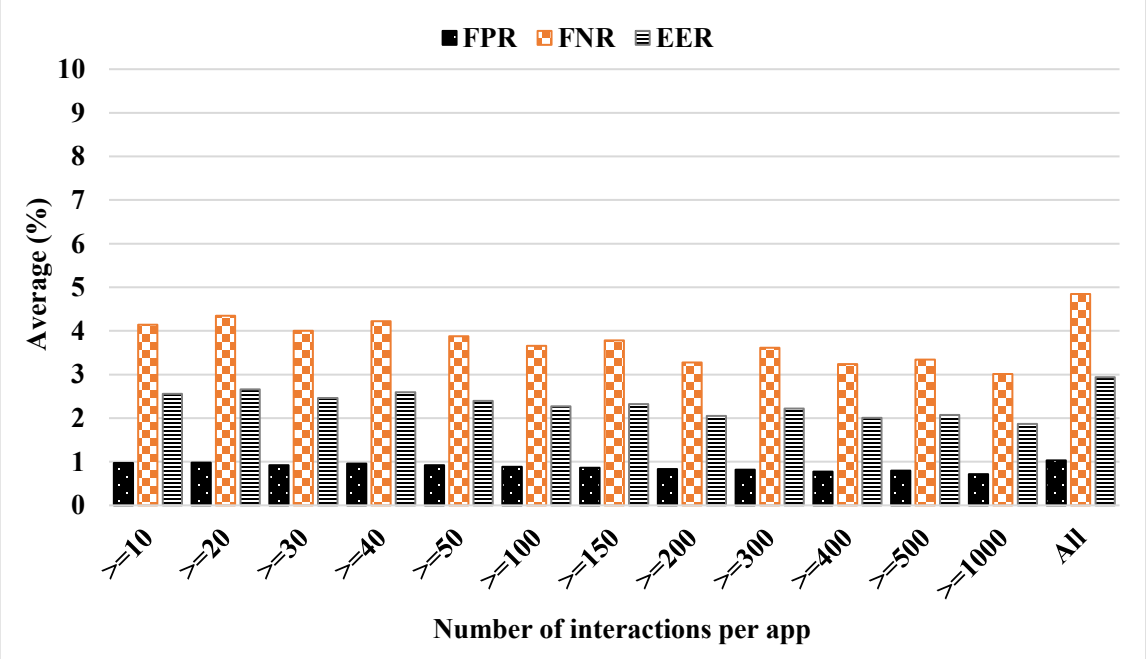

Figure 4.11. Model performance based on the number of interactions with apps for the *LiveLab* dataset

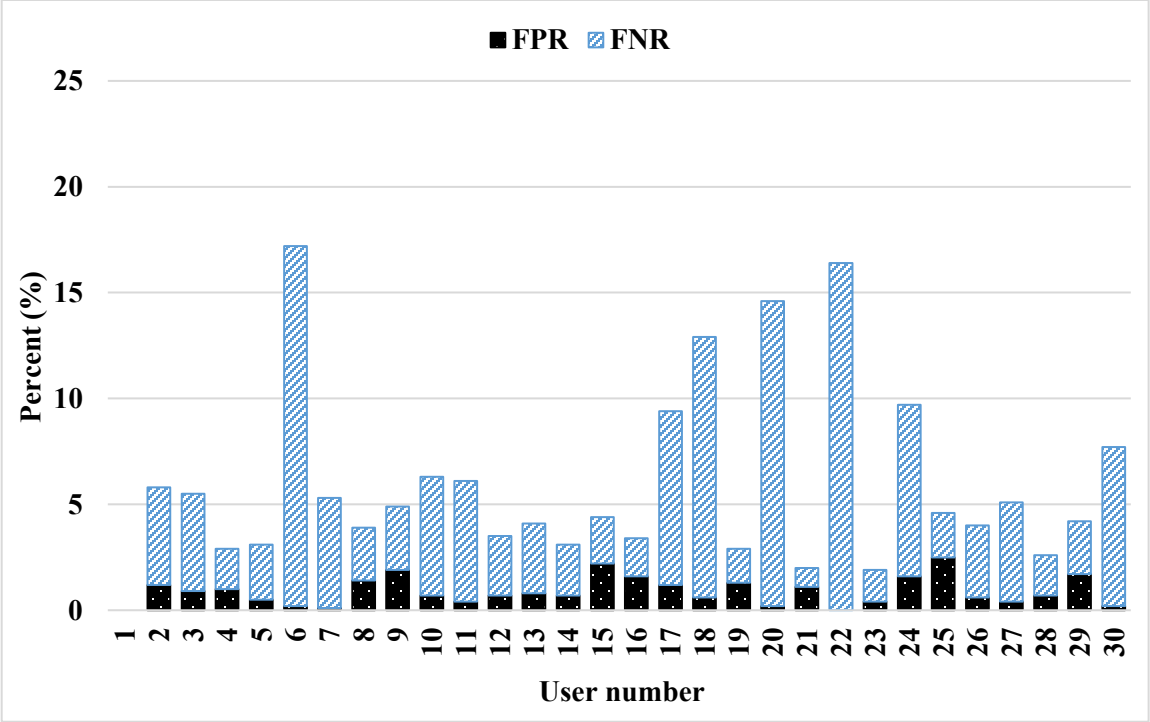

Figure 4.12. Model performance on unseen data for the *UbiqLog4UCI* dataset

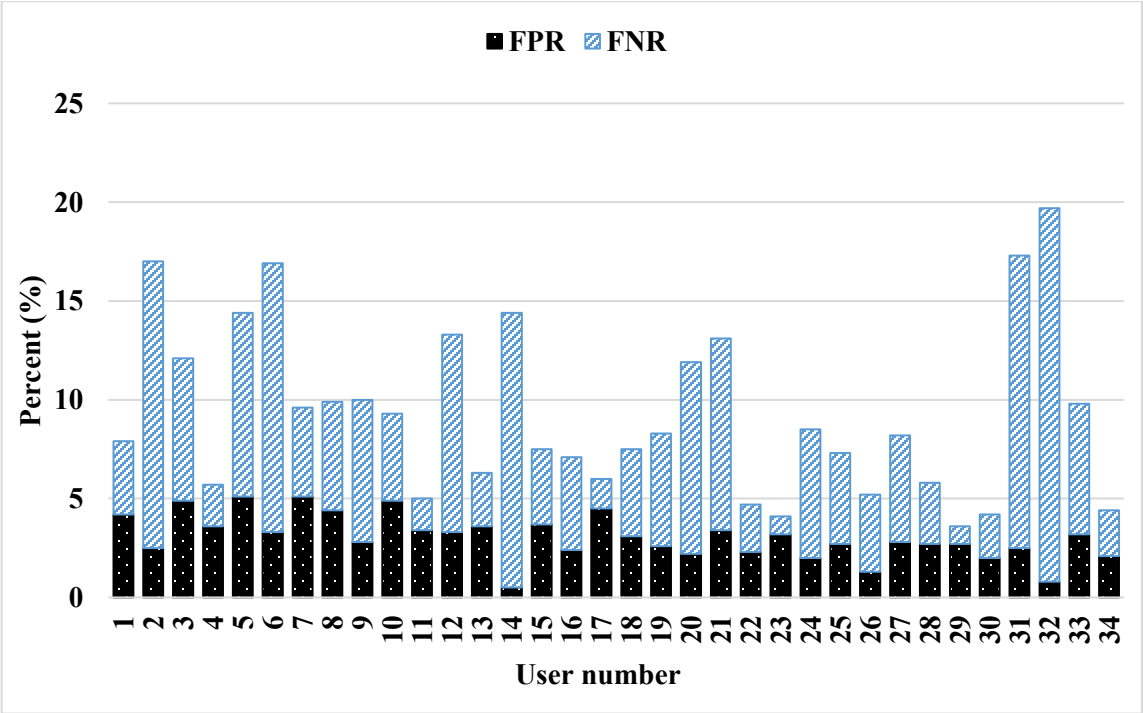

Figure 4.13. Model performance on unseen data for the *LiveLab* dataset

As seen from Figures 4.10 and 4.11, 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 thesis requires a minimum number of *app_cont_sq* for each app to be considered in the access decision, in addition to requiring a specific number of last events, in order to make the decisions at the time of the request. In other words, the app will be considered only when reaching a minimum number of *app_cont_sq,* and the access decision will be made based on the number of the last two events and their classification by the model. As the target is to discover any access anomalies in the network, the number of events required by the decision unit should be as small as possible. This will also reduce the processing time for user authentication. In order to find the best set of access sequence of events, the datasets were tested, the results of which are shown in Figures 4.10 and 4.11. Classifiers can then be trained on data from the owner and others, without assuming any known data from the attacker. Figures 4.12 and 4.13 show the evaluation results of this approach where the threshold is set to 50 or greater. From the results, it is clear that the model achieves good accuracy with a low FPR and FNR. Additionally, as seen from the results, users who have more access events provide a lower FPR and FNR.

