A Context-Aware Method for Verifying User Identity in Pervasive Computing Environments

Mohammed M. Hassoun Al-Jawad

School of Computing, Science and Engineering

University of Salford

Manchester, UK

Submitted in Partial Fulfilment of the Requirements of

the Degree of Doctor of Philosophy, 2017

ABSTRACT

The necessity of verifying user identity is a crucial element of any system to avoid potential identity attacks. Selecting an appropriate verification method impacts on the system's overall behaviour since it is a trade-off between security and usability. It is even more significant when that system is situated in a pervasive environment since this type of environment is more vulnerable to such attacks. Any proposed method for this environment needs to be seamless (non-intrusive) and secure. As users in such environments tend to access a variety of resources across multiple networking domains, verifying their identity in a secure way requires a real-time verification method. Therefore, a seamless verification process with a reliable level of security is required.

Most existing methods of user identity verification are obtrusive, as they are not devised to work within a pervasive computing environment. This obtrusiveness is particularly germane when the main system uses more than one method in the verification process to enhance system security. Most existing solutions are either unaware of the context of the user, or context-aware but rely on part of the context. The context (current status) of a user can be determined through some primitives such as time and location, which are interpreted in a meaningful user context such as role or privilege.

This research proposes a new approach for user identity verification, called Context-Aware Identity Verification (CAIV) which uses multiple context parameters to increase the reliability of the verification process, yet does not rely on obtrusive methods such as biometrics like iris and facial recognition. It uses fuzzy logic reasoning to infer the identity of the user from knowledge

about the user's context. The rules of the fuzzy system were derived by extracting experts' opinions and casting that knowledge into a fuzzy inference engine. The inference engine makes the system capable of taking decisions in a similar way to that of experienced security personnel. The output of the inference engine is a trust value which reflects how much trust the system has in the claimed identity of the user. Thus, the system interprets the current context of the user into a trust value which eventually enables the system to determine the trustworthiness of the claimed identity.

Results obtained from extensive testing of the implemented system on the designated simulator show that the proposed approach as a primary method for user identity verification in pervasive computing environments maintains satisfactory rates in specificity, sensitivity and accuracy. It maintains two aspects: security and seamless access to secured resources in pervasive computing environments.

Moreover, the proposed approach guarantees that any compromised user credential information will not threaten the user's security and privacy in other domains. This kind of threat happens when a user's credentials are stolen by an intruder, which may give the intruder the ability to use them in other domains. In CAIV situation, these parameters are extracted from contextual information of the system environment; hence, the data breach affects only the CAIV domain without compromising other domains.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

BLE	Bluetooth Low Energy
CAIV	Context-Aware Identity Verification
CIA	Credential, Integrity and Availability
DBMS	Database Management System
ERD	Entity Relationship Diagram
FAR	False Acceptance Rate
FN	False Negative
FNR	False Negative Rate
FP	False Positivize
FPR	False Positive Rate
FRR	False Rejection Rate
H2H	Human to Human
H2M	Human to Machine
HAK	Authentication Methods (Something you Have, You Are and You Know)
IETF	Internet Engineering Task Force
IoT	Internet of Things

M2H	Machine to Human
M2M	Machine to Machine
MAC	Media Access Control
NFC	Near Field Communication
PC	Personal Computer
PIR Motion	Motion Passive Infrared Sensor
RFC	Request for Comments
RFID	Radio Frequency Identification
SAML	Security Assertion Mark-up Language
TN	True Negative
TP	True Positive
Ubicomp	Ubiquitous Computing
URI	Uniform Resource Identifier
W4H	Where, When, Who, What, and How questions
WiFi	Wireless Fidelity

ACKNOWLEDGEMENT

First and foremost, I would like to thank my supervisor, Dr. Adil Al-Yasiri, without whose help and support it would not be possible to complete this thesis. His advice and guidance helped me to achieve my research goals. During the research, I occasionally diverted off the right track and he always helped to point me back in the right direction.

My late Dad, my Mum, my wife, my daughter, my son, my friends, brothers and sisters gave me the power and the motivation to complete my research. My Dad's motivational words are still there in my memory, along with my Mum's wishes and prayers which bring me up whenever I tumble. The smiles of my wife and my kids which bring me joy. My brothers, sisters and friends encourage and support me in overcoming any difficulties that I have faced.

I thank my sponsor, the Iraqi government, which is represented by the Ministry of Higher Education in Iraq and the Iraqi Cultural Attaché in London. I appreciate their support and help during my journey.

Thanks to our Iraqi troop heroes who sacrificed their lives to defend our homeland. I am inspired by them to proceed and complete my research, regardless of any difficulties.

To my beloved country, Iraq, which is still recovering from the deep wounds caused by inhumane attacks faced over the last six decades.

Thank you all.

Chapter 1
Introduction

1.1 Introduction

The proliferation of smart things over the last decade has brought a mature smart environment to the fore. The smart environment (a pervasive computing environment) has different characteristics and challenges that make it more vulnerable than a traditional computing environment.

Context-awareness is one pervasive computing characteristic. It emerges as a key area of research in pervasive computing. Context-aware systems can adapt their operations to the current context of the user without explicit user intervention and thus can increase usability and effectiveness by taking environmental context into account (Baldauf, 2007). Context-aware solutions can be used in different aspects of our life, e.g. healthcare, industry, tourism, education and others. However, pervasive computing has both challenges and opportunities.

One of the challenges in the pervasive computing environment is security and privacy (Conti *et al.*, 2012). Generally, security is represented by confidentiality, integrity and availability, often called a CIA-triad. The CIA-triad is important in every secure activity within this environment. Confidentiality is an essential CIA-triad goal which need to be preserved in the pervasive environment. Consequently, that goal is embodied by verifying the user's identity. There are three

main authentication methods to verify user's identity. In this thesis, HAK is used as an abbreviation of those three most common authentication methods, namely, something you Have, something you Are, and something you Know. Verification achieved by traditional HAK methods have many drawbacks in term of usability, latency, complexity, security and privacy; therefore, there is a need to devise a new method which is tailored and designed to the pervasive environment, which is one of the motivations behind our proposal.

1.2 Research Problem

Some researchers have employed pervasive computing features to solve security issues, while others have highlighted the challenges and opportunities in pervasive computing environments. First, Al-Karkhi, Al-Yasiri and Linge (2015) developed an approach (known as NIAS) to infer the user's identity by monitoring their behaviour while interacting with the environment. However, the NIAS approach assumes that a user performs recurrent activities on a daily basis. This assumption leads to a set of rules based on certain parameters (identity, time, location). Although NIAS can adapt and learn new rules, it takes a long time to learn new behaviour. Second, Emmanouilidis et al. (2013) argued that inferring users' preferences by monitoring their activity while interacting with the smart environment is usually a challenging undertaking. Third, Krumm (2009) stated that it is challenging to create security and privacy mechanisms that adequately take into consideration the technical as well as the usage challenges of ubiquitous computing systems. Finally, Chalmers (2011) claimed that, "By predicting the future we can take less frequent readings, perform less processing of data and, probably, more importantly, we can avoid communicating new readings so frequently."

Da Rocha & Endler (2012) and Chen & Kotz (2000) argued that location is the most dominant context in use while others are rarely used. “Only location-aware applications have been widely deployed, producing some commercial products” (Da Rocha and Endler, 2012). Covington et al. (2002) stated that “We can no longer assume that user “sessions” will persist for extended periods with the same authentication and authorization credentials”.

For the reasons above, a new unobtrusive and non-distractive real time verification method of user identity is required for such a dynamic and challenging environment. The new method has distinctive features to overcome the current challenges with existing approaches. It needs to maintain the available context to determine whether a particular activity is performed by a trusted user or an intruder. It will be argued that user context can be used to ascertain a level of trust of that user activity. Furthermore, Li, Martínez and Rubio (2017) described the benefit of maintaining the available context as the key to achieve context awareness.

Current verification methods are vulnerable. For instance, biometrics such as face and fingerprint recognition are regarded as a robust identity verification method but they are susceptible to a spoofing attack. Samples can be gathered from their data sources, for instance, by capturing a facial image for facial recognition or by lifting a fingerprint (Akhtar, Micheloni and Foresti, 2015).

1.3 Research Motivation

Identity verification in a pervasive computing environment needs to change in terms of security and usability. The ubiquitous computing environment contains implicit data about users (context). These data can be used to enhance real-time identity verification. Campbell et al. (2003) stated that “Often, traditional security is somewhat static and context insensitive” and that “Security services should make extensive use of context information available. Context data can provide valuable information for intrusion detection mechanisms”. Added to this, Schumacher (2012) stated that “in the context of security, the focus has moved from the question ***who you are*** towards ***how you are***”; he declared that it is **suspicious behaviour** which matters rather than the identity itself. We believe that behaviour can be inferred from the context. The aim of this research is to present a new method of verifying a user’s identity. Context-Aware Identity Verification (CAIV) approach, which employs context parameters of pervasive computing environment in a user’s identity verification process. The CAIV method is a context-based approach to infer a user’s identity. The inferred result is based on the available context; consequently, the system can verify the claimed identity of the user.

1.4 Research Aim and Objectives

This research aims to develop a new verification method of user identity which relies on multi-context parameters. The method guarantees a seamless verification process. Added to this, there are five specific objectives which are illustrated below:

i. To understand context parameters of pervasive computing environment and discover the association between them and the user identity.

ii. To propose a new method that relies on multi-context parameters to verify a user's identity.

iii. To speed up the verification process by reducing the lengthy initial learning process by relying on a proper reasoning method.

iv. To build a simulator mimicking real scenarios of the system to test and evaluate the performance of the CAIV method.

v. To build a prototype of the CAIV method and integrate it with an information system.

The proposed method will benefit the user by allowing them to focus on high-level activities without having to interrupt what they are doing every time they wish to access a secure facility within the domain. Furthermore, it improves the system's performance by reducing the initial learning process needed which is one of the limitations of previous approaches.

1.5 Research Questions

In the conducted research, two key questions are addressed which are illustrated below:

i. What are the relevant context parameters that we can use to verify a user's identity?

ii. What is the best reasoning method to predict a user's identity by using these parameters?

1.6 Research Methodology

Research has more than one definition and different types. Research interoperates with logic which is part of a scientific method. The research type can be selected according to the research environment.

Ostle & Mensing (1975) define research as "An inquiry into the nature of, the reasons for, and the consequences of any particular set of circumstances, whether these conditions are experimentally controlled or recorded just as they occur". Added to that, it implies those researchers who are interested in the repeatability of the results and their extension to more general and complicated scenarios, while Kothari (2004) defines research as a scientific investigation art which leads to adding an original contribution to existing knowledge. This investigation is going to be achieved by researchers. Kothari stated that the duty of researchers is to find the problem and formulate it to be susceptible to research. That problem (the research problem) needs to be solved in a systematic way which is called a research methodology.

Kothari (2004) classified research types into five pairs. Descriptive versus analytical, applied versus fundamental, qualitative versus quantitative, conceptual versus empirical.

"Scientific method attempts to achieve this ideal by experimentation, observation, logical arguments from accepted postulates and a combination of these three in varying proportions." (Ostle and Mensing, 1975). Logic plays a significant role in formulating propositions explicitly and accurately in the scientific method. The purpose of experimentation is to test the hypotheses and find if there is a new relationship among variables if existed (Kothari, 2004).

Hence, the research involves the testing of the precision of CAIV against different situations. Therefore, the adopted research methodology is based on empirical research and the scientific method. Empirical research relies on experience or observation; the research conclusions are based on data which are capable of being verified by observation or experiment.

The following steps describe the research methodology process according to the adopted methodology:

i. Identify the research problem by relying on the state of the art.

ii. Perform a literature review of previous studies.

iii. Review some related work to specify the most significant parameters of the context. Then select those parameters to be applied as an input to the system.

iv. Define research objectives.

v. Find an inference method that complies with the research objectives.

vi. Develop a framework based on the selected inference method in step v.

vii. Design and build the framework simulator and generate a dataset.

viii. Evaluate the developed system which is built according to the proposed framework.

Figure 1.1 shows the CAIV research process. It illustrates the steps required to define the problem, formulate the solution, design, test and interpret the results.

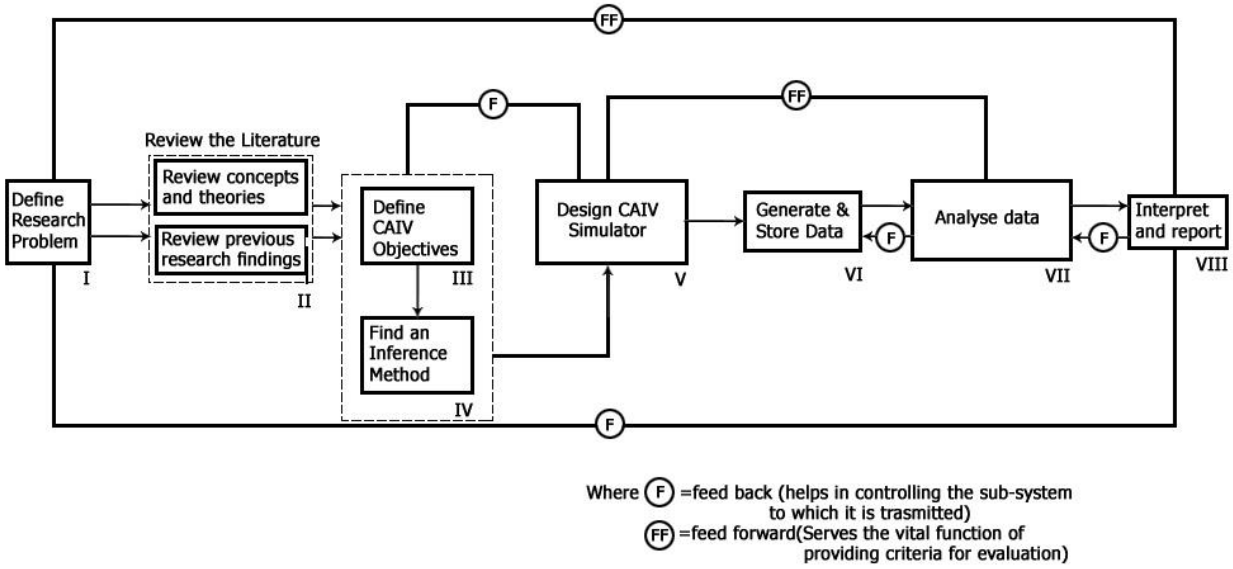


Figure 1.1: Research Process Flow Chart

1.7 Thesis Layout

The thesis is organised as follows.