The next evaluation is based on the number of users involved. The average EER is shown in Figures 16 and 17. It can be observed from the results that there is little change in the average EER when increasing the number of users. The maximum EER is 3.13% when considering the RF classifier. Moreover, as seen in Figures 4.14 and 4.15, the performance of the presented approach consistently remains below 3.13%. However, as the number of users increases, the performance slightly decreases. This decrease in performance is mainly a result of similarity in the users' access sessions, as usage may change and similarity

among users may be present. The results indicate that the model produces a low EER even when the number of users increases.

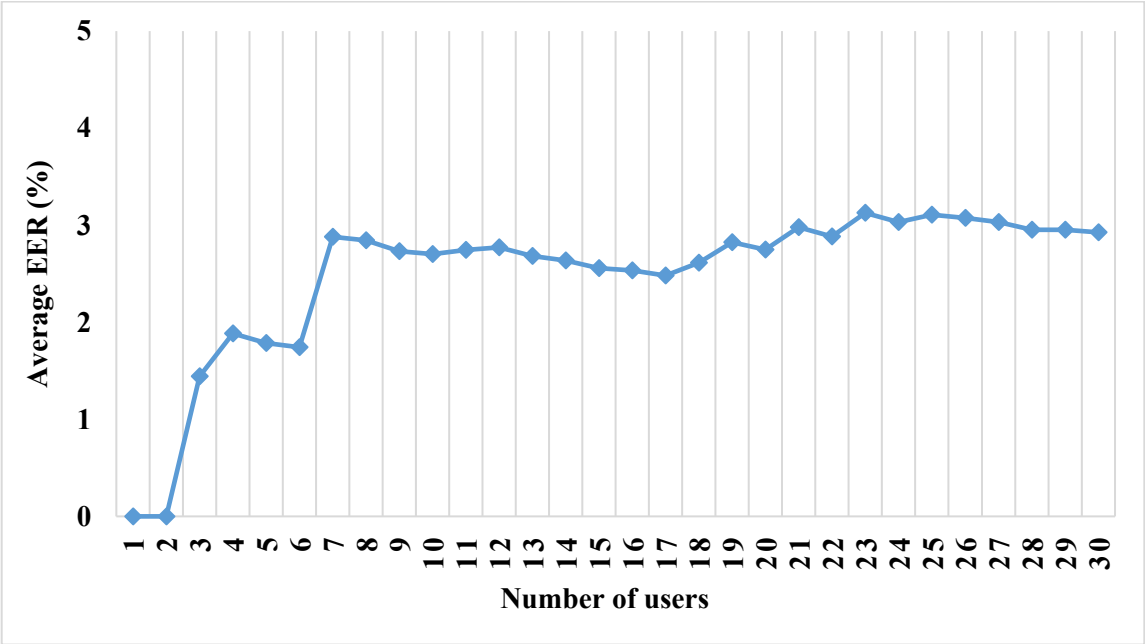

Figure 4.14. Model performance based on the number of enrolled users for the *UbiqLog4UCI* dataset

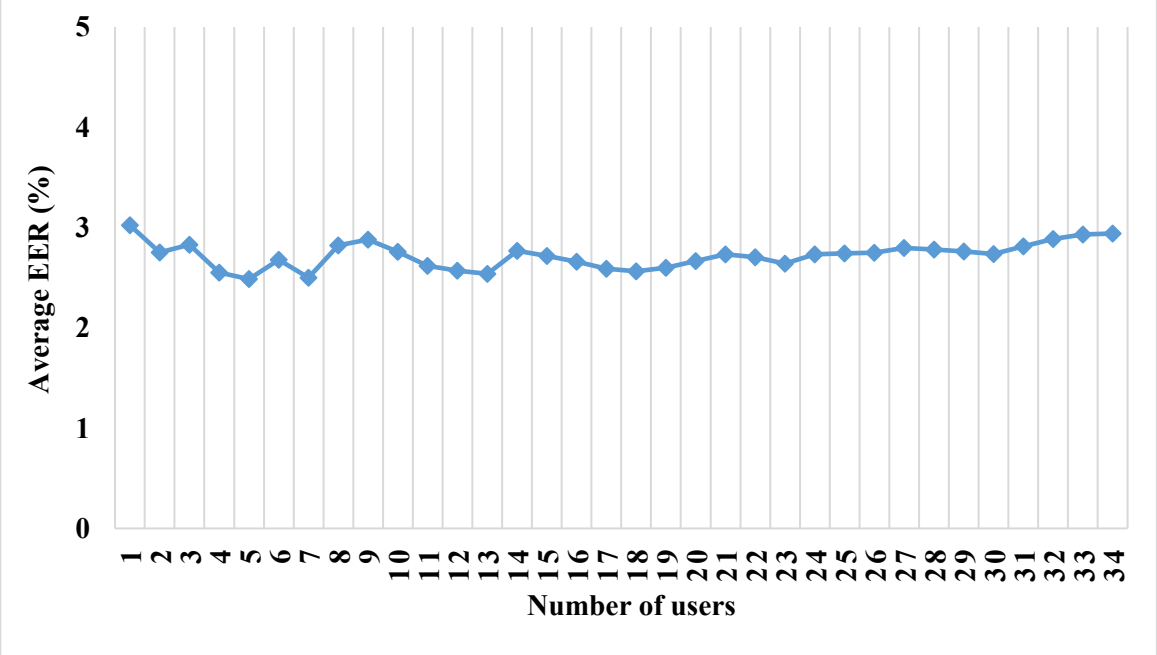

Figure 4.15. Model performance based on the number of enrolled users for the *LiveLab* dataset

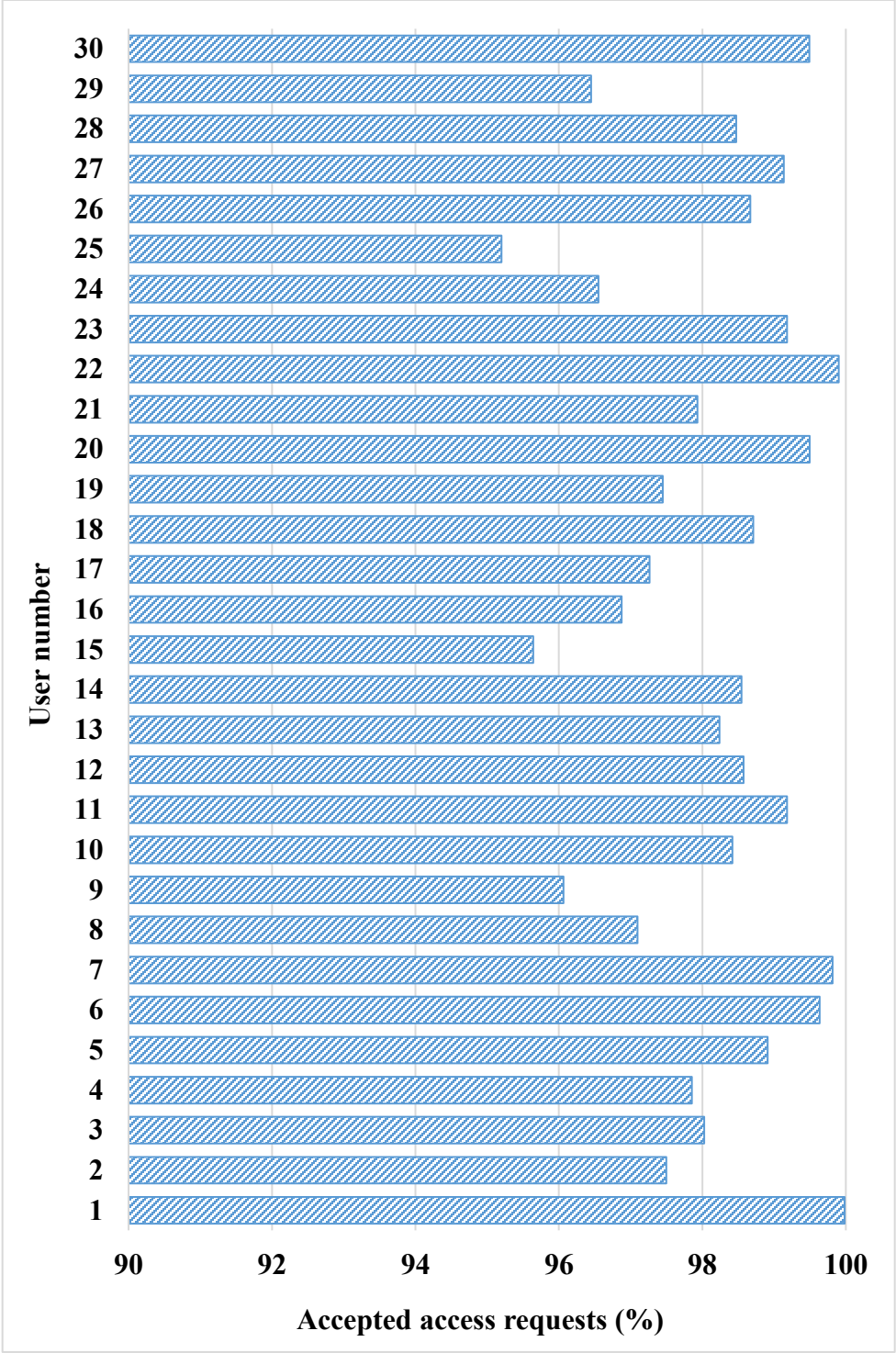

Figure 4.16. Model performance based on simulated access requests for the *UbiqLog4UCI* dataset