- **Chapter One** gives an introductory description of my approach performance and goals as well as challenges and limitations. Furthermore, research objectives and questions are enumerated.
- **Chapter Two** explores the literature review and lists some related work. This includes identity, authentication factors, fuzzy logic and context-awareness. It discussed the reason behind using fuzzy logic and how this makes the solution more elastic.
- **Chapter Three** presents an analytical model of the system.
- **Chapter Four** gives an overview of the CAIV elements, architecture, scenarios and prediction method selection.
- **Chapter Five** reviews the dataset description and design, the simulator design and implementation and the CAIV prototype.
- **Chapter Six** presents the system's dataset generation, experiments implementation, threshold selection and a discussion of the result.
- **Chapter Seven** summarises the findings of the proposed method and suggests some possible directions for future research.

Chapter 2

Review of Digital Identity and Context-Awareness

2.1 Introduction

Identity is a significant element of the user's interaction in the digital world that allows the system to recognise the user, which helps the system to deal with every individual in the proper way. Identity is moved from a physical into a digital representation and is used everywhere in the digital environment to identify that user. Identity needs to be verified to confirm the authenticity of the user's identity. Identity is regarded as a crucial aspect in the digital environment. Any breach of identity (e.g. identity theft) affects both the user and the environment. The vulnerability to identity attack is increased depending on the computing environment type which can be traditional, cloud or pervasive computing. The pervasive computing environment brings new challenges to identity verification since it does not have a fixed location or a single domain. A traditional

computing environment device such as a desktop computer is situated in a fixed location and belongs to a certain domain while a pervasive computing device, for example, a tablet, may be used in different locations and across multiple domains. Moving the user within a smart environment makes the system request the user's identity verification every time the user wishes to access a secure facility.

The process of verifying a user's identity in a pervasive computing environment is a real-time identity verification. It requires an instant response to the user's request and at the same time protects the user from any identity attack. This means that any proposed solution encounters a trade-off between security and usability, which remains an open issue. Context-awareness is an approach that is used to facilitate the interaction between the user and the system. There is a potential benefit of using it in the identity verification process.

2.2 Pervasive Computing

The world has witnessed a revolutionary development in the computing realm. The technological transformation from traditional to pervasive computing, from a huge device (mainframe computer) to a very tiny device (smart dust), affects computing environment characteristics dramatically. Krumm (2009) classified pervasive computing as the third era of computing, characterised by small and portable devices (smartphones, PDAs, embedded computers) of which a person may own many. The second era (the PC era) was when one person owned a personal computer used by this individual only. The first era (the mainframe computer era) was represented by a single large computer, belonging to an organisation and shared by many individuals simultaneously. The third era has raised a few challenges and opportunities within its environment, when different variables

are surrounding the device and these variables change over time. For instance, devices move among various domains within a specific environment and need to interact with the system's environments in these domains. There are different definitions, synonyms, challenges and opportunities within a pervasive computing environment.

Pervasive computing is also described as ubiquitous computing (ubicomp), ambient intelligence, everywhere (Greenfield, 2010), physical computing, the internet of things (IoT) or haptic computing. Each term asserts a specific aspect; however, they share common aspects such as portability, connectivity, usability and adaptability.

Weiser (1991) defined pervasive computing as "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it". Moreover, he described pervasive computing as **invisible computing** and **calm technology**.

Reddy (2006) stated that for the promise of pervasive computing to materialise fully, technology and support structures need to advance along four directions: computing, communication, cognition and collaboration - the 4Cs of pervasive computing. It is not difficult to speculate how the world of tomorrow will be as we make steady progress on the 4Cs and the resulting advances in pervasive computing. The progress affects several aspects such as knowledge workers, intelligent transportation, pervasive healthcare and context-aware appliances and enterprises.

2.3 Related Work

The state of the art shows different approaches which have used context-awareness in various areas such as healthcare, daily activity, security, etc. These approaches use different parameters from the context to facilitate the interaction between the user and the systems in the pervasive environment and make that interaction seamless. Furthermore, there are several security approaches that also use context to enhance the identity verification process.

Jia et al. (2014) proposed to adjust the authentication configuration of the user according to the context, and they stated that it is a trade-off between security and convenience while the user is in different contexts.

Maintaining limited parameters is another issue in using context-awareness in security. Nishiki & Tanaka (2005) used two parameters from context: user location and profiles. Jang et al. (2010) used context in security by proposing context-awareness role-based access control (RBAC); however, they relied on the user's profile and environment, and they tended to infer the service rather than the identity. Other approaches have a limited scope, such as that used by Kim & Lee (2006) that used context-awareness in security but it was limited to the home domain.

There are two aspects related to the previous solutions, which are the interaction type and the environment. The CAIV method maintains Human to Machine(H2M) interactions while there are two approaches in which they used Machine to Machine (M2M) or (Human to Human) H2H interactions. Choi et al. (2013) used context-awareness in healthcare services but they rely on M2M interaction, and Malek et al. (2008) utilised a context-aware concept to make users customise their preferences and rules to authenticate other members (H2H interaction). Regarding computing

scope, the CAIV framework will be applied in ubiquitous computing, while there is an approach by Hayashi et al. (2013) in which they used context-awareness to make authentication scalable in a mobile computing environment.

2.4 Identity

There are different definitions of identity in the literature. Deng (2011) defined identity as "the way individuals and groups define themselves and are defined by others on the basis of race, ethnicity, religion, language, and culture" while Taylor (1989) has another definition: "my identity is defined by the commitments and identifications which provide the frame or horizon within which I can try to determine from case to case what is good, or valuable, or what ought to be done, or what I endorse or oppose". From the two definitions above, it can be concluded that identity can be a claim of owning a facility or it might be the responsibility's boundaries which embody the person's identity. Also, Rundle et al. (2007) defined identity as "a limited notion of a set of claims", which supports the Deng definition. However, Bishop (2002) stated that identity covers a wider range of subjects, not just people. Subjects of identities can be software agents (e.g. Web services and user client software) and hardware devices (e.g. PCs, mobile phones and network equipment).

Similarly, (Chisholm, 1997; Fearon, 1999) described identity as the sameness of an object anywhere and anytime. In addition, the Fearon context approach represents the current scenario in the digital environment, for instance, when a person has two different roles at the same university. On one hand, he or she is a student who can access their course class by using their student ID, yet on the other hand, s/he is a lecturer and can access the staff lobby by using their staff ID.

Rountree (2012) classified identity into two types: physical identity and digital identity. Rountree has a definition of identity in general: “your identity is the set of characteristics that make you who you are”. He believed that physical identity and digital identity are not similar; however, they are conceptually the same. The difference between them is that it is easy to hide the characteristics of digital identity from a potential attacker, which is the opposite of physical identity. Additionally, he stated that trust is a vital factor in identity verification.

“An identity of an individual person may comprise many partial identities of which each represents the person in a specific context or role. A partial identity is a subset of attribute values of a complete identity, where a complete identity is the union of all attribute values of all identities of this person” (Pfitzmann and Hansen, 2010). They argue that identity (combined identity) represents the combination of different attributes from specific context and it is not a single attribute. A combined identity represents the comprehensive identifier of the user. This identity is a real representation of the user. Added to that, the Identity itself can be an attribute inside other concepts.

For instance, Al-Karkhi, Al-Yasiri and Linge (2015) included the identity as an attribute of user activity as shown in Equation 2-1 below:

$$\boldsymbol{A = [id, L, t]} \quad \textbf{Equation 2-1}$$

2.4.1 Identity Management

Identity management is an administrative area which is used to maintain digital identity during the identity life cycle (creation, modification and revocation). It organises and controls a user's access to the system services and resources. Identity is defined and classified in the identity management context in a way that makes it different from other disciplines.

Windley (2005) illustrated an identity concept that called tiers of identity. Tier 1 (My Identity) has its attributes (traits) derived from the person himself. These attributes make them unique (physical and psychological traits). Tier 2 (Shared Identity) attributes are derived from the environment which is assigned to a person by a third party to identify them temporarily, e.g. a driver's licence, credit card, health insurance, etc. Tier 3 (Abstracted Identity) deals with the identity of groups and with profiling, which is used to track a person or expose them to direct advertising.

ITU-T (2009) has another classification of identity. They stated that an identity consists of three different types of data: identifier, credentials and attributes.

• **Identifiers** are a series of digits, characters and symbols or any other form of data used to identify a subject. Identifiers can be scoped by time or space or by both. For example, a URI is globally unique over time. Pseudonyms can be temporal and efficient only for a specific service. Some examples are user account names, passport numbers, mobile phone numbers, employee numbers, pseudonyms and URI.

• **Credentials** are a set of data providing evidence for claims about parts of or an entire identity. For example, passwords, digital certificates, fingerprints, Kerberos tickets (Neuman and Ts'o,

1994) and SAML assertions (Hughes and Maler, 2005). Establishing a claim can be based on one or more credentials.

• **Attributes** are a set of data that describe the characteristics of a subject. The data include fundamental information for identifying a subject (e.g. full name, domicile and date of birth), his/her preferences and information generated based on his/her activities. Some examples include family names, domiciles, ages, genders, roles, titles, affiliations, activity records and reputation.

2.5 Authentication Factors

The process of verifying a user's identity is called electronic authentication (e-authentication) or authentication. This process can be performed through different approaches. These approaches are called factors which are used to verify a user's identity. Presently, there are three authentication factors (HAK methods). However, they are named differently. Cristofaro (2014) named authentication factors as 1) knowledge, something the user knows, e.g. a password; 2) possession, something the user has, e.g. a security token (also called a hardware token); 3) inherence, something the user is, e.g. a biometric characteristic, while the State Service Commission of New Zealand defined these factors as something you know, something you have or something you are (Commission, 2006).

There are three authentication approaches to combine this three-factor identity verification. First, the user can use only one factor for authentication which is called single factor authentication, second, user identity can be verified by using two factors which is called two-factor authentication, and, finally, to enhance the security of authentication, more than two factors can be utilised which

is called multi-factor authentication. The next three sections illustrate the three authentication factors.

2.5.1 Knowledge (something you know)

Knowledge-based authentication relies on a shared secret between the system and the validating user (Chun-Li, Hung-Min and Hwang, 2001). This secret can be generated by the user or by the system. Knowledge-based authentication has four categories: password (phrase), passcode, pattern recognition and challenge question. The password is the most common category (Cranor and Garfinkel, 2008). A password is less costly and is easy to use and maintain. However, it is vulnerable to replay attacks, shoulder surfing, password theft, etc. One of the reasons passwords became the dominant method of authentication in early computer systems is because of the accessibility they afforded to users (Morris and Thompson, 1979). It is classified as a well known method, because it is more usable than other methods.

2.5.2 Possession (something you have)

Possession is a physical device (an object) possessed by the user. It helps the user to prove their identity without the burden of having to remember the secret key which is already stored on the user's device and is recognisable by the system. The physical device can be a bankcard, key fob or smart card (O'Gorman, 2003). It can be an active device by adding the facility to generate a one-time passcode (Weiss, 1988). A passcode is like a password except it is generally longer and more sophisticated and is generated or stored on a device.

2.5.3 Inherence (something you are)

Inherence authentication factors are unique physical traits of the user such as iris, fingerprint, voice or face recognition or vein patterns; this is called biometrics or biometry. Biometrics have two subcategories: traditional biometrics and behavioural biometrics. Traditionally, biometrics rely on physical traits while behavioural biometrics rely on a behaviour trait that is acquired over time versus physiological characteristics or a physical trait (Schumacher, 2012). It is possible to breach both of them. However, the available technology makes traditional biometrics more vulnerable to biometric spoofing.

Added to this, two other approaches have proposed a fourth authentication factor but they have not yet been approved. Brainard *et al.* (2006) claimed to add somebody you know as a fourth authentication factor and Choi and Zage (2012) claimed to add the location of the user, "where you are," as a new authentication factor.

2.6 Context-Awareness

Context-awareness is a technology which makes the actor (user, device) adapt, interact or link with the environment or with each other in an unobtrusive way without distracting the user during sophisticated operations. There are different definitions of the context and context-awareness in the literature.

2.6.1 Context and Context-Aware

Brown (1995) defines context as the elements of the user's environment about which the computer knows. Ryan et al. (1998) referred to context as the user's location, environment, identity and time.

Abowd et al. (1999) defined context as “any information that can be used to characterise 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 application themselves”. Dourish (2004) suggested that the notion of context in pervasive computing has a dual origin. First, it is a technical notion, figuring out the human action and the relationship between that action and the computational platform to support it. Second, it is a social science notion, making analytic considerations to certain aspects of social settings. These four definitions of context represent the evolution of context definition which is represented by a very general definition by Brown to a very specific one by Ryan. However, it can be seen as a comprehensive definition for both Abowd and Dourish who give a new definition which conveys a picture of the transformation from traditional to ubiquitous computing. There are particular types of context that are, in practice, more important than others. These are location, identity, time and activity. The importance of these parameters is due to their abilities to characterise the situation of a particular entity (user, device). The importance of context is of interest to many areas of software design in computer science, especially in the pervasive computing (ubiquitous computing) environment which is exacerbated by the new need to keep track of ubiquitous things in the pervasive environment. In this situation in which the environment is variable, there is a greater need to understand more about the environment. Hence, the most significant concern in pervasive computing research is to anticipate the relationship between computation and context (Dourish, 2004). Dourish (2004) reconsiders context as an interactional problem rather than a representational problem. He focuses on an alternative view of context and discusses some of its implications.

Context can be used to adapt the system's functions according to any change in the context, a facility called context-awareness. The first time context-awareness was discussed in 1995 by Schilit et al. as software that "adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time"(Schilit and York, 1995). Hong defined it as follows, "Context-aware systems can provide users with better service based on analysing physical context and personal context such as user schedule, user preferences and so on" (Hong, Suh and Kim, 2009). Context-awareness is used in this regard to make the adapted systems of the concept more usable and user-friendly. Context-awareness has also been described as being when "A system is context-aware if it uses context to provide relevant information or services to the user; or both of them, where relevancy depends on the user's task" (Abowd *et al.*, 1999).

Context-aware applications are becoming more prevalent and can exist in the areas of wearable computing, mobile computing, robotics, adaptive and intelligent user interfaces, augmented reality, adaptive computing, intelligent environments and context-sensitive interfaces (Krumm, 2009). Additionally, context-awareness has become somewhat synonymous with other terms such as adaptive (Brown, 1996), situated (Hull, Neaves and Bedford-Roberts, 1997) and environment directed (Fickas, Kortuem and Segall, 1997), context sensitive (Rekimoto, Ayatsuka and Hayashi, 1998).

According to previous definitions of context-awareness, Krumm (2009) classified context-aware computing into two categories: using context and adapting to context.

The first category has two definitions. The first represents the dynamicity of computing devices and how they are affected by sensed and detected context which interpret and respond to the user's

devices and the local environment (Hull et al., 1997; Pascoe, 1998; Ryan et al., 1998). The second definition uses context to automate a software system, to modify an interface that will maximise computing service flexibility (Dey, 1998; Salber et al., 1999).