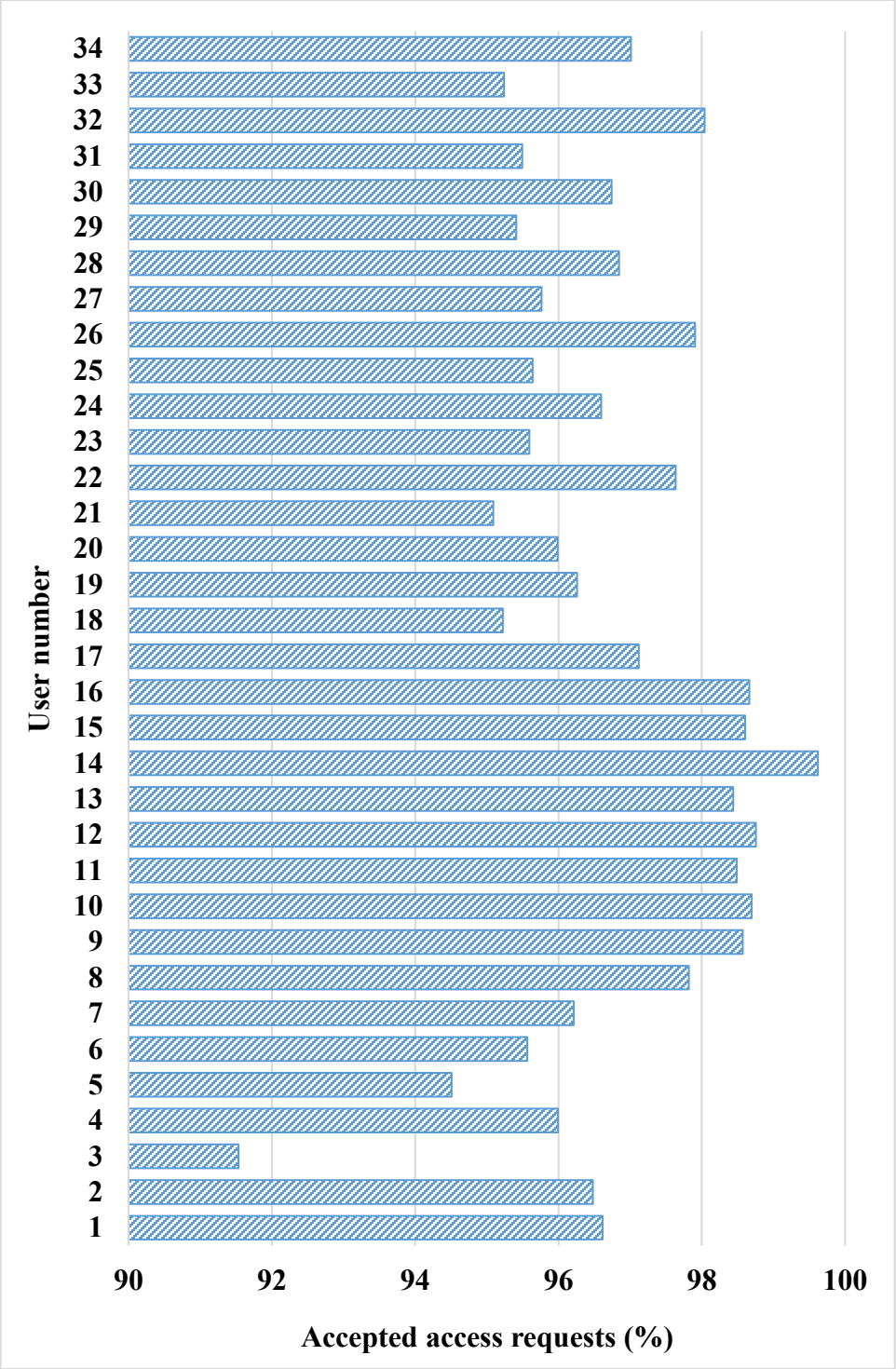

Figure 4.17. Model performance based on simulated access requests for the *LiveLab* dataset

Table 4.10. Comparison with related work

Ref No.	Utilized Information	Validation Dataset(s)	Performance (TPR, FPR and EER) (%)
[25]	Calls, SMS and GPS based location	MIT (one month)	EER=5.4, 35.1 and 35.7
[60]	Calls, SMS, Web browsing history.	Collected by the authors and only include telephone calls, SMSs and Web browsing history.	TPR=99.3, FPR= 0.7, EER=1.6
[85]	Basic apps, telephone calls, SMS and multi-instance (combined basic and extended apps).	MIT (one month)	EER=13.5, 5.4, 2.2 and 9.8
[88]	Unique app usage data	UMDAA-02	Mean EER ≈ 30
		Securacy	Mean EER ≈ 16
[89]	Text entered via soft keyboard, apps including SMS, websites visited, location.	Carried out by the authors for a period of at least one month.	FAR= 30, FRR=18, EER=5 after 1 minute and 1 after 30 minutes
[92]	Most used apps and location.	MIT	EER =9.004
		UCCS carried out by the authors for a period of 4 weeks.	EER= 1.98
Proposed Model	All foreground apps	*UbiqLog4UCI*	FPR=0.91, FNR=4.94, EER=2.92
		LiveLab	FPR=1.03, FNR=4.84, EER=2.94

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 3.2, while the results are shown in Figures 4.16 and 4.17. From these figures, we can see that the minimum percentage 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 percentage 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. 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 4.6 and 4.7. This presented model can be compared and differentiated from other presented research studies in the literature as shown in Table 4.10.

4.4 Security Analysis

Security analysis of possible attacks on the framework are discussed here in order to validate the ability of the framework components to protect against them. In this subsection, we present security analysis of the implemented prototype, the proposed device-to-device message authentication scheme, and the presented app-based user authentication model.

4.4.1 Context-Based Prototype

There are possible scenarios that can take place after the user registration stage and before completing the usage pattern model. The first possible attack is obtaining a user's login and password. In the event that a user's login and password were stolen, compromised, or even loaned out, the other security contexts would still be able to react by denying the new user access from another device identification or scheduling. In addition, the user's login and password are protected in transit to the TS through the use of HTTPS encryption and in the database, using a salted hash function. The second attack scenario is obtaining a user's device. This type of attack could be mitigated by considering the device's location through the IP address and proximity detection. If the device were stolen and not yet reported to the administrator, its absence from the home network or known remote locations would be sufficient to deny access to a malicious user. The other type of attack is brute force, and this type of attack is countered by monitoring and recording both login

attempts and service access attempts, locking out a user's account or device until reviewed by the administrator. For unauthorized modification of external data such as the schedule, a cached copy of the calendar is kept on the TS. In addition, any update in the schedule will be flagged and reported to the administrator, and permission is required to update the cached copy on the TS. Moreover, the assigned weight values can be set by the administrator based on the availability of contextual information. For example, if a user does not have a calendar, the weights can be set based on other available contextual attributes. Furthermore, the time it would take to detect the availability of intrusion is the same as the re-check time. This time is determined based on the average of the previous access sessions of the user.

The weighting of the various contexts also prevents any single data source being compromised from significantly affecting the security of the system as a whole, with the calendar scheduling not explicitly permitting any user access as a prime example. An IP address conflicts or is malformed, or any non-IEEE compliant MAC addresses be other indicators of possible spoofing attempts. Hence, the use of certificate-based authentication would prevent the possibility of the TS being impersonated by an attacker. For data protection, no information related to users is stored on the end-user devices as they are susceptible to loss or theft. This ensures that if the device is compromised, the adversary cannot learn the user profile and simulate the user's behavior to gain system access. Furthermore, all data is verified and replicated to a backup location to avoid both data loss and the need to retrain user profiles, thus preventing a malicious user from rebuilding a user profile to match their own contexts.