The second category is "adapting to context". It includes context-aware applications which dynamically change or adapt their behaviour based on the context of the application and the user (Schilit, Theimer and Welch, 1993; Brown, Bovey and Chen, 1997; Ward, Jones and Hopper, 1997; Abowd *et al.*, 1998, 1999; Kortuem, Segall and Bauer, 1998; Salber *et al.*, 1999). Ryan et al. (1998) define them as applications that monitor input from environmental sensors and allow users to select from a range of physical and logical contexts according to their current interests or activities. This category is slightly more restrictive than the previous one by identifying the method in which applications act upon context. Brown (1998) defines context-aware applications as applications that automatically provide information and take action according to the user's present context as detected by sensors. He also adopts a narrow view of context-aware computing by stating that these actions can take the form of presenting information to the user, executing a program according to context or configuring a graphical layout according to context.

Jaroucheh et al. (2010) represented context knowledge as the context primitive which represents the base context constructs: entity classes, entity attributes, entities associations and rules, as shown below in Figure 2.1.

- Entity class: represents a group of entities (e.g. users, places, devices, etc.) sharing some properties.

- Attribute class: represents entities' attributes e.g. position, temperature, etc.

- Association class: represents a relationship between one entity and either another entity or an attribute.

- Rule class: two types of rule can be identified: (i) **Consistency rules** provide a mechanism for context consistency by specifying conditions that must be held in the context information. For example, consistency rules could specify that if the person is cooking, s/he must be in the kitchen. (ii) **Inference rules** are used to generate new context information after reasoning on an existing one. For example, an inference rule could conclude that a person is sleeping if the light is off and it is night time.

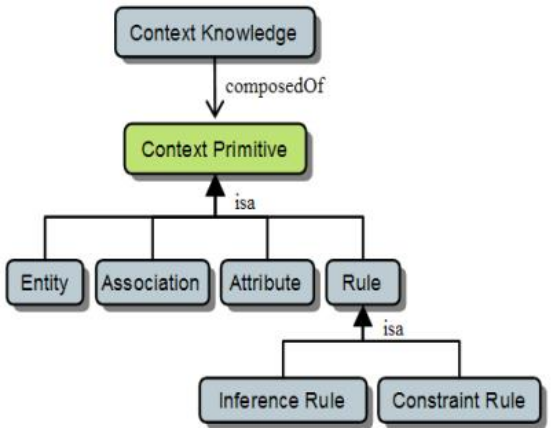


Figure 2.1: Context Primitives (Jaroucheh, Liu and Smith, 2010)

We conclude that the location and time are very significant elements in both context and identity. There are different context parameters they existed in the literature which are relying on location and time or one of them, such as user schedule, user preferences, nearby object (user), and user profile. These four parameters are going to be considered in the CAIV modelling and design. These parameters are linked to the user task (activity) to find how much the system trust that user of a specific task (activity).

2.6.2 Context-Aware Taxonomy

There are different taxonomies for context-awareness. These are three core taxonomies: first, a taxonomy presented by Schilit & York (1995) which has two orthogonal dimensions: 1- The task obtains information or executes a command, 2- The task is executed manually or automatically. This taxonomy brings four terms: first, proximate selection applications; information retrieval is manual based by the user according to the available context to list-related objects or places which are linked to the user to facilitate the user's choice (preferences). Second, automatic contextual configuration, where information retrieval is automated based on available context without user intervention; the system creates an automatic binding according to context. Third, contextual command applications that execute commands for the user manually based on the context. Finally, context-trigger actions, an application that executes commands for the user according to the context simply based on if-then rules; if the combination of context exists, the service executes automatically.

The second taxonomy, proposed by Pascoe (1998), deals with context-aware features. The first feature is contextual sensing and is the ability to detect contextual information and present it to the user, augmenting the user's sensory system. The next feature is contextual adaptation and is the capacity to execute or modify a service automatically based on the current context. The third feature, contextual resource discovery, allows context-aware applications to locate and exploit resources and services that are relevant to the user's context; this is mapped directly to automatic contextual reconfiguration. The final feature, contextual augmentation, is the ability to associate

digital data with the user's context. A user can view the data when they are in that associated context.

The third taxonomy is presented by Abowd et al. (1999), whereby they combine the ideas of the two approaches above and take into account the three major differences. Similar to Pascoe's taxonomy, it is a list of context-aware features that context-aware applications may support. There are three categories:

1. Presentation of information and services to a user

2. Automatic execution of a service

3. Tagging of context to information for later retrieval

Abowd et al. (1999) claimed that their taxonomy has two benefits; the first is that it further specifies the types of application for which the researchers provide support. The second benefit is that it describes the types of feature that developers should be thinking about when building context-aware applications.

The three taxonomies show different views of how to use information from context and present systematic methods. They aggregate data, interpret them and take action or store these data to be retrieved later.

2.6.3 Context Acquisition (Sensing)

Chen (2004) presents three different approaches of how to acquire contextual information**:**

(1) Direct sensor access. The client software gathers the sensed information from a local built-in device.

(2) Middleware infrastructure. It uses encapsulation methods by introducing a layered architecture to context-aware systems with the intention of hiding low-level sensing details.

(3) Context server. The next logical step is to permit multiple clients to access remote data sources.

Indulska & Sutton (2003) classified sensors into three groups:

• Physical sensors: the most frequently used types of sensor are physical sensors. Many hardware sensors are available today which are capable of capturing almost any physical data. Table 2-1 shows some examples of physical sensors (Schmidt and Van Laerhoven, 2001).

• Virtual sensors: these source context data from software applications or services. For example, physical location can be sensed by an electronic calendar, emails, etc. Other context attributes can be user activity such as mouse movement and keyboard input.

• Logical sensors: these sensors make use of several information sources and combine physical and virtual sensors with additional information from databases or various other sources to solve higher tasks.

Table 2-1: Commonly Used Physical Sensor Types

Type of Context	**Available sensors**
Light	Photodiodes, colour sensors, IR and UV-sensors etc.
Visual context	Various cameras
Audio	Microphones
Motion, acceleration	Mercury switches, angular sensors, accelerometers, motion detectors, magnetic fields
Location	Outdoor: Global Positioning System (GPS), Global System for Mobile Communications (GSM); Indoor: Active Badge system. etc.
Touch	Touch sensors implemented in mobile devices
Temperature	Thermometers
Physical attributes	Biosensors to measure skin resistance, blood pressure

Chen & Kotz (2000) describe four sensing mechanisms: location, low-level context beyond location, high levels of context and context changes.

2.6.4 Context Interpreting

Context-aware systems take data as input and then determine how to adapt or respond to these data. Context inferencing is the act of making sense of these input data from sensors and other sources, to determine or infer the user's situation. Once the user's situation has been inferred, the application can take an appropriate action. The sensed input is often not enough to infer the situation appropriately so this brings up additional issues such as how to resolve ambiguity or uncertainty in context, and the role of rules and machine learning (Krumm, 2009).

Krumm makes a comparison between rules and machine learning which can be used to overbalance of rules. Context-aware applications are most commonly designed from a set of if-then rules, such as:

IF the application senses a particular situation, **THEN** it should perform a particular action.

Rules are easy to create because all the knowledge for each rule is represented in a homogeneous format and rule-based systems are relatively easy to build because there is a large number of existing rules engines that determine when a rule has been satisfied. Rules are also relatively intuitive and are thus easy to work with (Krumm, 2009).

Additionally, there are some disadvantages of applying machine learning. As it is applied to learn the probabilistic relationships between the situation and adaptions, that relationship may be difficult to learn, it may require a significant amount of data to learn, it is hard to debug and may not be intuitive to the application developer or end user.

Sometimes it might be necessary to have access to historical context data. Such context histories may be used to establish trends and predict future context values. As most data sources constantly provide context data, the maintenance of a context history is mainly a memory concern so a centralised high-resource storage component is needed. Since in a server-based architecture the context data provided by sensors has to be stored at the server-side to offer it to customers, the majority of these systems have the facility to query historical context data. Managing historical

context data provides the ability to implement intelligent learning algorithms which allow provision of highly flexible context-aware services. Furthermore, based on learning algorithms, contextual information can be predicted to provide a certain set of services to the user proactively. Many of the systems store contextual information, but they all use learning techniques to provide context-aware service proactively (Baldauf, 2007).

2.7 Summary

Reviewing the literature on both identity and context-awareness gives us a clear vision of the challenges and opportunities of these two correlated fields. Context-awareness relies on context and is deployed mainly in pervasive computing. User identity verification is one of the challenges in pervasive computing environment because of the environment variances. The pervasive environment is different from a traditional computing environment since the user (object, user) is moving around and does not have a fixed location. In a traditional computing environment, the PC has a fixed location for a month or a year(s), while a smart object can move between locations during an hour or a day. The available verification methods of user identity were devised for the traditional computing environment and reused in pervasive computing. Using them in the pervasive computing environment systems make the system vulnerable to intruder attacks such as spoofing.

Context-awareness has different taxonomies. However, they share a common goal. The unique goal is how to make use of the available context and provide the user with a convenient service without user intervention. It aggregates data about the user from physical or logical sensors and interprets them to trigger an action within the system.

Context-awareness was used in the early 1990's for tracking users, as a tourism guide and other solutions. It tends to bring a seamless service to the user and made the system less intrusive. It was used in a variety of fields such as health, environment, daily activity and security. Security approaches have some issues related to context limitations and slowness of the learning process due to use of some methods which need sometimes to adapt, such as machine learning.

Reviewing the literature highlighted the significance of location and time for both identity and context-aware parameters. Four parameters have been chosen from the context to be considered in the CAIV modelling and design. The next chapter illustrates the details of the CAIV method analytical model.

Chapter 3

CAIV Analytical Model

3.1 Introduction

The CAIV method aims to verify a user's identity by measuring the trust level of the user's activity and take a decision upon the current context. It also makes the proposed method more integrative with existing systems in the pervasive computing environment and tends to be less intrusive to the user. The trust value, which is inferred from the aggregated context of the user, affects the CAIV decision. That decision either accepts the claimed identity to identify the user as a legitimate user or rejects that claim. Trust has a certain range of values. Modern trust models define trust in terms of both the probability and the certainty of a good outcome (Jsang & Ismail, 2002; Paradesi et al., 2009; Huynh et al., 2006). Since the probability values are between zero and one, the value of the trust is between [0,1]. CAIV finds the probability of an activity occurring. Each activity occurs at a certain time and location, which is called an event.

Events play a major role in everyday life, organising and depicting the way that the user interacts with a certain environment. Laliwala & Chaudhary (2008) refer to the event as an entity that relies on two components, which are time and location; Craig and Whitty(2017 defined the event as a record of location and time For instance, the calendar events in the user diary have two main significant components which are time and location, as shown below in Figure **3**.**1**.

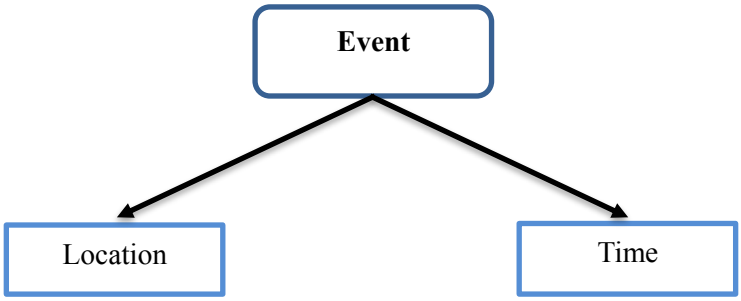


Figure 3.1: Event Components

We believe that each component of the event represents a part of the truth; confirming one or both of these components by context parameter can affects the truth value positively. The obtained value of truth for each context parameter is interpreted as the probability of attending the user at a certain (current) location and time. The union of truth values of context parameters is the trust value. The trust value represents how far the system is confident that the claimed identity is considered as a genuine user identity. The next sections illustrate the trust value extraction and show how this value is aggregated, mapped and banded together from each parameter of CAIV model.

3.2 Context Parameters

The context-aware identity verification (CAIV) method defines four context parameters about a particular user which is based on the literature review, known as the user profile. CAIV uses the four context parameters, which are *user preferences, ambient objects, history log, and calendar events* in the verification process. The parameters rely on event component(s), as shown in Figure 3.2, the calendar relies on location and time, while the others rely on the location only. More discussion of each of these parameters is provided in the next sections. This profile is then used to infer (verify) the user identity from the user's context, which can represent user behaviour. A user behaviour is categorised by a set of activities performed by the user. The verification process is based on the process of mapping users' behaviour to their profile to discover the possibility of finding a user at a certain event which is interpreted as a trust value.

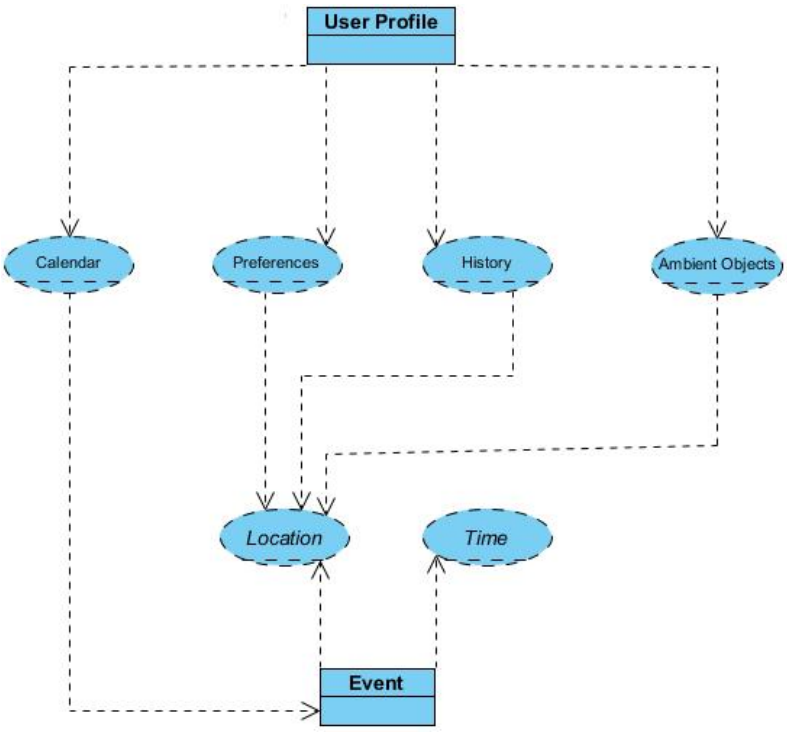


Figure 3.2: User Profile Dependency

A user profile defined in terms of the context parameters as described in (3-1) is:

$$\boldsymbol{Uc = [Cal, Amb, Pref, Hist]} \qquad \textbf{3-1}$$

Where: Uc – *the user profile (context)*

Cal – *user calendar events*

Amb – *ambient objects assigned to the user*

$Pref$ – *user's preferences*

$Hist$ – *user's history log*

U: set of registered users.