4.4.2 Device-to-Device Message Authentication Scheme

The analysis focuses on how the proposed scheme can realize security requirements for smart home IoT device communications. Three main security requirements are considered in the proposed scheme. The first security factor is the authentication. Only the messages signcrypted by a legitimate sender device with ID_{d1} could be unsigncrypted by the intended receiver with the corresponding ID_{d2}. Furthermore, only the *TS* can compute the correct private key of the communicated devices. Thus, the proposed scheme can guarantee the message authentication between the end-user device and the home device as well as with the *TS* since this server is the only one that generates the public parameters and issues the private keys of the involved devices. Additionally, any messages cannot be signed by an adversary without having the sender's secret key S_{d1}. The second security factor is the integrity. Based on the complexity assumptions outlined in (Chapter 3) subsection 3.4, which are provably secure in the random oracle model [144], any intercepted message by an attacker cannot be encrypted without reaching the randomly chosen number $r_i \in Z_p^*$ which is used for each message to be sent. A different random number is included in each new message and in the case of knowing this number, the future sent message will not use the same number, hence will resist against reply attack. As a result, the proposed scheme can protect against active attacks to target data integrity, such as modifying or altering a command during its transmission, thus protecting the integrity of transmitted messages.

Confidentiality is the third security goal. As the private key is generated and issued by the *TS*, which is trusted, and these keys are transferred offline over a secure channel, an attacker cannot access a private key of the connected devices. Thus, since an attacker cannot decrypt

the transmitted message without knowing the receiver's private key, message confidentiality is ensured.

4.4.3 App-Based User Authentication Model

The presented app-based user authentication model is based on user access patterns 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 from the results. 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. However, 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 of statistical analysis of the extracted features. Thus, by including the *app_cont_sq* of newly launched apps, we guarantee that the apps will not be considered in the authentication process until having enough training data. Further, utilizing multi-events (last accessed app patterns) and the *inter_pi* in the access decision will increase security and reduce the chance of the access being sent from an illegitimate user. Additionally, as the authentication takes place, other registered users will be allowed access from registered devices, which increases the usability of the proposed method.

Chapter 5. Conclusion and Future Work

This thesis presents a contextual authentication framework for smart home environments. This framework consists of three main parts: the context-based user authentication method, device-to-device message authentication, and app-based user authentication model. The first part is a context-based authentication method that utilizes contextual information to verify the current user who is trying to access smart home IoT devices from his/her registered end-device (smartphone/tablet). The second part is a device-to-device message authentication scheme to protect transmitted messages among smart home IoT devices. The third part is a user authentication model that builds a user profile based on previous apps' access history in order to make the right decision at the login stage, at subsequent access requests regarding authorized user access and during the access sessions.

From the results obtained, each of the proposed user authentication mechanisms can operate individually and can be complementary to other approaches, according to the situation. In addition, the proposed architecture of the framework can be easily extended to include other authentication mechanisms. The security aspect of the proposed framework is analyzed, and its performance is evaluated in practical scenarios. Additionally, this framework can be adopted in other IoT applications, such as smart medical care, smart grid and smart environmental protection. Furthermore, the framework has the ability to protect home devices against unauthorized access from anonymous and known users, whether locally or remotely, by monitoring all communication to said devices by the TS. Moreover, the framework monitors all access-related activities, such as attempted logins, service requests, and access durations, storing them in a database for future analysis by establishing usage patterns and detecting brute force attempts and other anomalies.

Additionally, the proposed IBS scheme, as compared to other schemes, can achieve the security requirements for device-to-device communication in the smart home. Based on the security analysis and assumptions, the presented IBS scheme can efficiently achieve the security requirements for device-to-device communication in the smart home. Ultimately, the framework provides user authentication both at the point-of-entry and during the access session in addition to providing secure communication between the home IoT devices. In addition, the statistical analysis of the extracted features, show that the adoption of decision-making based on last app events leads to high accuracy and that considering the *inter_pi* and *intra_pi* features increases the level of security and reduces the chance of the network being accessed by unauthorized users. Hence, this approach will provide transparent user authentication as users interact with apps and services.

The next subsections summarise the contributions of this thesis and the directions for future work.

5.1 Conclusion

This thesis provides a contextual authentication framework that considers context-based user authentication, device-to-device message authentication, and app-based user authentication. The contributions of this thesis are:

1. A literature review and classification of authentication mechanisms that can be adopted or presented for smart homes. In addition to providing a discussion of the limitations of the existing approaches with regard to security and usability, this review presents proposals for: smart home security frameworks; pattern-based authentication

approaches for mobile phone users; and public cryptographic mechanisms to secure communication between smart home IoT devices.

2. The design, implementation and evaluation of a proof of concept prototype, examination of the available contextual information with regards to its permission requirements and retrieval time, and utilization of this contextual information for user authentication. The objective of the proposed framework is to increase security in addition to delivering usability and protecting user information. Moreover, information will not be saved on the user's device, but on a TS by applying security protection for data storage as well as a backup server. Thus, in the case of losing the end-device, the user will not lose any information.
3. The design and evaluation of an efficient and secure scheme for information exchange among smart home IoT devices. This scheme is efficient regarding cipher-text length and computational cost. During the authentication process, the proposed scheme does not require access to the TS; access to this server is only needed at the time of registration or for updating secret keys. In addition to providing authentication, the proposed scheme provides integrity and confidentiality as well as the ability to protect communication among devices against various possible attacks. Compared to other schemes, the proposed scheme can efficiently achieve the security requirements for device-to-device communication in the smart home.
4. The design and evaluation of an app-based user authentication model to protect smart homes from unauthorized access in addition to protecting the access of users who have weak or no security protection on their end-devices. This model adapts to app usage changes, such as using new apps or stopping the use of others. This model is able to

authenticate registered users utilizing app interactions, as well as generated network traffic on mobile devices, with considerably high accuracy. The proposed model does not require specific action from the user in order to be authenticated; rather, it is based on regular actions while accessing apps, which enhances usability to users. The presented authentication method is performed in the background, based on the user's general access routine. In order to improve the usability and efficiency of the proposed approach, only minimal features are utilized.

In conclusion, the proposed architecture of the framework can be easily extended to include other authentication mechanisms. It can also be concluded that the security measures do not impose significant connection overhead, which amounts to the system acting almost entirely transparently. The security aspect of the proposed framework is analyzed, and its performance is evaluated in practical scenarios. Furthermore, this framework can be adopted in other IoT applications, such as smart medical care, smart agriculture, smart grid, and smart environmental protection. Ultimately, the framework provides user authentication both at the point-of-entry and during the access session in addition to providing secure communication between the home IoT devices.

5.2 Future Work

The limitations of the proposed framework are:

1. Historical data retrieved from end-user devices and home environments, including sensor measurements such as temperature and other data measures that can enhance the proposed behavior-based model, are not considered.
2. The proposed app-based authentication model is based on frequent access patterns for foreground apps and is limited to discovering foreground app access patterns.

Background apps are not considered since they have inconsistent access patterns. In addition, the modeling of background apps needs more data and the currently utilized datasets have limited background events which, in turn, are difficult to model. Since the proposed model cannot detect rare behavior patterns such as using an app only once (or limited time), it will instead request a second-factor authentication from the user, which may affect the usability.

3. In the presented identity-based signcryption (IBS) scheme, the trusted server (TS) is assumed to be trusted. However, considering that IBE systems avoid using certificates and simplify certificate management, the key escrow problem can be considered as a limitation in cases where the TS is attacked.

To address the limitations, future work includes:

1. Measuring the quality of the retrieved information can explore the possibility of further analysis of the historical data (long-term contextual information) retrieved from end-user devices and home environments, and include other data measures such as sensors, in order to generate higher fidelity user profiles using machine learning and other techniques.
2. Investigation modeling of the usage access pattern to background apps can enhance the user authentication model. In addition, examining rare behavior patterns, such as using limited accessed interaction to specific apps, may improve the usability of the proposed model.
3. For the presented IBS scheme, an investigation into the ability to apply the certificate-less technique to situations where the TS joins in the communication process will be employed.

References

[1] D. Marikyan, S. Papagiannidis, and E. Alamanos, "A Systematic Review of the Smart Home Literature: A User Perspective," *Technol. Forecast. Soc. Change*, vol. 138, pp. 139–154, 2019.

[2] E. Fernandes, J. Jung, and A. Prakash, "Security Analysis of Emerging Smart Home Applications," in *EEE Symposium on Security and Privacy, SP 2016*, 2016, pp. 636–654.

[3] H. Thapliyal, R. K. 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, 2017.

[4] ICTC, "The Application of Everything: Canada's Apps Economy Value Chain," 2014.

[5] A. Sharma and S. K. Sahay, "Group-Wise Classification Approach to Improve Android Malicious Apps Detection Accuracy," *arXiv Prepr. arXiv1904.02122*, 2019.

[6] H. Saevanee, N. L. Clarke, and S. M. Furnell, "Multi-Modal Behavioural Biometric Authentication for Mobile Devices," *IFIP Adv. Inf. Commun. Technol.*, pp. 465–474, 2012.

[7] E. J. Hom, "Mobile Device Security: Startling Statistics on Data Loss and Data Breaches | The ChannelPro Network," 2017. [Online]. Available: http://www.channelpronetwork.com/. [Accessed: 20-Aug-2017].

[8] M. B. Barcena and C. Wueest, "Insecurity in the Internet of Things." Security response, symantec, 2016.

[9] A. Gheorghe, "The Internet of Things: Risk in the Connected Home," *Bitdefender*, 2016.

[10] M. Abomhara, "Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks," *J. Cyber Secur. Mobil.*, vol. 4, no. 1, pp. 65–88, 2015.

[11] B. Ur, J. Jung, and S. Schechter, "The Current State of Access Control for Smart Devices in Homes," *Work. Home Usable Priv. Secur.*, pp. 1–6, 2014.

[12] Hewlett Packard, "Internet of Things Research Study," *HP*, p. 6, 2015.

[13] P. Ruggiero and J. Foote, "Cyber Threats to Mobile Phones Mobile Threats Are

Increasing.” US-CERT, pp. 1–6, 2014.

[14] R. Beek, Christiaan; Dunton, Taylor; Fokker, John; Grobman, Steve; Hux, Tim; Polzer, Tim; Lopez, Marc Rivero; Roccia, Thomas; Saavedra-Morales, Jessica; Samani, Raj; Sherstobitof, “McAfee Labs Threats Report,” *McAfee, Santa Clara, CA, USA, Technical Report (2019).* pp. 1–41.

[15] F. Zindi, “Mobile Telecommunications Security Threat Landscape.” GSMA, pp. 1–20, 2019.

[16] M. Alshahrani, I. Traore, and I. Woungang, “Design and Implementation of a Lightweight Authentication Framework for the Internet of Things (IoT),” in *IEEE 6th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, 2019, pp. 185–194.

[17] “Symantec Internet Security Threat Report,” *Symantec Corp.*, vol. 24, 2019.

[18] N. Alkaldi and K. Renaud, “Why Do People Adopt, or Reject, Smartphone Password Managers?,” in *EuroUSEC , Darmstadt, Germany*, 2016, pp. 1–14.

[19] Z. Hills, D. F. Arppe, A. Ibrahim, and K. El-Khatib, “Compound Password System for Mobile,” in *IEEE International Conference on Cyber Situational Awareness, Data Analytics and Assessment, CyberSA 2018*, 2018, pp. 1–4.

[20] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, “uWave: Accelerometer-Based Personalized Gesture Recognition and its Applications,” *Pervasive Mob. Comput.*, vol. 5, no. 6, pp. 657–675, 2009.

[21] L. Li, X. Zhao, and G. Xue, “Unobservable Re-authentication for Smartphones,” *NDSS - Netw. Distrib. Syst. Secur. Symp.*, vol. 56, 2013.

[22] C. Nickel, T. Wirtl, and C. Busch, “Authentication of Smartphone Users Based on the Way They Walk Using K-NN Algorithm,” *Intell. Inf. Hiding Multimed. Signal Process. (IIH-MSP), Eighth Int. Conf. IEEE,* pp. 16–20, 2012.

[23] M. Trojahn and F. Ortmeier, “Toward Mobile Authentication with Keystroke Dynamics on Mobile Phones and Tablets,” *27th Int. Conf. Adv. Inf. Netw. Appl. Work.*, pp. 697–702, 2013.

[24] J. Zhu, P. Wu, X. Wang, and J. Zhang, “SenSec: Mobile Security Through Passive Sensing,” *IEEE Int. Conf. Comput. Netw. Commun.*, pp. 1128–1133, 2013.

[25] F. Li, N. Clarke, M. Papadaki, and P. Dowland, “Behaviour Profiling for

Transparent Authentication for Mobile Devices," *Eur. Conf. Cyber Warf. Secur. Acad. Conf. Int. Ltd.*, pp. 307–315, 2011.

[26] M. Rahman and K. El-Khatib, "Secure Time Synchronization for Wireless Sensor Networks Based on Bilinear Pairing Functions," *IEEE Trans. Parallel Distrib. Syst.*, 2010.

[27] 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.

[28] A. Aupy and N. Clarke, "User Authentication by Service Utilisation Profiling," *Adv. Netw. Commun. Eng. 2*, no. 18, 2005.

[29] S. Yazji, X. Chen, R. P. Dick, and P. Scheuermann, "Implicit User Re-Authentication for Mobile Devices," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5585 LNCS, pp. 325–339, 2009.

[30] M. Ben Salem and S. J. Stolfo, "Modeling User Search Behavior for Masquerade Detection," in *International Workshop on Recent Advances in Intrusion Detection*, 2011, vol. Springer, pp. 181–200.

[31] Y. (Catherine) Yang, "Web User Behavioral Profiling for User Identification," *Decis. Support Syst.*, vol. 49, no. 3, pp. 261–271, 2010.

[32] M. Abramson and D. W. Aha, "User Authentication from Web Browsing Behavior," in *The Twenty-Sixth International FLAIRS Conference*, 2013, pp. 268–273.

[33] 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, 2018.

[34] A. Forget, "A World with Many Authentication Schemes," *Doctoral dissertation, Carleton University Ottawa*. 2012.

[35] W. Anani and A. Ouda, "The Importance of Human Dynamics in the Future User authentication," in *IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2017, pp. 1–5.

[36] A. Ouda, "A Framework for Next Generation User Authentication," in *IEEE 3rd MEC International Conference on Big Data and Smart City (ICBDSC)*, 2016, pp. 1–4.

[37] T. Fandakly and N. Caporusso, “Beyond Passwords: Enforcing Username Security as the First Line of Defense,” in *International Conference on Applied Human Factors and Ergonomics. Springer, Cham*, 2019, pp. 48–58.

[38] L. Li, B. Pal, J. Ali, N. Sullivan, R. Chatterjee, and T. Ristenpart, “Protocols for Checking Compromised Credentials,” in *Proceedings of the ACM Conference on Computer and Communications Security*, 2019, pp. 1387–1403.

[39] E. Shi, Y. Niu, M. Jakobsson, and R. Chow, “Implicit Authentication Through Learning User Behavior,” *Springer, Berlin, Heidelberg.*, pp. 99–113, 2011.

[40] A. AI Abdulwahid, N. Clarke, I. Stengel, S. Furnell, and C. Reich, “Continuous and Transparent Multimodal Authentication: Reviewing the State of the Art,” *Cluster Comput.*, vol. 19, no. 1, pp. 455–474, 2016.

[41] I. Agudo, R. Rios, and J. Lopez, “A Privacy-Aware Continuous Authentication Scheme for Proximity-Based Access Control,” *Comput. Secur.*, vol. 39, pp. 117–126, 2013.

[42] M. D. Corner and B. D. Noble, “Protecting Applications with Transient Authentication,” *Proc. 1st Int. Conf. Mob. Syst. Appl. Serv.*, pp. 57–70, 2003.

[43] F. Aloul, S. Zahidi, and W. El-Hajj, “Two Factor Authentication Using Mobile Phones,” *Comput. Syst. Appl. 2009. AICCSA 2009. IEEE/ACS Int. Conf.*, pp. 641–644, 2009.

[44] K. Mock, J. Weaver, and M. Milton, “Poster : Real-time Continuous Iris Recognition for Authentication Using an Eye Tracker,” *Ccs*, pp. 1007–1009, 2012.

[45] P. W. Tsai, M. K. Khan, J. S. Pan, and B. Y. Liao, “Interactive Artificial Bee Colony Supported Passive Continuous Authentication System,” *IEEE Syst. J.*, vol. 8, no. 2, pp. 395–405, 2014.

[46] K. El-Khatib, Z. E. Zhang, N. Hadibi, and G. V. Bochmann, “Personal and Service Mobility in Ubiquitous Computing Environments,” *Wirel. Commun. Mob. Comput.*, vol. 4, no. 6, pp. 595–607, 2004.

[47] A. K. Dey and G. D. Abowd, “Towards a Better Understanding of Context and Context-Awareness,” *Comput. Syst.*, vol. 40, no. 3, pp. 304–307, 1999.

[48] M. J. Covington, M. R. Sastry, and D. J. Manohar, “Attribute-Based Authentication Model for Dynamic Mobile Environments,” in *International Conference on Security*

in Pervasive Computing., 2006, vol. Springer, pp. 227–242.

[49] K. Zhou and J. Ren, "PassBio: Privacy-Preserving User-Centric Biometric Authentication," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 12, pp. 3050–3063, 2018.

[50] W. Qin, D. Zhang, Y. Shi, and K. Du, "Combining User Profiles and Situation Contexts for Spontaneous Service Provision in Smart Assistive Environments," in *International Conference on Ubiquitous Intelligence and Computing, Springer, Berlin, Heidelberg*, 2008, pp. 187–200.

[51] K. Henricksen, "A framework for Context-Aware Pervasive Computing Applications," *Sch. Inf. Technol. Electr. Eng. Univ. Queensl.*, 2003.

[52] B. Schilit, N. Adams, and R. Want, "Context-Aware Computing Applications," *First Work. Mob. Comput. Syst. Appl.*, pp. 85–90, 1994.