L: set of locations.

Each of the profile elements defined in terms of the location (L), and time (t) where the user behaviour (activities) performed. Therefore, we define a function that represents each of these parameters as follows:

$$Cal = f_c(L, t)$$

$$Amb = f_a(L, t)$$

$$Pref = f_p(L, t)$$

$$Hist = f_h(L, t)$$

Hence (3-1) can be re-written as:

$$\boldsymbol{Uc = f\ (L, t)} \qquad \textbf{3-2}$$

where f is an aggregation of f_p, f_a, f_h and f_c .

Also, a user behaviour is defined in terms of the activities (A) performed within the pervasive environment. An activity defined as a tuple (vector) of user identity, resource location and time of accessing the resource as described in Equation 2-1:

$$\boldsymbol{A = [id, L, t]}$$

In order to use the user activity to infer the user's identity, it mapped to the user profile, such that:

$$\boldsymbol{id = RM(A, Uc)} \qquad \textbf{3-3}$$

3-3 asserts that the user's identity is a measure of mapping a user's activity to the user's profile. We call this as the system's *Trust* in the user. Therefore 3-3 can be restated as:

$$\boldsymbol{Trust = RM(A, Uc)} \qquad \textbf{3-4}$$

3-4 represents the basis of CAIV's verification process. In the next sections, we explain how each of the functions f_c, f_a, f_p and f_h represents the trust value of the corresponding parameter of Uc.

3.2.1 Mapping an Activity to User Preferences

User Preferences are those locations that the user tends to visit frequently. Each preferred location is stored with a description, location and the probability of attending a user in that location. Section 4.3.1 provides further details about this parameter. The mathematical representation of this parameter is as follows:

Given a user j with an identity id_j, there is a list of preferences Pr_j such that:

$Pr_j = \{pr_{\mathrm{jv}} : v = 1, \dots, q_j\}$ where q_j is the number of preferences for user j

Each user preference is represented as a tuple (vector) as follows:

$pr_{\mathrm{jv}} = [descr, id_j, l_v, wp_v]$

where $descr$ - textual description of pr_{jv}

l_v - the location where pr_{jv} is associated with that preference

wp_v - the user assigned weight to pr_{jv}, $wp_v \in [0,1]$

As an example, a user id1 registers one preference as swimming with weight 0.5; then this preference is stored in the system as:

$pr_{18} = ['Swimming', id_1, loc\_8, 0.5]$

Now as user j performs an activity $(a, e.g. swipe\ the\ card)$ as in instance of A,

see Equation 2-1, Al-Karkhi, Al-Yasiri and Linge (2015) defined activity as:

$a = [id_j, l_a, t_a]$

The system maps activity (a) to each preference in Pr_{j} such that a list of weight values (WP) is produced, as follows:

$$\boldsymbol{WP} = \begin{cases} \forall\, \boldsymbol{pr_{jv}} \in \boldsymbol{Pr_{\mathrm{j}}}, & \boldsymbol{wp_{jv}}, \quad \textbf{if } \boldsymbol{l_a = l_v} \\ & \mathbf{0}, \quad \boldsymbol{otherwise} \end{cases} \qquad \textbf{3-5}$$

Now let f_{pr} be a function representing the contribution of the preferences context to the trust such that:

$$\boldsymbol{f_{pr} = \max(WP)} \qquad \textbf{3-6}$$

3-6 states that the system maps the activity (a) to the preference associated with location l_a with the highest weight value. The weighting values are determined by a trust policy which determines the possible values to be given for each wp_v. For instance, during registration, ranked questions can be used with rating scale (seldom, occasional, and frequent, which represent the values (0.25, 0.5, 0.75) respectively as determined by the administrator, for each registered preference.

3.2.2 Mapping an Activity to User's Ambient Objects

Pervasive computing has smart objects equipped with a communication technology such as Bluetooth, Wi-Fi, RFID and others. Identifying these objects and link them to the user make the system more confident about the user identity. Each object has an identifier (physical, MAC address, SN), description, location. The object can be owned by a user, group of users, or a location. Section 4.3.3 provides more details about ambient object. The mathematical representation of an ambient object is as follows:

Given a user j with an identity id_j, there is a list of ambient objects AO_j such that:

$AO_j = \{ao_{\text{jk}} : k = 1, \ldots, o_j\}$ where o_j is the number of ambient objects for user j

Each ambient object is represented as a tuple (vector) as follows:

$ao_{\text{jk}} = [identifier, descr, ownership, id_j, l_k]$

where

$identifier$ - is the object identifier if it exists

$descr$ - textual description of ao_{jk}

$ownership$ - determines the ownership of the device (whether owned by one user, more than one user, or a location)

l_k - the location where ao_{jk} is associated

$$wa_k = f(ownership) = \begin{cases} wm & if\ ownership = 'multiple\ user' \\ wl & if\ \ ownershiship = 'location' \\ wu & if\ \ ownership = 'user' \end{cases}$$

Where

wa_k - $Ambient\ object\ weight,\ wa_k \in [0,1]$

Now as user j performs an activity (a) as in instance of A, see Equation 2-1:

$a = [id_j, l_a, t_a]$

The system maps activity (a) to each ambient object in AO_{j} such that a list of weight values (WA) is produced, as follows:

$$\boldsymbol{WA} = \begin{cases} \forall\ \boldsymbol{ao_{jk}} \in \boldsymbol{AO_{\text{j}}}, & \boldsymbol{wa_k}, \quad \textbf{if}\ \boldsymbol{l_a = l_k\ or\ id_a = id_k} \\ & \mathbf{0}, \qquad\qquad \boldsymbol{otherwise} \end{cases}$$

where $\boldsymbol{l_a}$the location of activity$(\boldsymbol{a})$ **3-7**

Now let $f_a$be a function representing the contribution of the ambient object context to the trust such that:

$$\boldsymbol{f_a} = \mathbf{max}(\boldsymbol{WA}) \qquad \textbf{3-8}$$

3-8 states that the system maps the activity (a) to the ambient object associated with location l_a with the highest weight value. The weighting values are determined by trust criteria which are based on experts' opinion to determine the possible values to be given to each wa_k based on ownership of that object. The experts proposed to assign 0.2 for the object that belong to group a of people (wm), 0.5 for the object whom belong to a location (wl), and 0.9 for the object whom

belong to a person (wu). Alternatively, the system administrator can assign the proper value based on ambient objects ownership.

Three examples are covering the potential scenarios. Firstly, user id_2registers one of his devices and that device is owned by id_2, the function wa_k calculates the ambient object weigh.

$$ao_2 = ['18-67-80-5D-3F-4B', 'Macbook', 'one\ user', id_2, null]$$

Secondly, user id_3 and user id_4 *are both* registered one of their device and, that device is owned by $id_3\ and\ id_4$, the function wa_k calculates the ambient object weigh.

$$ao_3 = ['23-A7-D0-5A-4B-42', 'Laptop', 'more\ than\ one\ user', id_3, null]$$

$$ao_3 = ['23-A7-D0-5A-4B-42', 'Laptop', more\ than\ one\ user, id_4, null]$$

Finally, ao_4 is an object belongs to main office (loc_9), a registered user was holding that device and heading to the main office door.

$$ao_{49} = ['iPad', 'location', id_4, loc\_9]$$

3.2.3 Mapping an Activity to User History

User's history implies the user's behaviour which regards as a clue to find the potential location of the user at certain time and location. Mining this information can be individually or as a group of users. CAIV approach is to find the individual behaviour of each user which make it more relevant to the user's attitude. We need to find the probability of the user to access a new location (visited location) when he moved from a current location (pre-visited location) based on history logs.

The history probability relies on Markov chain. Markov chain is stochastic a process named after Andrey Markov a Russian mathematician. It is called memoryless since it is categorised as a random process. In a Markov chain, the probability of an event at any point in time is a function of the likelihood with which events occurred at previous time periods (Luger, 2005). Time-homogeneous Markov chain with one-step transition probability has been used in CAIV modelling since we don't need more than one step transitions between the environment locations, (**3-9**) showing the Markov Chain formula.

$$\mathbf{P} = \begin{bmatrix} \boldsymbol{p_{11}} & \boldsymbol{p_{12}} & \boldsymbol{p_{13}} & \cdots & \boldsymbol{p_{1g}} \\ & \vdots & & \ddots & \vdots \\ \boldsymbol{p_{i1}} & \boldsymbol{p_{i2}} & \boldsymbol{p_{i3}} & \cdots & \boldsymbol{p_{ig}} \end{bmatrix} \quad \text{3-9 (Behrends, 2000)}$$

Where:

- i, g is the number of locations in the pervasive computing environment

- $0 \leq p_{ig} \leq 1$

- $i, g \in L$

- $\sum_{g=1}^{n} p_{ig} = 1$

$\forall\, id_j\ \exists!\ P_j$

Given a user j with an identity id_j, there is a list of potential locations (based on the environment's available locations) which are possible to be attended by users; the history logs of the users stored in the history log (H_j) such that:

$H_j = \{h_{jn} : n = 1, \ldots, hs\}$, $hs\ is\ the\ number\ of\ history\ logs\ of\ user_j$

Each user history log is represented as a tuple (vector) as follows:

$h_{\text{jn}} = [id_j, l_n, stat_n, pl_n, t_n]$ where

id_j - the identifier of user j

l_n - the location where the activity is performed

$stat_n$ – the history log status, success =1, or failed =0

pl_n - pre-visited location of l_n

t_n - is the time of the activity

Now as user j performs an activity $(a, e.g. swipe\ the\ card)$ as in instance of A,

see Equation 2-1:

$a = [id_j, l_a, t_a]$

To find the probability (History Weight (WH)) of attending the activity location,

$wh_a = \text{P}(pl_a, l_a)_j, \quad pl_a, l_a \in L,$

wh_a - the weight of user j to be at the activity location when he moved from the previous location pl_a which stored in h_{jn}

, $wh_a \in [0,1]$,

$l_a \exists! pl_a$ - pl_a is the pre-visited location of activity location l_a

The system maps activity (a) to each history in H_j such that a list of weight values (WH) is produced, as follows:

$$\boldsymbol{f_h = WH = \begin{cases} \forall\, \boldsymbol{h} \in \boldsymbol{H_j}, & \boldsymbol{wh_a}, \quad \textbf{if } \boldsymbol{l_a = l_n, t_n = t_a, stat_n = 1 \; and \; pl_a = pl_n} \\ & \boldsymbol{0}, \qquad\qquad \boldsymbol{otherwise} \end{cases}}$$

3-10

As an example, a user $id_1$with pre-visited location (loc_4) is heading to the destination (loc_7), the system was able to calculate the history weight of that activity which is 0.6. The history weight can find it by the system, based on Markov Chain method which manipulated the history logs of the past month; the history record had been stored as follows:

$h_{17} = [id_1, loc_7, 1, loc_4, '2016 - 12 - 12\ 9{:}00']$

Now let $f_h$be a function representing the contribution of the history context to the trust such that:

3-10 showed that the system maps the activity (a) to the history log associated with location l_a. The weighting value is determined by Markov Chain algorithm for each associated pair of (pre-visited and destination locations) and determine how much is that weight wh_k if existed based on the number of visits to that location on a period of time. For instance, the user 1 had three history logs from location 4 to three different locations; which are$[id_1, loc_2, loc_4]$ the calculated p_{47} is 0.3, $[id_1, loc\_7, loc\_4]$ the calculated p_{46} is 0.6, and $[\, id_1, loc\_6, loc\_4]$ the calculated p_{42} is 0.1. That is based on the proportional visits for one location over the total visits (10) done by user 1. The credits of these visits are three times from loc_4 to loc_2, six times from loc_4 to loc_7, and

once from loc_4 to loc_6 over the past month. Based on the number of visits for location 2,7,6; the weight(probability) has been calculated for each potential location.

3.2.4 Mapping an Activity to User Calendar

Calendar event organises and stores information about location(s) that the user tends to access in the pervasive computing environment. The iCalendar format is used to help the CAIV model to understand the priority of each event. According to iCalendar, the event has three different types, which are VEVENT, VTODO, and VJOURNAL. Section 4.3.2 illustrates further details about iCalendar. The mathematical representation of Calendar event as follows:

Given a user j with an identity id_j, there is a list of calendar events C_j such that:

$C_j = \{c_{\text{js}} : s = 1, \dots, ev_j\}$, ev_j is the number of calendar events for user j

Each calendar event is represented as a tuple (vector) as follows:

$c_{\text{js}} = [descr, type, t_s, id_j, l_s]$

where

$descr$ - textual description of c_{jk}

type - the event type; it has three kinds (vjounal, vtodo, and vevent)

, they are ranked low, medium, and high respectively

t_s - the event time

l_s - the location where c_{jk} is associated

$$\boldsymbol{we_s = f(type)} = \begin{cases} \boldsymbol{wj} & \boldsymbol{if\ type = \text{vjounal}} \\ \boldsymbol{wt} & \boldsymbol{if\ \ type = \text{vtodo}} \\ \boldsymbol{we} & \boldsymbol{if\ \ type = \text{vevent}} \end{cases} \quad \textbf{3-11}$$

Now as user j performs an activity $(a, e.g. swipe\ the\ card)$ as instance of A, see Equation 2-1:

$a = [id_j, l_a, t_a]$

The system maps activity (a) to each calendar in C_{j} such that a list of weight values (WE) is produced, as follows:

$$\boldsymbol{WE} = \begin{cases} \forall\ \boldsymbol{c\ \in C_{\text{j}},} & \boldsymbol{we_s,} \quad \textbf{if}\ \boldsymbol{l_a = l_s\ and\ |t_a - t_s| \leq tt} \\ & \boldsymbol{0,} \qquad\qquad \boldsymbol{otherwise} \end{cases} \quad \textbf{3-12}$$

Where:

we_s -is a derivative weight based on calendar type of c_{js}, $we_s \in [0,1]$

tt - is the time tolerance in minutes

Now let f_c be a function representing the contribution of the calendar context to the trust such that:

$$\boldsymbol{f_c = \max(WE)} \quad \textbf{3-13}$$

3-13 states that the system maps the activity (a) to the calendar associated with location l_a, time t_a with the highest weight value. The weighting values (we_s) is derived from the calendar type. Therefore, the values of we_s for the event with vjouranl, vtodo, and vevent are 0.2, 0.5, and 0.9 respectively. These values have been assigned based on expert opinion which they are ranking

calendar types with low, medium and high respectively. The importance of calendar type has been decided based on their knowledge and the description of those three types.