[53] C. Perera, S. Member, A. Zaslavsky, and P. Christen, "Context Aware Computing for The Internet of Things : A Survey," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[54] R. Jalali, K. El-Khatib, and C. McGregor, "Smart City Architecture for Community Level Services Through the Internet of Things," in *18th IEEE International Conference on Intelligence in Next Generation Networks, ICIN*, 2015, pp. 108–113.

[55] W. S. Lima, E. Souto, K. El-Khatib, R. Jalali, and J. Gama, "Human Activity Recognition Using Inertial Sensors in a Smartphone: An Overview," *Sensors (Switzerland)*, vol. 19, no. 14, 3213, 2019.

[56] D. C. Nazário, I. V. B. Tromel, M. A. R. Dantas, and J. L. Todesco, "Context Management: Toward Assessing Quality of Context Parameters in a Ubiquitous Assisted Environment," *JISTEM-Journal Inf. Syst. Technol. Manag.*, vol. 11, no. 3, pp. 569–590, 2014.

[57] A. Manzoor, H. L. Truong, and S. Dustdar, "On The Evaluation of Quality of Context," in *European Conference on Smart Sensing and Context, Springer Berlin Heidelberg*, 2008, pp. 140–153.

[58] K. Benzekki, A. El Fergougui, and A. E. B. Elalaoui, "A context-Aware Authentication System for Mobile Cloud Computing," in *Procedia Computer Science*, 2018, pp. 379–387.

[59] A. Mhamed, P. E. Abi-char, and B. E. M. Mokhtari, “A Dynamic Trust-Based Context-Aware Authentication Framework with Privacy Preserving,” *Int. J. Comput. Netw. Secur.*, vol. 2, no. 2, pp. 87–102, 2010.

[60] 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. Networks*, vol. 5, no. 1, pp. 3–14, 2012.

[61] D. M. Dobkin and B. Aboussouan, “Low Power Wi-Fi™(IEEE802. 11) for IPsmart Objects,” *GainSpan Corp.*, 2009.

[62] T. Bhattasali, “LICRYPT: Lightweight Cryptography Technique for Securing Smart Objects in Internet of Things Environment,” *J. CSI Commun.*, no. May, pp. 26–28, 2013.

[63] J. Ayuso, L. Marin, A. Jara, and A. Skarmeta, “Optimization of Public Key Cryptography (RSA and ECC) for 16-bits Devices Based on 6LoWPAN,” in *1st International Workshop on the Security of the Internet of Things, Tokyo, Japan*, 2010, pp. 1–8.

[64] S. Al Salami, J. Baek, K. Salah, and E. Damiani, “Lightweight Encryption for Smart Home,” in *11th International Conference on Availability, Reliability and Security (ARES),* 2016, pp. 382–388.

[65] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, “Radio-Telepathy: Extracting a Secret Key From an Unauthenticated Wireless Channel,” in *14th ACM international conference on Mobile computing and networking*, 2008, pp. 128–139.

[66] K. Han, J. Kim, T. Shon, and D. Ko, “A Novel Secure Key Paring Protocol for RF4CE Ubiquitous Smart Home Systems,” *Pers. Ubiquitous Comput.*, vol. 17, no. 5, pp. 945–949, 2013.

[67] E. Ayday and S. Rajagopal, “Secure Device Authentication Mechanisms for the Smart Grid-Enabled Home Area Networks,” *Technical Report*. pp. 1–18, 2013.

[68] Y. Chen and B. Luo, “S2A: Secure Smart Household Appliances,” in *the second ACM conference on Data and Application Security and Privacy*, 2012, pp. 217–228.

[69] Y. Li, “Design of a Key Establishment Protocol for Smart Home Energy Management System,” in *Fifth International Conference on Computational Intelligence, Communication Systems and Networks*, 2013, pp. 88–93.

[70] M. M. Fouda, Z. M. Fadlullah, N. Kato, R. Lu, and X. S. Shen, "A Lightweight Message Authentication Scheme for Smart Grid Communications," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 675–685, 2011.

[71] A. C. Jose, R. Malekian, and N. Ye, "Improving Home Automation Security; Integrating Device Fingerprinting into Smart Home," *IEEE Access*, vol. 4, pp. 5776–5787, 2016.

[72] D. M. Konidala, D.-Y. Kim, C.-Y. Yeun, and B.-C. Lee, "Security Framework for RFID-Based Applications in Smart Home Environment," *J. Inf. Process. Syst.*, vol. 7, no. 1, pp. 111–120, 2011.

[73] G. Dua, N. Gautam, D. Sharma, and A. Arora, "Replay Attack Prevention in Kerberos Authentication Protocol using Triple Password," *Int. J. Comput. Networks Commun.*, vol. 5, no. 2, pp. 59–70, 2013.

[74] D. Hardt, "The OAuth 2.0 Authorization Framework," *Microsoft, RFC 6749*. pp. 1–76, 2012.

[75] E. Ghazizadeh, Z. S. Shams Dolatabadi, R. Khaleghparast, M. Zamani, A. A. Manaf, and M. S. Abdullah, "Secure openID Authentication Model by using Trusted Computing," *Abstr. Appl. Anal.*, 2014.

[76] K. Bicakci, D. Unal, N. Ascioglu, and O. Adalier, "Mobile Authentication Secure Against Man-In-The-Middle Attacks," in *IEEE 2nd International Conference on Mobile Cloud Computing, Services, and Engineering*, 2014, pp. 273–276.

[77] W. Lee and R. Lee, "Implicit Sensor-Based Authentication of Smartphone Users with Smartwatch," in *the Hardware and Architectural Support for Security and Privacy. ACM*, 2016, pp. 1–8.

[78] W. Meng, D. S. Wong, S. Furnell, and J. Zhou, "Surveying the Development of Biometric User Authentication on Mobile Phones," *IEEE Commun. Surv. Tutorials.*, vol. 17(3), no. 3, pp. 1268–1293, 2015.

[79] F. Li, "Behaviour Profiling for Mobile Devices," *PhD thesis, the Plymouth University*. pp. 1–216, 2012.

[80] S. Furnell, N. Clarke, and S. Karatzouni, "Beyond the PIN: Enhancing User Authentication for Mobile Devices," *Comput. Fraud Secur.*, no. 8, pp. 12–17, 2008.

[81] P. Gosset, "Fraud Detection Concepts : Final Report," *CiteSeer, Doc Ref.*

AC095/VOD/W22/DS/P/18/1, pp. 1–27, 1998.

[82] R. Murmuria, A. Stavrou, D. Barbará, and D. Fleck, "Continuous Authentication on Mobile Devices Using Power Consumption, Touch Gestures and Physical Movement of Users," in *International Workshop on Recent Advances in Intrusion Detection, Springer, Cham.*, 2015, pp. 405–424.

[83] M. Jakobsson, E. Shi, P. Golle, and R. Chow, "Implicit Authentication for Mobile Devices," in *the 4th USENIX conference on Hot topics in security*, 2011, pp. 1–6.

[84] F. Li, N. Clarke, M. Papadaki, and P. Dowland, "Behaviour Profiling on Mobile Devices," in *IEEE International Conference on Emerging Security Technologies (EST),* 2010, pp. 77–82.

[85] 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, 2014.

[86] Y. Ashibani and Q. H. Mahmoud, "A Machine Learning-Based User Authentication Model Using Mobile App Data," in *International Conference on Intelligent and Fuzzy Systems (INFUS)*, 2019, pp. 408–415, (Best Paper Award).

[87] D. Bassu, M. Cochinwala, and A. Jain, "A New Mobile Biometric Based Upon Usage Context," in *IEEE International Conference on Technologies for Homeland Security, HST*, 2013, pp. 441–446.

[88] 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, 2019.

[89] 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, 2017.

[90] Z. M. Zhang, Lide; Tiwana, Birjodh; Qian, Zhiyun;Wang, Zhaoguang;Dick, Robert P.; Mao and L. Yang, "Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones," in *the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis.*, 2010, pp. 105–114.

[91] 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 Secur.*, vol. 8, no. 1, pp. 136–148, 2013.

[92] A. A. Alzubaidi, "Continuous Authentication of Smartphone Owners Based on App Access Behavior," University of Colorado Colorado Springs, 2018.

[93] Y. Ashibani and Q. H. Mahmoud, "A Behavior-Based Proactive User Authentication Model Utilizing Mobile Application Usage Patterns," in *32nd Canadian Conference on Artificial Intelligence*, 2019, pp. 284–295.

[94] Y. Ashibani and Q. H. Mahmoud, "User authentication for smart home networks based on mobile apps usage," in *28th IEEE International Conference on Computer Communications and Networks, ICCCN*, 2019, pp. 1–6.

[95] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Cell Phone-Based Biometric Identification," *4th Int. Conf. Biometrics Theory, Appl. Syst. BTAS , IEEE*, pp. 1–7, 2010.

[96] K. Chang, J. Hightower, and B. Kveton, "Inferring Identity Using Accelerometers in Television Remote Controls," in *International Conference on Pervasive Computing, Springer, Berlin, Heidelberg.*, 2009, pp. 151–167.

[97] C. Paniagua, H. Flores, and S. N. Srirama, "Mobile Sensor Data Classification for Human Activity Recognition using MapReduce on Cloud," in *9th International Conference on Mobile Web Information Systems (MobiWIS 2012), Procedia Computer Science*, 2012, vol. 10, pp. 585–592.

[98] J. Guerra-Casanova, C. Sánchez-Ávila, G. Bailador, and A. de Santos Sierra, "Authentication in Mobile Devices Through Hand Gesture Recognition," *Int. J. Inf. Secur.*, vol. 11, no. 2, pp. 65–83, 2012.

[99] W. Lee and R. B. Lee, "Multi-sensor Authentication to Improve Smartphone Security," in *IEEE International Conference on Information Systems Security and Privacy (ICISSP)*, 2015, pp. 1–11.

[100] W.-H. Lee and R. B. Lee, "Implicit Authentication for Smartphone Security," in *International Conference on Information Systems Security and Privacy. Springer*, 2015, pp. 160–176.

[101] K. Zhou, J. Medsger, A. Stavrou, and J. M. Voas, "Mobile Application and Device Power Usage Measurements," in *IEEE Sixth International Conference on Software*

Security and Reliability (SERE), IEEE, 2012, pp. 147–156.

[102] A. Shye, B. Scholbrock, and G. Memik, "Into the Wild: Studying Real User Activity Patterns to Guide Power Optimizations for Mobile Architectures," in *42nd Annual IEEE/ACM International Symposium on Microarchitecture.*, 2009, pp. 168–178.

[103] M. S. Obaidat, I. Traore, and I. Woungang, "Continuous Authentication Using Writing Style," in *Biometric-Based Physical and Cybersecurity Systems. Springer, Cham*, 2019, pp. 211–232.

[104] D. Damopoulos, G. Kambourakis, and S. Gritzalis, "From Keyloggers to Touchloggers: Take the Rough with the Smooth," *Comput. Secur.*, vol. 32, pp. 102–114, 2013.

[105] N. Sae-Bae, N. Memon, K. Isbister, and K. Ahmed, "Multitouch Gesture-Based Authentication," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 4, pp. 568–582, 2014.

[106] H. M. Thang, V. Q. Viet, N. Dinh Thuc, and D. Choi, "Gait Identification Using Accelerometer on Mobile Phone," in *IEEE International Conference on Control, Automation and Information Sciences (ICCAIS)*, 2012, pp. 344–348.

[107] Sangil Choi, Ik-Hyun Youn, R. LeMay, S. Burns, and J.-H. Youn, "Biometric Gait Recognition Based on Wireless Acceleration Sensor Using K-Nearest Neighbor Classification," in *International Conference on Computing, Networking and Communications (ICNC)*, 2014, pp. 1091–1095.

[108] A. Kale, A. N. Rajagopalan, N. Cuntoor, and V. Krüger, "Gait-Based Recognition of Humans Using Continuous HMMs," in *IEEE 5th International Conference on Automatic Face Gesture Recognition, FGR*, 2002, pp. 336–341.

[109] D. Gafurov, K. Helkala, and T. Søndrol, "Biometric Gait Authentication Using Accelerometer Sensor," *J. Comput.*, vol. 1, no. 7, pp. 51–59, 2006.

[110] X. Zhao, T. Feng, and W. Shi, "Continuous Mobile Authentication Using a Novel Graphic Touch Gesture Feature," in *IEEE 6th International Conference on Biometrics: Theory, Applications and Systems, BTAS*, 2013.

[111] M. Wolff and University, "Behavioral Biometric Identificatio on Mobile Devices," in *International Conference on Augmented Cognition, Springer, Berlin, Heidelberg.*, 2013, pp. 783–791.

[112] I. Traore, I. Woungang, M. S. Obaidat, Y. Nakkabi, and I. Lai, "Combining Mouse

and Keystroke Dynamics Biometrics for Risk-Based Authentication in Web Environments," in *IEEE 4th International Conference on Digital Home, ICDH*, 2012, pp. 138–145.

[113] Cheng-Jung Tasia, T.-Y. Chang, P.-C. Cheng, and J.-H. Lin, "Two Novel Biometric Features in Keystroke Dynamics Authentication Systems for Touch Screen Devices," *Secur. Commun. Networks*, vol. 7, no. 4, pp. 750–758, 2014.

[114] C. C. Lin, C. C. Chang, D. Liang, and C. H. Yang, "A New Non-Intrusive Authentication Method Based on the Orientation Sensor for Smartphone Users," in *IEEE 6th International Conference on Software Security and Reliability*, 2012, pp. 245–252.

[115] H. Kayacık, M. Just, L. Baillie, D. Aspinall, and N. Micallef, "Data Driven Authentication: On the Effectiveness of User Behaviour Modelling with Mobile Device Sensors," *Mob. Secur. Technol.*, pp. 1–11, 2014.

[116] W. Shi, J. Yang, Y. Jiang, F. Yang, and Y. Xiong, "SenGuard: Passive User Identification on Smartphones Using Multiple Sensors," in *IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2011, pp. 141–148.

[117] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon, "Biometric-Rich Gestures: A Novel Approach to Authentication on Multi-touch Devices," in *SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 977–986.

[118] J. Sun, R. Zhang, J. Zhang, and Y. Zhang, "TouchIn: Sightless Two-Factor Authentication on Multi-Touch Mobile Devices," in *IEEE Conference on Communications and Network Security*, 2014, pp. 436–444.

[119] F. Monrose, M. K. Reiter, and S. Wetzel, "Password Hardening Based on Keystroke Dynamics," *Int. J. Inf. Secur.*, vol. 1, no. 2, pp. 69–83, 2002.

[120] T. J. Hazen, E. Weinstein, B. Heisele, A. Park, and J. Ming, "Multi-Modal Face and Speaker Identification for Mobile Devices," *Face Biometrics Pers. Identification, Springer Berlin Heidelberg.*, pp. 123–183, 2007.

[121] J. Angulo and E. Wästlund, "Exploring Touch-Screen Biometrics for User Identification on Smart Phones," *Priv. Identity Manag. Lif, . Springer*, pp. 130–143, 2012.

[122] L. Xu, X. Zheng, W. Guo, and G. Chen, “A Cloud-Based Monitoring Framework for Smart Home,” in *IEEE 4th International Conference on Cloud Computing Technology and Science Proceedings*, 2012, pp. 805–810.

[123] W. M. Kang, S. Y. Moon, and J. H. Park, “An Enhanced Security Framework for Home Appliances in Smart Home,” *Human-centric Comput. Inf. Sci.*, vol. 7, no. 1, pp. 1–12, 2017.

[124] D. Damopoulos, G. Kambourakis, and G. Portokalidis, “The Best of Both Worlds: A Framework for the Synergistic Operation of Host and Cloud Anomaly-based IDS for Smartphones,” in *Proceedings of the Seventh European Workshop on System Security*, 2014, pp. 1–6.

[125] Y. Ashibani and Q. H. Mahmoud, “A Behavior Profiling Model for User Authentication in IoT Networks based on App Usage Patterns,” in *44th IEEE Annual Conference of the Industrial Electronics Society (IECON)*, 2018, pp. 2841–2846.

[126] A. Ceccarelli, L. Montecchi, F. Brancati, P. Lollini, A. Marguglio, and A. Bondavalli, “Continuous and Transparent User Identity Verification for Secure Internet Services,” *IEEE Trans. Dependable Secur. Comput.*, vol. 12, no. 3, pp. 270–283, 2015.

[127] N. A. Mahadi, M. A. Mohamed, A. I. Mohamad, M. Makhtar, M. F. A. Kadir, and M. Mamat, “A Survey of Machine Learning Techniques for Behavioral-Based Biometric User Authentication,” *Recent Adv. Cryptogr. Netw. Secur.*, 2018.

[128] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley, “A Brief Survey of Machine Learning Methods and Their Sensor and IoT Applications,” in *8th IEEE International Conference on Information, Intelligence, Systems and Applications, (IISA) 2017*, 2017, pp. 1–8.

[129] Y. Qi, “Random Forest for Bioinformatics,” *Ensemble Mach. Learn. Methods Appl.*, pp. 307–323, 2012.

[130] J. Xia, P. Ghamisi, N. Yokoya, and A. Iwasaki, “Random Forest Ensembles and Extended Multiextinction Profiles for Hyperspectral Image Classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 202–216, 2017.

[131] I. I. Abu Sulayman and A. Ouda, “Data Analytics Methods for Anomaly Detection: Evolution and Recommendations,” in *IEEE International Conference on Signal*

Processing and Information Security (ICSPIS), 2018, pp. 1–4.