Since the event type has three categories, there are three different use cases of calendar tuples.

Use Case 1

Firstly, a registered user with id1 has added a calendar event (IoT seminar) as a vjournal type. Accordingly, the calendar tuple of that event is stored in the system as:

$$c_{110} = [\text{'}IoT\ \ seminar\text{'}, \text{'}vjournal\text{'}, \text{'}2016-12-12\ 11{:}00\text{'}, id_1, loc\_10]$$

Use Case 2

Secondly, a registered user with id2 has added a calendar event (Fog Computing Workshop) as a vtodo type. Accordingly, the calendar tuple of that event stored in the system as:

$$c_{222} = [\text{'}Fog\ computing\ workshop\text{'}, \text{'}vtodo\text{'}, \text{'}2016-12-12\ 02{:}00\text{'}, id_2, loc\_22]$$

Use Case 3

Finally, a registered user with id3 has added a calendar event (Advance Programming II) as a vevent type. Accordingly, the calendar tuple of that event stored in the system as:

$$c_{313} = [\text{'}Advance\ programming\ II\text{'}, \text{'}vevent\text{'}, \text{'}2016-12-12\ 10{:}00\text{'}, id_2, loc\_13]$$

3.3 Conclusion

To sum up the CAIV approach; it is required to come up with a result or conclusion of each profile. That profile represents the available context at a certain time and location. As illustrated above in (3-4) and the way in which it maps these parameters to infer the trust value, therefore, there is a need to use an inferring technique to find the trust value. A reasong method need to be used to infer the trust value by using a Multiple Input Single Output (MISO) approach. The four inputs are interpreted into a single output. That output represents the trust value which consequently reflects how much the system trusts that user in a certain event.

Therefore, the target formula for user j is as illustrated in 3-14 below:

$$\boldsymbol{Trust_j = RM(f_{cj}, f_{hj}, f_{aj}, f_{pj})}, \textit{ where RM is a Reasoning Method} \qquad \textbf{3-14}$$

Chapter 4

CAIV Architecture and Design

4.1 Introduction

This chapter describes the CAIV architecture and design. It presents the architecture of CAIV by illustrating the core components, parameters, trust, scenarios and fuzzy concept of CAIV. It shows how the CAIV framework mimics human beings in taking a decision to decide the user level of trust. Furthermore, it shows how the system is capable of utilising and exploiting available context to find the trust level. Knowing the trust value level makes the system capable of verifying user identity by trusting or distrusting a user's activity.

4.2 Overview of the CAIV Inference

Reasoning is the practice of inferring information about some unobservable aspect of a situation based on information about the observed parts of the situation (Bhatnagar and Kanal, 1992). There are several methods can be used to build the reasoning engine. Fuzzy logic is chosen for the ensuing reasons in section 4.6 There are two types of fuzzy inference systems, which are Sugeno and Mamdani. Mamdani has been used to infer the trust value of the user since it uses membership function rather than constant function.

There are seven types of inference methods: deduction, induction, default reasoning, non-monotonic reasoning, analogy, abduction and heuristics. Deductive logical reasoning relies on previous facts to derive a conclusion. Heuristics are based on rules of thumb which are extracted from experience (Novák, 1999).

The heuristics approach is used to find CAIV's fuzzy rules. Later on, these rules are utilised by a deductive logic approach in the reasoning process. Ross (2009) mentioned that fuzzy logic is primarily used in deductive reasoning.

4.3 Components of an Identity Inferring System

There are different approaches to classify context parameters (Perera *et al.*, 2014). Abowd et al. (1999) believe that there are two types of parameter: primary parameters and secondary parameters. However, others (Henricksen, 2003; Van Bunningen, Feng and Apers, 2005) have a different classification. We agree with the first approach, yet we defined the context parameters slightly differently.

The primary parameters have been acquired from the pervasive environment based on W4H classification for context data (Truong, Abowd and Brotherton, 2001). We advocate their approach with some modifications to mapping them into our system:

- Who are the activity participants? (Identity) (Ambient Objects)
- Where does the activity take place? (Location) (Event Location)
- When does the activity take place? (Time) (Event Time)
- What does the user prefer? (User Preferences)
- How was the user behaving before? (History Previous Logs)

ambient objects, calendar event, user preference and history log are used as inputs to the system to infer (verify) the user's identity as shown in Figure 4.1 below.

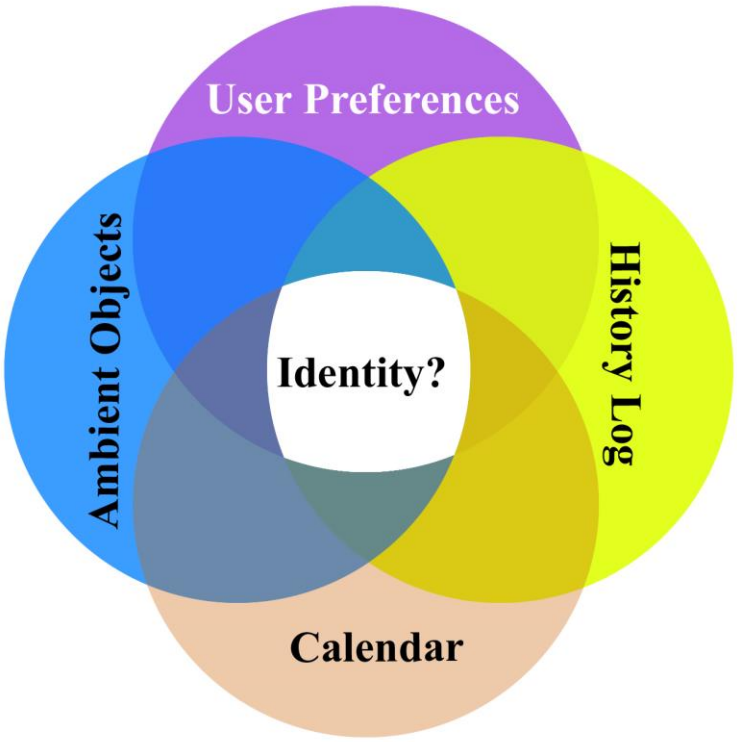


Figure 4.1 : CAIV Parameters Union to Infer User Identity

4.3.1 User Preferences

User preferences are the locations that the user would like or prefer to visit if he/she has spare time in order to tailor a system based on them. There are two approaches to finding user preferences. It can be fixed or dynamic. The fixed one can be aggregated from the user directly during the user's registration process; the system asks the user to answer a few questions. These questions are related to the user's preferences and hobbies, if applicable. Consequently, these answers are used to weight the destination locations which are related to the preferences.

For instance, ranked questions with a rating scale (seldom, occasionally and frequently) can be used to feed the system with the user preferences which consequently can be utilised as input to the CAIV system; the rating scale is respectively ranked as low, medium, high. The other approach is used to infer a user's preferences without user intervention such as that used by Lin and Fu (2007), which used Bayesian Network in the learning process; the data are aggregated over a period of time and then the system learns from these data the user preference. In the CAIV's case, the first approach has been chosen because of the lack of data and that help to speed up the test and evaluation process. Table 4-1 shows the sample of three answers of a user's questions about preferable places that they are likely to visit if they have spare time.

Table 4-1: Sample Data of User Preferences

Preference variables	**Preference**	**Preference's question Answers**	**Location representation**
P_1	Reading	Frequently	L_1
P_2	Play Tennis	Seldom	L_2
P_3	Drink coffee	Occasionally	L_3

4.3.2 Calendar

The calendar organises users' tasks and helps them to fulfil their commitments at the work environment. It includes all important notes and events. It records notes (e.g. event, task, meeting, lecture and others) which are noted with an associated location and time. These notes are stored in a specific form in the system, such as iCalendar. The event should be completely noted on the calendar, and any missing parts (location or time) make that event unusable. The standard iCalendar is used to make the proposed solution interoperable with any existed solution (system) that already uses iCalendar. The iCalendar has three main types of event. Journal entry (VJOURNAL), To-do (VTODO) and Events (VEVENT). They are ranked as low, medium, high, respectively. Table **4-2** is the sample data of the calendar.

Table 4-2: Sample Data of Calendar

Calendar variables	**Activity**	**Location**	**Event type**
C_1	Meeting	L_1	VTODO
C_2	Workshop	L_2	VEVENT
C_3	Meet Alice at Lobby	L_4	VJOURNAL

4.3.2.1 iCalendar

Events play an enormous role in the CAIV system, therefore, choosing a proper event standard is essential for two reasons. First, it makes the solution portable and interoperable, and second, the standard solution gives robustness to the proposed solution by complying with a well-known standard. For these reasons, iCalendar has been chosen. iCalendar is a unified format of any event.

It is used to send tasks and events between internet users. iCalendar was proposed by Dawson et al. (1998) and published as RFC2445 by the IETF.

Events (VEVENT) describes an event, for a specific period of time which has a DTSTART as a start time and DTEND as an end time. This event can be taken into account after the user accepts the calendar event. **To-do (VTODO)** is a long-term event which can be achieved by additional time. The SEQUENCE element is used to maintain the number of times it shows how many times the event has been modified. **Journal entry (VJOURNAL)** is a journal entry. It works as a note reminder. It does not affect the status of the user (Free, Busy). Below is a sample file of iCalendar version 1.0 (Dawson *et al.*, 1998).

ICalendar sample code:

```
BEGIN: VCALENDAR
        VERSION: 1.0
        BEGIN: VEVENT
                CATEGORIES: MEETING
                STATUS: TENTATIVE
                DTSTART: 19960401T033000Z
                DTEND: 19960401T043000Z
                SUMMARY: Your Proposal Review
                DESCRIPTION: Steve and John to review newest proposal material
                CLASS: PRIVATE
        END: VEVENT
END: VCALENDAR
```

4.3.3 Ambient Objects

Ubiquitous technology is growing exponentially and appearing everywhere. Therefore, this technology needs to be exploited in CAIV's approach to facilitate a user's identity verification. It exists in many different things around us in the pervasive environment, such as smart objects and devices. These objects are most likely to be equipped with a connectivity technologies; these technologies are Wi-Fi, BLE, RFID, NFC, etc.

In the CAIV method, ambient objects are classified based on possession into three types. The ambient object is owned by only one user, a location or by a group of users. An ambient object means any object(s) surrounding the user during the identity verification process which is (are) detectable by the CIAV terminal sensors. Schilit et al. (1994) described the ambient object implicitly by "who you are with". The possession is assigned to the object in the registration phase; Table **4-3** shows three different objects with three different possession types.

Table 4-3: Sample Data of Ambient Objects

Ambien Object Identifier	**Object**	**Possession**	**User ID**
A_1	Laptop	Owned by User	ID1
A_2	Book	Owned by Group of Users	ID1
A_3	Mug[1]	Owned by Location	-

(1)The canteen is the location of the Mug

4.3.4 History Logs

A user's history can reveal a certain behaviour of the user in the pervasive environment. It can tell the potential action of a user by relying on previous activities. There are different approaches to exploit history data to predict the future. The user history log has been used in CAIV to find the probability of the user to attend a location with time, the system learns about the user to build enough knowledge (history) of users. This knowledge may vary depending on context sizes (environment) such as number of locations and number of events per day. Once the system learns its knowledge about a user, it becomes capable of predicting the next destination of the user based on prior knowledge. There are different prediction techniques that can be used to infer the next location. CAIV uses Markov Chain as a proof of concept. Table **4-4** shows a sample of pre-visited locations of a user history log.

Table 4-4: Sample Data of User's History Logs

History Log ID	Pre-visited Location	Next Location	User ID
1	l_1	l_6	1
2	l_1	l_5	1
3	l_1	l_8	1

4.4 CAIV Trust Algorithm

The CAIV trust algorithm maintains four parameters as input and one single output. The four parameters are user preferences, ambient objects, history logs, and calendar event; Algorithm 1 gives an overview of the trust algorithm. There are four functions to aggregate the probability value of each of the context parameters which represent a part of the truth by using a Find function. This function relies on the user's identity, time and location to find related information about the identity holder. Later, the find_trust_value function finds the final value of trust which is resulted based on the fuzzy inference engine. The value of trust is compared against a preset threshold value. Finally, the user status returns as an output of the algorithm.

Algorithm 1: CAIV Algorithm

for all Login activity **do**

If User is valid user **Then**

    A=Find (User preferences weight for this activity)

    B=Find (Ambient Object weight for this activity)

    C=Find (History weight for this activity)

    D=Find (Calendar weight for this activity)

    Trust = find_trust_value(A,B,C,D)

    If (Trust ≥ threshold **):**

        User_Status=Trusted User

    Else

        User_Status=not-trusted user

    Output User_Status

Else

Output (Unregistered User)

CAIV architecture has three main layers. The first layer is the aggregation layer which is responsible for aggregating the related data from databases (logical sensors) or the available data from physical sensors such as Bluetooth beacons, RFID antenna or any other sensor. The acquisition data tagged with an associated parameter of the context and transferred to the next layer. The second layer is the reasoning layer, which is responsible for retaining the extracted rules

and applying them into the aggregated data to find the trust value. Reasoning layer act as a CAIV brain, contains the main reasoning method which is tailored to fit the CAIV context parameters. This layer manipulates the received data and send the result to the next layer. Finally, the presentation layer which represents the output layer. It works as interface between the user and the core system. It obtains the trust value from the second layer and compares it against the threshold value. The threshold value is customisable by the system administer. Eventually, it returns the user activity status. The status can be a failure or success message displayed on the terminal screen, an alarm(alert), trigger an action (such as send a signal to open an electronic lock), and so on. Figure 4.2 shows the CAIV architecture, while Figure **4.3** shows the general flow of the CAIV method.