[132] Bosubabu Sambana, "A Survey on Machine Learning: Concept, Algorithms and Applications," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 5, no. 2, pp. 1301–1309, 2017.

[133] D. Naboulsi, M. Fiore, S. Ribot, and R. Stanica, "Large-Scale Mobile Traffic Analysis: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 124–161, 2016.

[134] 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 *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2013, pp. 275–284.

[135] V. Srinivasan, S. Moghaddam, A. Mukherji, K. K. Rachuri, C. Xu, and E. M. Tapia, "MobileMiner: Mining your Frequent Patterns on your Phone," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014, pp. 389–400.

[136] Y. Ashibani and Q. H. Mahmoud, "Cyber Physical Systems Security: Analysis, Challenges and Solutions," *J. Comput. Secur. Elsevier*, vol. 68, pp. 81–97, 2017.

[137] M. Guennoun and K. El-Khatib, "Securing Medical Data in Smart Homes," in *2009 IEEE International Workshop on Medical Measurements and Applications (MeMeA) 2009*, 2009, pp. 104–107.

[138] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A Vision of IoT: Applications, challenges, and Opportunities With China Perspective," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 349–359, 2014.

[139] P. N. Mahalle, B. Anggorojati, N. R. P. Prasad, and R. Prasad, "Identity Authentication and Capability Based Access Control (IACAC) for the Internet of Things," *J. Cyber Secur. Mobil.*, vol. 1, no. 4, pp. 309–348, 2013.

[140] I. Muslukhov, Y. Boshmaf, C. Kuo, J. Lester, and K. Beznosov, "Know Your Enemy: The Risk of Unauthorized Access in Smartphones by Insiders," in *15th International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2013, pp. 271–280.

[141] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba, "An Efficient Secure Distributed

Anonymous Routing Protocol for Mobile and Wireless ad hoc Networks," *Comput. Commun.*, vol. 28, no. 10, pp. 1193–1203, 2005.

[142] A. Ibrahim and A. Ouda, "A Hybrid-Based Filtering Approach for User Authentication," in *IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2017, pp. 1–5.

[143] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Springer Science & Business Media, 2006.

[144] D. Boneh and M. Franklin, "Identity-Based Encryption From the Weil Pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.

[145] Y. Ashibani and Q. H. Mahmoud, "Classification and Feature Extraction for User Identification for Smart Home Networks Based on Apps Access History," in *18th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2019, pp. 376–380.

[146] A. Girardello and F. Michahelles, "Explicit and Implicit Ratings for Mobile Applications," *GI-Jahrestagung*, pp. 606–612, 2010.

[147] 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, 2011.

[148] Y. Ashibani and Q. H. Mahmoud, "A User Authentication Model for IoT Networks Based on App Traffic Patterns," in *9th IEEE Annual I Information Technology; Electronics and Mobile Communication Conference (IEMCON)*, 2018, pp. 632–638.

[149] Y. Ashibani, D. Kauling, and Q. H. Mahmoud, "Poster: A Context-Aware Authentication Service for Smart Homes," in *14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2017, pp. 588–589.

[150] "Linksys E1200 N300 Wireless Router." [Online]. Available: http://www.linksys.com/ca/p/P-E1200/. [Accessed: 15-Jul-2018].

[151] "DD-WRT." [Online]. Available: http://www.dd-wrt.com/site/index. [Accessed: 15-Jul-2018].

[152] "Flask Web Server." [Online]. Available: http://flask.pocoo.org/. [Accessed: 15-Jul-

2018].

[153] “Paramiko.” [Online]. Available: http://www.paramiko.org/. [Accessed: 15-Jul-2018].

[154] Y. Ashibani, D. Kauling, and Q. H. Mahmoud, “A Context-Aware Authentication Framework for Smart Homes,” in *IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2017, pp. 1–5.

[155] 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.

[156] Y. Ashibani and Q. H. Mahmoud, “An Efficient and Secure Scheme for Smart Some Sommunication Using Identity-Based Signcryption,” in *36th IIEEE nternational Performance Computing and Communications Conference, IPCCC*, 2017, pp. 1–7.

[157] B. S. Adiga, P. Balamuralidhar, M. a. Rajan, R. Shastry, and V. L. Shivraj, “An Identity Based Encryption Using Elliptic Curve Cryptography for Secure M2M Communication,” in *First International Conference on Security of Internet of Things*, 2012, pp. 68–74.

[158] H. K.-H. So, S. H. M. Kwok, E. Y. Lam, and K.-S. Lui, “Zero-Configuration Identity-Based Signcryption Scheme for Smart Grid,” in *IEEE First International Conference on Smart Grid Communications*, 2010, pp. 321–326.

[159] T. W. Chim, S. M. Yiu, and L. C. K. Hui, “VSPN : VANET-Based Secure and Privacy-Preserving Navigation,” *IEEE Trans. Comput.*, vol. 63, no. 2, pp. 510–524, 2014.

[160] Y. Sun, R. Lu, X. Lin, X. Shen, and J. Su, “An Efficient Pseudonymous Generation Scheme With Privacy Preservation for Vehicular Communication,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 7, pp. 3589–3603, 2014.

[161] V. García, J. S. Sánchez, and R. A. Mollineda, “Knowledge-Based Systems On the Effectiveness of Preprocessing Methods When Dealing with Different Levels of Class Imbalance,” *Knowledge-Based Syst. Elsevier*, vol. 25, pp. 13–21, 2012.

[162] Y. SUN, A. K. C. WONG, and M. S. KAMEL, “Classification of Imbalanced Data: A Review,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 04, pp. 687–719, 2009.

[163] “Android App-Usage Data.” [Online]. Available:

http://sei.pku.edu.cn/~liuxzh/appdata/. [Accessed: 15-Jul-2018].

[164] 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, 2016.

[165] A. Rahmati *et al.*, "Seamless TCP Migration on Smartphones Without Network Support," *IEEE Trans. Mob. Comput.*, vol. 13, no. 3, pp. 678–692, 2014.

[166] M. F. Amasyali and O. K. Ersoy, "Classifier Ensembles with the Extended Space Forest," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 549–562, 2014.

[167] M. Segal, "Decision Tree and SVM-Based Data Analytics for Theft Detection in Smart Grid," *IEEE Trans. Ind. Informatics*, vol. 12, no. 3, pp. 1005–1016, 2016.

[168] K. S. Kim, H. H. Choi, C. S. Moon, and C. W. Mun, "Comparison of K-Nearest Neighbor, Quadratic Discriminant and Linear Discriminant Analysis in Classification of Electromyogram Signals Based on the Wrist-Motion Directions," *Curr. Appl. Phys.*, vol. 11, no. 3, pp. 740–745, 2011.

[169] J. Hensman, A. Matthews, and Z. Ghahramani, "Scalable Variational Gaussian Process Classification," in *18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

[170] Tiago A. Almeida and A. Yamakami, "Compression-Based Spam Filter," *Secur. Commun. Networks*, vol. 9, no. 4, pp. 1327–335, 2016.

[171] T. Windeatt, "Accuracy/Diversity and Ensemble MLP Classifier Design," *IEEE Trans. Neural Networks*, vol. 17, no. 5, pp. 1194–1211, 2006.

[172] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

[173] M. Li *et al.*, "Coupled K-Nearest Centroid Classification for Non-iid Data," *Trans. Comput. Collect. Intell. XV*, pp. 89. – 100, 2014.

[174] A. Singh, S. Prakash.B, and K.Chandrasekaran, "A Comparison of Linear Discriminant Analysis and Ridge Classifier on Twitter Data," in *International Conference on Computing, Communication and Automation (ICCCA)*, 2016, pp. 133–138.

[175] T. A. Almeida and A. Yamakami, "Content-Based Spam Filtering," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1–7.

[176] C. Beleites, U. Neugebauer, T. Bocklitz, C. Krafft, and J. Popp, “Sample Size Planning for Classification Models,” *Anal. Chim. Acta*, vol. 760, no. June 2012, pp. 25–33, 2013.

[177] L. Abdi and S. Hashemi, “To Combat Multi-Class Imbalanced Problems by Means of Over-Sampling Techniques,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 238–251, 2016.

[178] A. Massey and S. J. Miller, “Tests of Hypotheses Using Statistics,” *Math. Dep. Brown Univ. Provid. RI, 2912*, pp. 1–32, 2006.

[179] H. Cao and M. Lin, “Mining Smartphone Data for app Usage Prediction and Recommendations: A Survey,” *Pervasive Mob. Comput.*, vol. 37, pp. 1–22, 2017.

[180] 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, 2016.

[181] M. Rawat, N. Goyal, and S. Singh, “Advancement of Recommender System Based on Clickstream Data Using Gradient Boosting and Random Forest Classifiers,” in *8th International Conference on Computing, Communications and Networking Technologies, ICCCNT*, 2017, pp. 1–6.