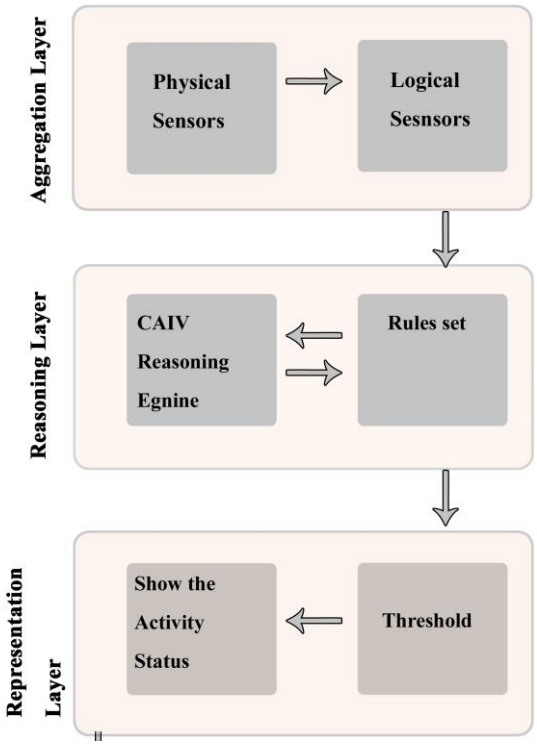


Figure 4.2 : CAIV Architecture

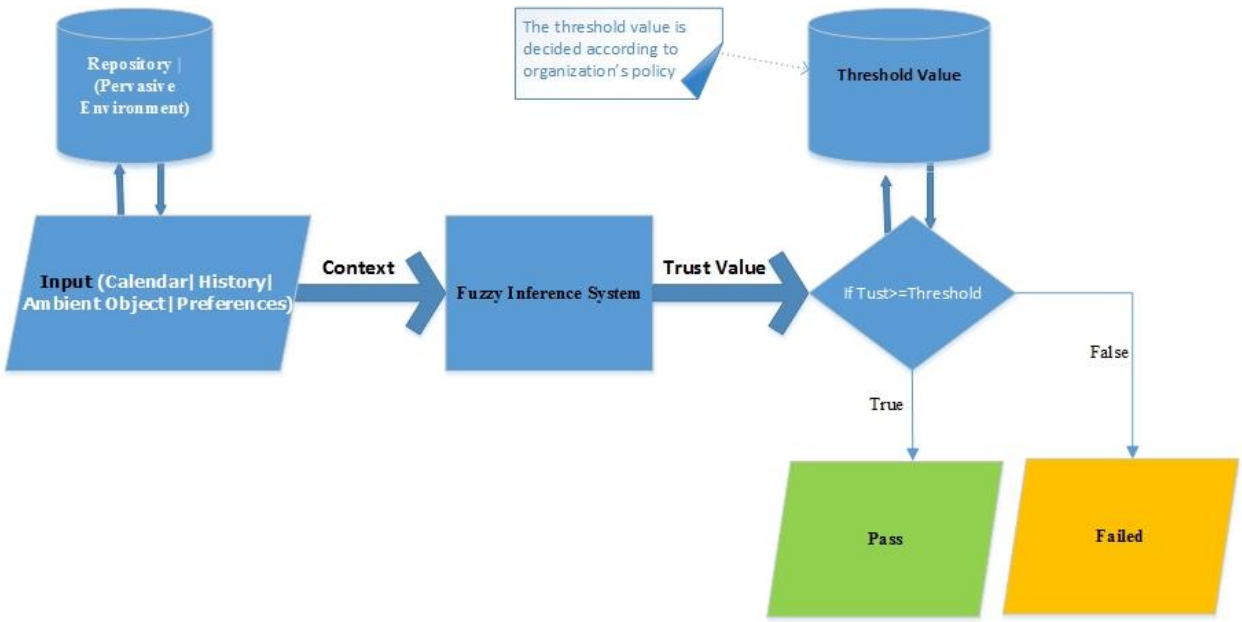


Figure 4.3: General Flow of CAIV Verification Process

4.5 CAIV Scenarios

A scenario is an essential and descriptive tool for brainstorming ideas, thoughts and visions of any deployment environment, especially when this environment does not exist or is out of the researcher's budget and time scale. In the literature, there are different definitions, explanations and uses of scenarios. They help us during the design phase of getting some of the system's requirements and at the same time validating the CAIV approach.

Carroll (2000) defined a scenario as "A direct approach is explicitly to envision and document typical and significant activities early and continuingly in the development process. It supports reasoning about situations of use, even before those situations are actually created"

Potts (1995) defined scenarios as narrative descriptions of interactions between users and proposed systems; concrete scenarios help users and designers to develop a shared understanding of the proposed system's functionality.

Carroll (2000) described the criteria which can be used to define the optimum number of scenarios as "A set of scenarios has good coverage if it includes examples of the significant uses of a system and the major types of agents, goals, actions, events, obstacles, contingencies, and outcomes that constitute these uses", which is quite similar to Potts definition.

Alexander & Beus-Dukic (2009) divided scenario building into three phases, discovering, documenting and validation phases, as shown in Figure 4.4. The discovering phase can be achieved through three contexts methods, which are the interviews, observation and workshops. They also divided the discovering scenarios into normal and negative scenarios. The first type is the normal scenario "happy day", while the latter can be exceptions, intentional threats or unwanted scenarios.

Figure 4.4 : Building Scenarios Phases

The documenting phase represents the way that the developer depicts the acquired scenarios which are consequently quantified by a number of details of scenarios. They divided scenario documenting into four types: unstructured stories, storyboards, operational scenarios and use cases.

The validation phase is used since scenarios are a basic way for operational stakeholders to describe how they work and how they want to work in the future. The point of validating scenarios is to ensure that the same stakeholders are happy with what has been documented. Alexander and Beus-Dukic stated that there are four methods to validate the produced scenarios; these methods are scenario walkthroughs, animation, simulation and prototyping.

Moreover, Alexander & Maiden (2005) stated that simulation is capable of giving precise answers about whether such a scenario can be realised with any plausible design. Therefore, building the CAIV's simulator is for validating the scenarios and, consequently, the whole approach.

4.5.1 Scenarios

Every system has a large number of usage scenarios. Therefore, it is necessary to have a limited number of scenarios which are capable of representing the whole relevant system cases. According to Caroll, sound scenarios should cover the significant uses of a system, therefore, the CAIV proposed scenarios cover all CAIV parameters and their combinations. The parameters have been used to derive and develop the proposed scenarios. Since the scenarios need an environment to use as a case study, a brief description of the selected environment and consequent proposed scenarios are illustrated below.

The description of a case study domain of the pervasive environment, which uses iCalendar, ambient objects, user preferences and user's history log as a knowledge base is as follows. It is assumed that the pervasive computing environment domain includes one building, B1. The building has a number of rooms, numbered from '01' to 'n'. There are two main entrances to the B1 building. Everybody has access to the first entrance; it leads to the reception front desk. The

second entrance is only accessible by registered users. The system uses context (the four parameters) to verify the user's identity. The user is asked to verify(prove) their identity by another method (additional credential(s)) such as pin code, fingerprint or any other method if the system CAIV method fails to verify their identity. RFID card used in the domain as a user identifier.

There is a certain level of trust (the threshold) that the user needs to reach to make the system more confident of their identity (to trust the user). On achieving this level of trust, the system does not ask the user for further credentials to verify their identity and it will settle for the RFID card only as user's identification. The trust level is a range between 0 and 1; it can be chosen according to the system's policy which can be low, medium or high, which is called system threshold. The trust level can be decided according to certain criteria. Moreover, the trust value which is aggregated from the context should be greater than or equal to the threshold value (the trust level). Table **4-5** shows the possible rules combinations of context parameters; the next section gives operational scenarios of the CAIV system.

An operational scenario is an engineering story that describes operations that the product or service is intended to support. The real value of this is to show developers, who may be far from actual operations, what result they should be trying to achieve, rather than a set of disconnected features.

Table 4-5: Parameters Combinations Table

ID	Calendar	User Preferences	Ambient Objects	History
#1	✓	-	-	-
#2	✓	✓	-	-
#3	✓	✓	-	✓
#4	✓	✓	✓	✓
#5	-	✓	-	-
#6	-	✓	-	✓
#7	-	✓	✓	✓
#8	-	-	-	-

Key:

The parameter is not existed -

The parameter is existed ✓

Scenario 1: The time is 2:55 pm, Bob is heading to B1 to reach room no. 3; he swipes his ID card in order to verify his identity and enter the room. The system checks his calendar and finds that there is an event noted on Bob's calendar. This event shows that Bob has an appointment in B1, room no. 3 with Alice at 3:00 pm. It is five minutes before the appointment. The system lets Bob in by opening the door lock; the system tolerance time is up to 15 minutes before or after the exact time.

Scenario 2: The time is 9:55 am, Bob is heading to B1 to use the gym. The system checks his calendar and there is no event related to this location. Then the system moves to the preferences

and finds that the gym is one of Bob's preferred locations. The system checks Bob's status and it is active. At this point, the system lets Bob in by opening the door lock without asking him for more credentials (the system trusts Bob).

Scenario 3: Bob borrowed a book from the library. He is heading to his room. He is holding the book. He swipes his card to enter his room. The system cannot find any event on the calendar or a preference related to this location. The system detects an object near Bob. It finds that object (the book) is related to Bob (an object belonging to a user). The system lets Bob enter the room.

Scenario 4: Bob has an experiment to do at Mike's lab; it is Lab 1. Mike hands him two of the experiment components to help him run the experiment in his lab. Bob swipes his ID to access Lab 1. He has an active status but he does not have preferences, a calendar event or a history log to make the system trust him (verify his identity). The system detects two objects belonging to Lab 1 near Bob. Bob reaches a certain level of trust. The system lets him enter the Lab.

Scenario 5: Bob has left his office and is heading to Lab 3. He swipes his ID to get into the lab. The system cannot find any related information from the context except the history. Bob, for the last four weeks, has gone from his office to Lab 3 when he has a chance to do this. The system lets Bob enter Lab 3.

Scenario 6: Bob forgets his ID card at home. He tends to enter the main office. He holds his gadget (smartphone, smartwatch, smart glasses, etc.) with enabled Bluetooth. He was previously asked by the system whether he desires to use these devices as an alternative identifier in the future and he accepted. Bob realises that he has forgotten his ID card. He uses his gadget to confirm his identity. The system verifies his identity and starts looking for a relevant event, preference, ambient

object or history. The system has detected Bob's smartphone and the main office is one of Bob's preferences. The system lets Bob enter the main office.

Scenario 7: It is 2:30 pm. Bob would like to do some exercises so he goes to the gym and swipes his ID. The system finds that the gym is one of Bob's preferences and he has some history logs which show that Bob used to access the gym repeatedly after leaving the school lobby. The system lets Bob in.

Scenario 8: It is 12:30 pm. Bob reaches the canteen door; he swipes his ID to enter the canteen. The system detects Bob's mug which he took with him yesterday to his office as usual. Moreover, the system found some history logs which show that Bob used to attend at the canteen hall after leaving his office. The system trusts Bob and lets him in without further verification.

Scenario 9: Viv heads to the conference room to attend a conference session. The coordinator hands her the seminar portfolio and conference attendance badge (a temporary ID card). Viv is heading to the conference hall. The system detects that Viv holds two things: a seminar portfolio which is linked to the meeting room and an attendance badge which is also related to the same location (conference hall) and identifies Viv as a conference contributor. The system lets her in.

Scenario 10: Viv attends a conference situated at B1 building but she is not registered in the system. The organiser issues a new badge to Viv. It is valid for the whole event duration. Moreover, with the calendar event she does not need further credentials to access any facility within the event locations. Viv would like to explore the place and she tries to enter a facility which is not one of the event locations. The system will not let her in.

Scenario 11 (Exception): Bob presents his slides in conference room 2. While he is walking through the room, the ID falls from his pocket. One of the audiences picks up the ID and leaves the presentation. He heads to Lab 1. He swipes the ID. The system checks and finds that Bob is not used to go to Lab 1 when he leaves conference room 2 and there is no event, preference or an object belong to this location. The system raises the alarm.

Scenario 12 (Exception): Bob ends his workday and has permission for two days' vacation. He goes to the car park, while he is trying to grab his car key from his pocket the ID falls on the ground. He goes home and he does not notice that he lost his ID. Trudy finds the ID in the car park. The next day, Trudy (the intruder) wants to impersonate Bob's identity. She tries to enter B1 but the system denies her access since Bob is on vacation (inactive status) based on the calendar information.

Scenario 13 (Exception): Trudy stole Bob's ID card while Bob was having his cup of tea at the canteen. She reaches Lab 1 and swipes Bob's ID card. The system cannot find any related information to this activity. Therefore, it requests a pin code to verify his identity. Trudy cannot access Lab 1 since she does not know the pin code. The system raises an alarm alert notification to the administrator.

4.6 The Reasoning Layer Design

Fuzzy logic was first introduced by Lotfi A. Zadeh in 1965. Zadeh (1965) defined a fuzzy set as a class of objects without a precisely defined criterion of membership. The difference between fuzzy logic and classical logic (crisp logic) is that classical logic has two truth values, false and true, while the truth values of fuzzy logic are described by linguistic terms such as high, low, moderate and so forth (Bělohlávek and Klir, 2011). Furthermore, they deemed that classical logic is a subset of fuzzy logic. This leads to fuzzy logic being represented as a generalisation form of crisp logic.

Van Leekwijck & Kerre (1999) divided fuzzy systems into two categories: fuzzy reasoning (or fuzzy knowledge) systems and fuzzy control systems. Fuzzy knowledge systems aim to provide some qualitative reasoning system for a specific domain. Fuzzy sets are used to map qualitative facts onto numerical entities that can be manipulated by the computer. The result of a "computation" in these systems is a qualitative expression based on the input to the system.

Fuzzy controllers always need a crisp value as a result; a result stating that a certain valve has to be opened "somewhat" is not very useful in a control system. Fuzzy sets are used as a convenient tool to define control rules and to make inferences. Figure **4.5** shows the general flow of a fuzzy system.

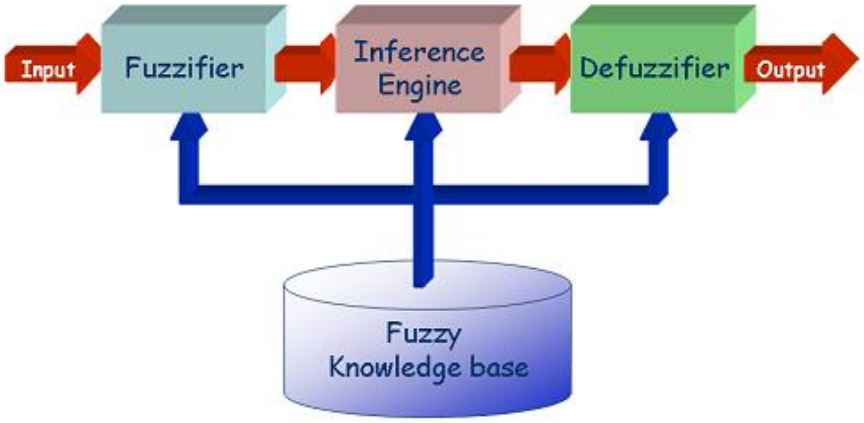


Figure 4.5: Overview Diagram of a Fuzzy System

Fuzzification process is to convert crisp value to fuzzy value. While the defuzzification is the conversion of a fuzzy quantity to a precise quantity (crisp value). There are many methods which proposed for defuzzifying fuzzy output function, such as max membership principle, centroid method, weight average method, and mean max membership (Ross, 2009).

Table **4-6** shows that fuzzy logic flexibility and reliability are the reasons behind choosing fuzzy logic to infer a user's identity. Added to this, it does not need a large dataset and it is more understandable and traceable by the developer while some other approaches need a big dataset and are regarded as a black box so the result cannot be explained or backtracked. Fuzzy logic can hold the uncertainty of CAIV model in finding the trust value.

Table 4-6: Comparison of Techniques in Terms of Modelling Capabilities (Gray and MacDonell, 1997)

Technique	**Model-free**	**Can resist outliers**	**Explains output**	**Suits small data sets**	**Can be adjusted for new data**	**Reasoning process is visible**	**Suits complex models**	**Includes known facts**
Least squares regression	X	X	~	X	X	✓	X	~
Robust regression	X	✓	~	~	X	✓	X	~
Neural networks	✓	X	X	X	~	X	✓	~
Fuzzy systems (adaptive)	✓	~	✓	✓	~	✓	✓	✓
Hybrid neuro-fuzzy systems	✓	~	✓	~	~	~	✓	✓
Rule-based systems	X	N/A	✓	N/A	N/A	✓	✓	✓
Case-based reasoning	✓	~	✓	~	✓	~	✓	X
Regression trees	✓	✓	✓	~	✓	~	✓	~
Classification or decision Tress	✓	✓	✓	~	✓	~	✓	~

Key:

Yes ✓

No X

Partially ~

Furthermore, Pradhan & Pirasteh (2010) have validation results that show that the prediction accuracy of fuzzy logic and neural networks are slightly different; the accuracy for both of them are around 90%. Their result shows that fuzzy logic can be as reliable as a neural network.

Bělohlávek & Klir (2011) defined a fuzzy set as appropriate sets of numbers that represent linguistic terms such as many, most, almost all, very few and so forth.

Given a fuzzy set A defined on a universal set U.

U: A(x) >0 is …. Called a support of A

U: A(x) =0 is …. Called a core of A

U: A(x) not empty is …. Called a normal fuzzy set

All other fuzzy sets are called subnormal.

Level cut α-cut of A (A_α) is a classical set of all objects x of U for which $A(x) \geq \alpha$

$A_\alpha = \{x \in U: A(x) \geq \alpha\}$

4.6.1 Fuzzy Rules

Information is step forward knowledge; the latter is the core of a rule-based approach. Fuzzy rules are generated from two sources: numerical information acquired from sensors and linguistic information gained from experts (Wang and Mendel, 1992). Wang & Mendel stated that they could rely on the experience of human beings to extract some rules. Figure 4.6 shows the fuzzy expert system architecture by describing the knowledge(rules) extraction and the dialogue and validation processes between the knowledge engineer and the experts.

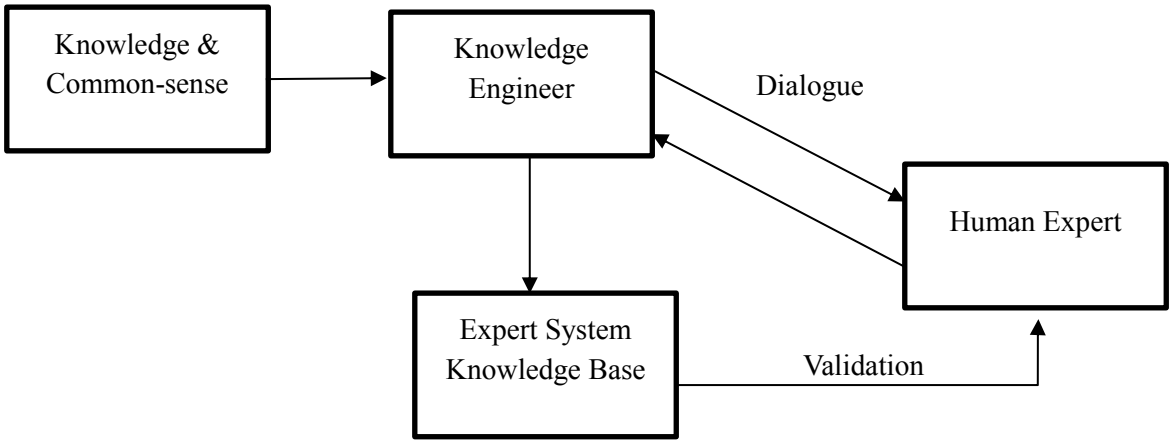


Figure 4.6: Architecture of a Fuzzy Expert System

Hong & Lee (1996) stated that "to apply expert systems in decision-making, having the capacity to manage uncertainty and noise are quite important". They also stated that fuzzy logic is frequently used in expert systems because of its simplicity and similarity to human reasoning. Additionally, they describe the expert as a crucial factor in building expert systems. They stated that experts and experienced users contribute to building the member function.

There are two main approaches of a FIS (fuzzy inference system), namely, the approaches of Mamdani (Mamdani and Assilian, 1975) and Sugeno (Takagi and Sugeno, 1985). The differences between the two approaches arise from the following. Mamdani's approach uses fuzzy membership functions (MFs) whereas Sugeno's approach uses linear or constant functions. Since the CAIV approach does not have a constant function, a Mamdani approach is utilised for the CAIV inference engine.

The CAIV inference engine is designed on Matlab. Figure **4.7** shows the fuzzy decision engine inputs and output. The setting of the inputs and the output membership function set based on the CAIV membership functions. The CAIV rules fed into the Matlab tool box. As CAIV method uses Mamdani inference, the defuzzification is used the centroid method.

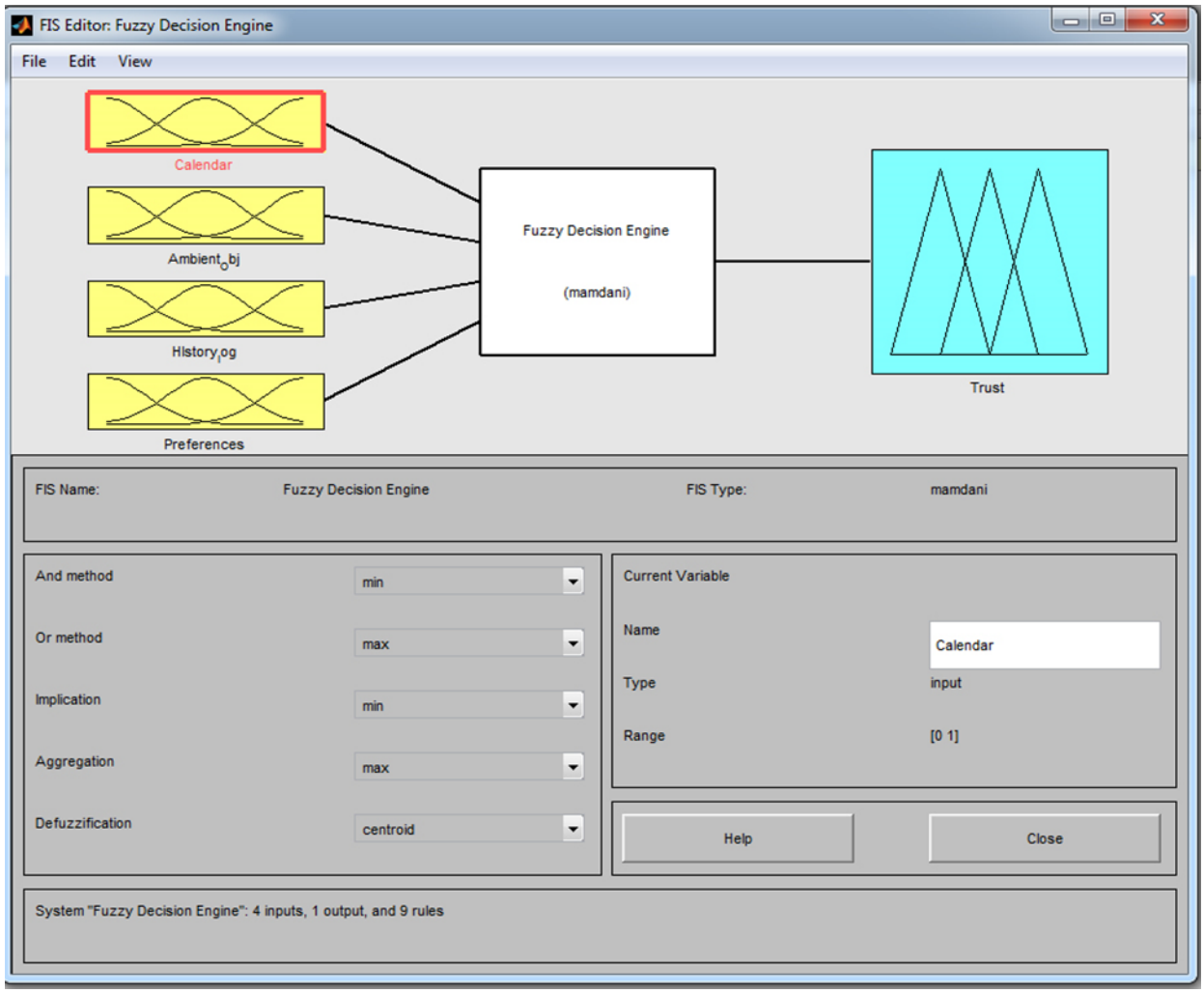


Figure 4.7 : CAIV Fuzzy Trust Model

Figure 4.8 shows the fuzzy rules of the inference engine in the Fuzzy Matlab toolbox. CAIV has 81 rules. These rules were discussed with experts. The CAIV is designed with the ability to

rectify these rules in the future, in case the method performance is low or is breached because of some inaccurate rule.

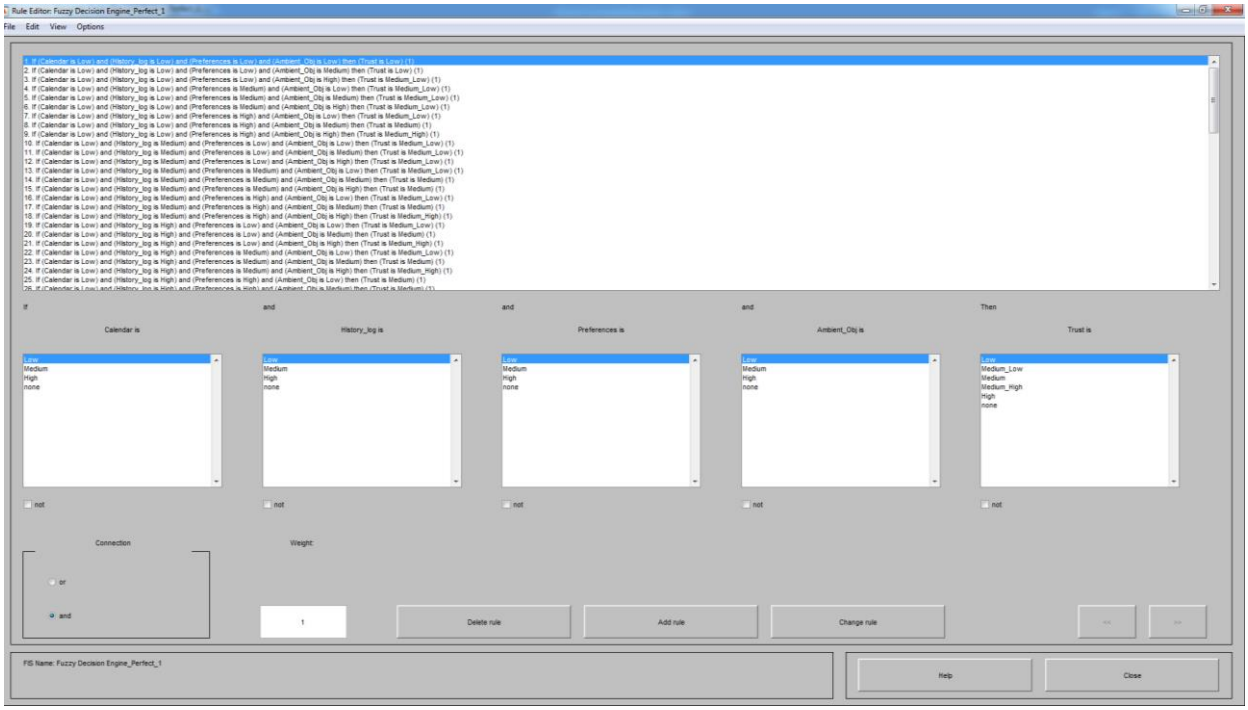


Figure 4.8 : Fuzzy Decision Engine Rules

Below are the sample the 81 rules which are used in the Matlab rules engine, Appendix A illustrates the list of the rest of the rules:

1- if (Calendar is Low) and (History Log is Low) and (User Preferences is Low) and (Ambient Object is Low) Then (Trust is Low)

2-if (Calendar is Low) and (History Log is Low) and (User Preferences is Low) and (Ambient Object is Medium) Then (Trust is Low)

……

81- if (Calendar is High) and (History Log is High) and (User Preferences is High) and (Ambient Object is High) Then (Trust is High)

4.6.2 Membership Function

The output membership function is divided into five intervals based on Lesani & Montazeri. Lesani & Montazeri (2009) divided the membership function of trust into five intervals with five linguistic variables, low, medium low, medium, medium high and high, as shown in Figure 4.9.

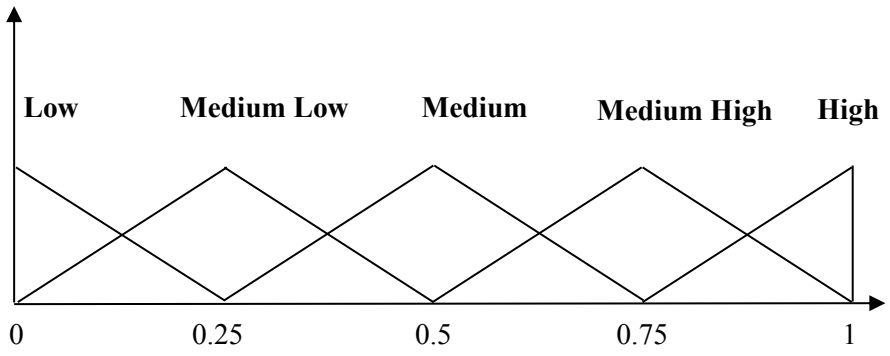


Figure 4.9 : Fuzzy Membership Functions of Trust Linguistic Terms (Output)

The input membership function for all parameters falls into three intervals, the lingustic variables are (Low, Medium and High) as shown in Figure 4.10. The experts defined the intervals as follow, 0.1-0.3 for low, 0.3-0.6 for medium, and 0.6-1.0 for high.

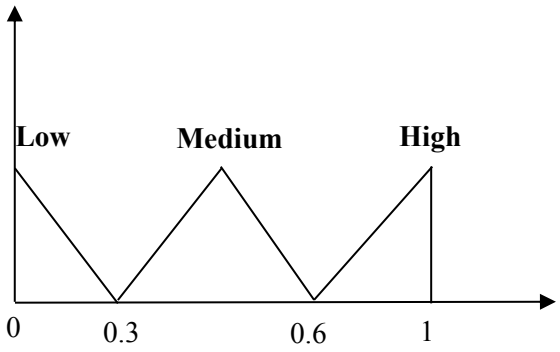


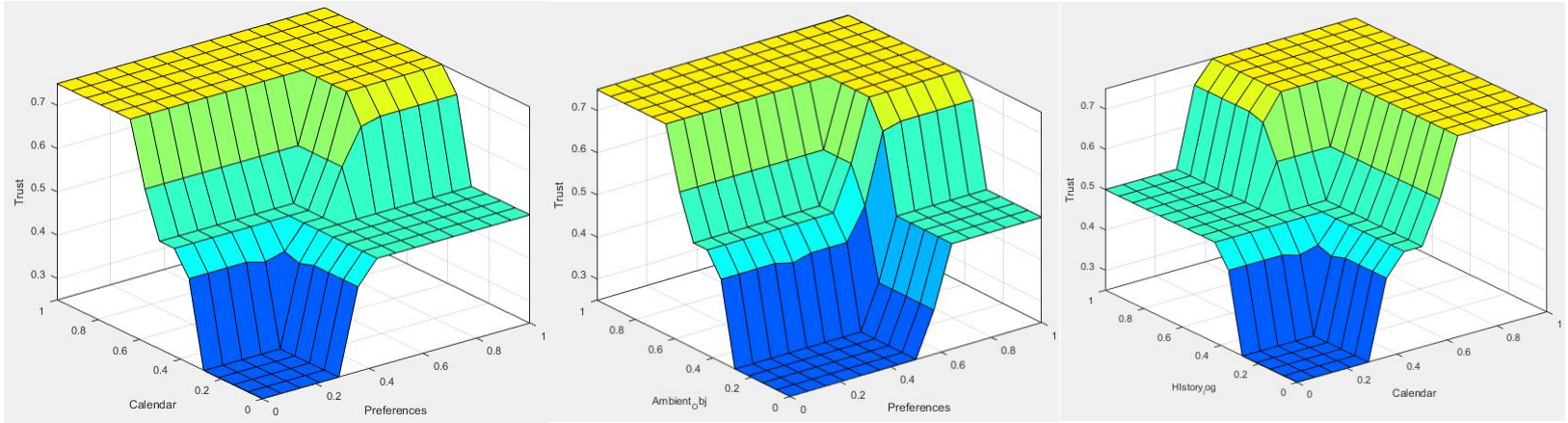
Figure 4.10 : Fuzzy Membership Functions of Input Linguistic Terms

The fuzzy inference engine is implemented on Matlab R2016a.

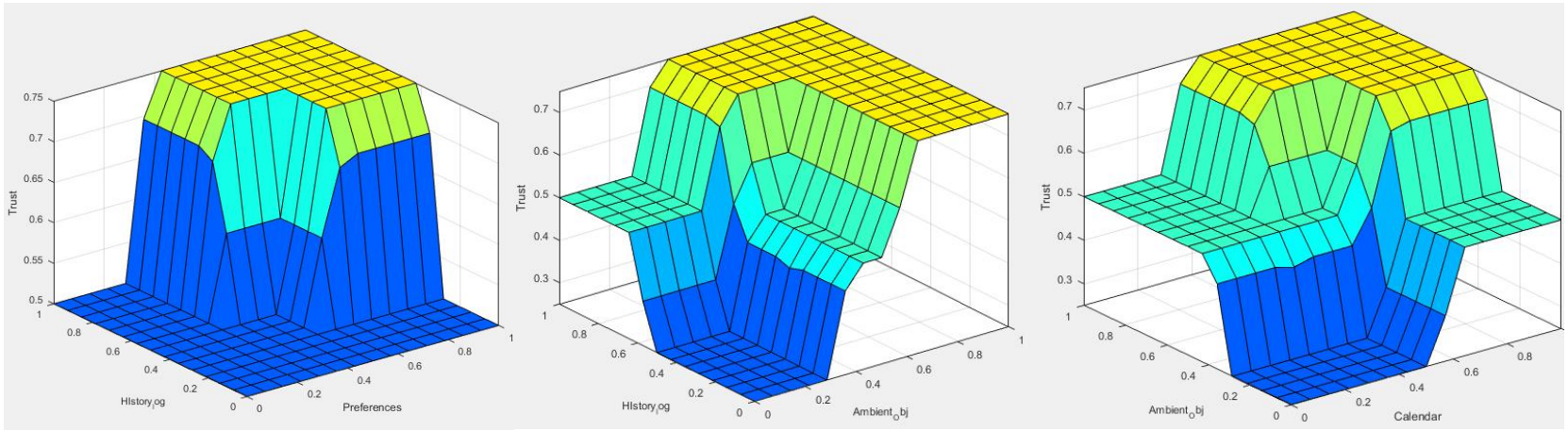
Figure 4.11 shows the six figures of interdependency between the context-parameter pairs (inputs) and the output (trust). These six figures were generated by the Matlab surface plot. Every parameter, when combined with other parameters, has a specific point(value) which influence on the trust value. The calendar parameter, combined with the history logs parameter and the preferences parameter, better influenced the trust value when the influence was started at 0.23, while it influenced the trust value less, when combined with the ambient object parameter, when the influence of trust value was started at 0.43. The history log parameter better influenced the trust value when it was combined with the calendar parameter, while it influenced the trust less when it was combined with the ambient object parameter and the preferences parameter. The preferences parameter, combined with the calendar parameter, better influenced the trust value

than it did when combined with the ambient object parameter and history logs parameter. The ambient object parameter exhibits a stable, and high, influence over all the other three parameters when the trust influence was started at 0.23. We conclude from the six combinations, involving two parameters each, which are shown in

Figure 4.11 that the influencing of the trust value by the context parameters exhibits different levels. They significantly affect the influence of the ambient object parameter on the output (trust), they have less of an effect on the calendar parameter's influence on the trust value and the least effects on the other two parameters, the history logs and the preferences. The influences which cause the input parameters to affect the trust values positively(better) cause the aggregated values of the input parameters to affect the trust value, and thus even input parameters with low values have some effect. On the other hand, the influences which cause the input parameters to affect the trust value negatively(less) are such that affect the trust value with quite moderate input value.



(a) Calendar and Preferences (b) Ambient Objects and Preferences (c) History logs and Calendar



(d) History Logs and Preferences (e) History Logs and Ambient Objects (f) Ambient Objects and Calendar

Figure 4.11: Interdependency between Two Context Parameters and Trust

Figure 4.12 shows a demonstration of giving four different values as input to the fuzzy rule engine and how that affects the trust value, which is situated at the right side of the Matlab Fuzzy toolbox snapshot. The calendar, history log, preferences and ambient object are 0.241, 0.5, 0.5, and 0.5 respectively, and the trust value (the output) was 0.5 as a result of rule 14.

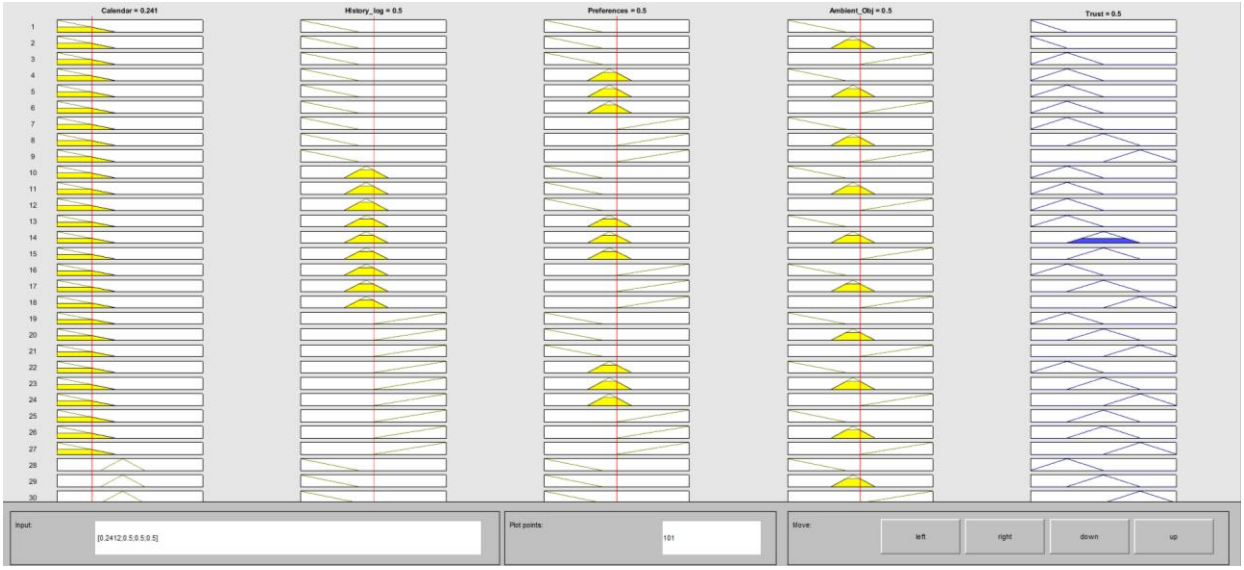


Figure 4.12: Rules Aggregations and Outputs

4.7 Summary

After selecting the input and output parameters, the aggregation technique of each of them is found by relying on literature and brainstorming which achieved through the operational scenarios. The potential result shows the interdependency between the input parameters and the output; it also shows the overall behaviour of the input parameters compared with the trust value. It shows the importance of both calendar and ambient objects in the CAIV method which are more distinctive than the other two parameters. The next step is the implementation and testing of the CAIV approach.

Chapter 5

CAIV Implementation and Testing

5.1 Introduction

To implement and test the CAIV method which is presented in Chapter 4, this chapter discusses the steps required to achieve this, such as preparing data, choosing and building the necessary environment to test CAIV approach and finally building the CAIV prototype. Simulation is a virtual representation of real-world scenarios which can mimic a real system process or operations of specific solutions run over time. It generates an artificial history of a system and builds up an environment to fulfil the whole functional characteristics of the system in a real-world implementation. The behaviour of the system is monitored over time. A simulation model needs to have a set of assumptions to establish a sound environment. Simulation can also be used to study systems in the design stage before such a system is built. Thus, simulation modelling can be used

both as an analysis tool for predicting the effect of changes to existing systems and as a design tool to predict the performance of new systems under different circumstances (IL, Barry L. Nelson, Jerry Banks, 2009). Richard Gran (2012) defined simulation as “the creation of a model that can be manipulated logically to decide how the physical world works”. The simulator needs a dataset (test data) to quantify the system performance.

5.2 Test Data Structure and Generation

We believe that is hard to find a dataset that uses the CAIV four parameters for two reasons. First, the data is sensitive and affects user privacy, and the second reason is availability as it is not possible to ask a company to adopt our prototype which is not yet tested. In order to test and evaluate the proposed framework, it is necessary to build a CAIV simulator and generate a dataset to feedback to it. The dataset is generated randomly. The CAIV simulator design has four essential phases as shown in Figure **5**.**1**, which are defining the input data, generating a dataset, performing deterministic computation on the input and aggregating the result.

There are two types of simulation: time-driven and event-driven.

Time-driven discrete event simulation has been used in CAIV. The time-driven approach needs a jump interval within a time frame (Lo, 2014); a 1-hour slot has been used for CAIV simulator intervals.

A random dataset was generated with 79 subjects (users). Each user has a set of attributes which are related to the work environment. Python programming language was used to generate this dataset and a Python standard random number generator library is used. The Python random library is based on the Wichmann-Hill algorithm (Wichmann and Hill, 1982) by relying on uniform distribution. The generated data have been stored in a MYSQL database. Later, the CAIV simulator runs the test data (dataset) to get and store the result.

5.2.1 Datasets

Since real data is not available, therefore, a synthetic dataset is used which is generated by the CAIV dataset generator that we developed. It is designed based on the following model:

Each calendar event e has a probability, p(e), of appearing in the daily calendar events. For each calendar event, the generator specifies four probabilities: (i) the probability of title, p(t|e), (ii) the probability of description, p(d|e,t), (iii) the probability of task type, p(tt|e,t,d) and (iv) the probability of location, p(l|e,t,d,tt). For example, we have a list of first names and last names, the probability to have a complete name is by picking up a random name from these two lists. That procedure is applied for the aforementioned features of the calendar.

For the ambient object generation, it is assumed that each user has a set of objects. Each set has six objects generated randomly; each object has four essential attributes of description, ambient object type and ambient object location which are assigned randomly.

Finally, user preferences are assumed to have three different preferences for each user. The probability of description, preference rate and preference location are assigned randomly.

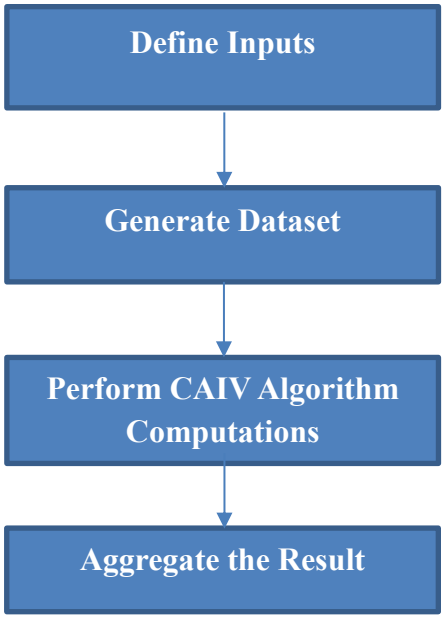


Figure 5.1 : CAIV Simulation Phases

The dataset generator executed on Python programming language and the results stored in the MYSQL database which is ready to be used by CAIV simulator.

5.3 Simulator Environment

The CAIV environment that we need to simulate should include all four CAIV parameters to be tested against all possible scenarios. The university environment can include the CAIV parameters, therefore the university information system has been chosen. It is assumed that the university information system is linked with an identity verification module which in our case is the CAIV module; the aim of this module is to verify the user's identity to grant users access to any secure location within the environment such as offices, lecture halls, labs, etc.

The CAIV module represents the main identity verification method of this environment. If the CAIV module failed to verify the user identity, the system would ask the user for more credentials to verify the user's identity, to avoid any user interruption. These credentials could be a fingerprint,

face recognition, passcode, etc. The passcode was chosen for use in the system to confirm the user's identity since it does not need any additional hardware to be added to the terminal.

The physical design of the domain is described as a university building which has three floors with different facilities as shown in Figure 5.2.

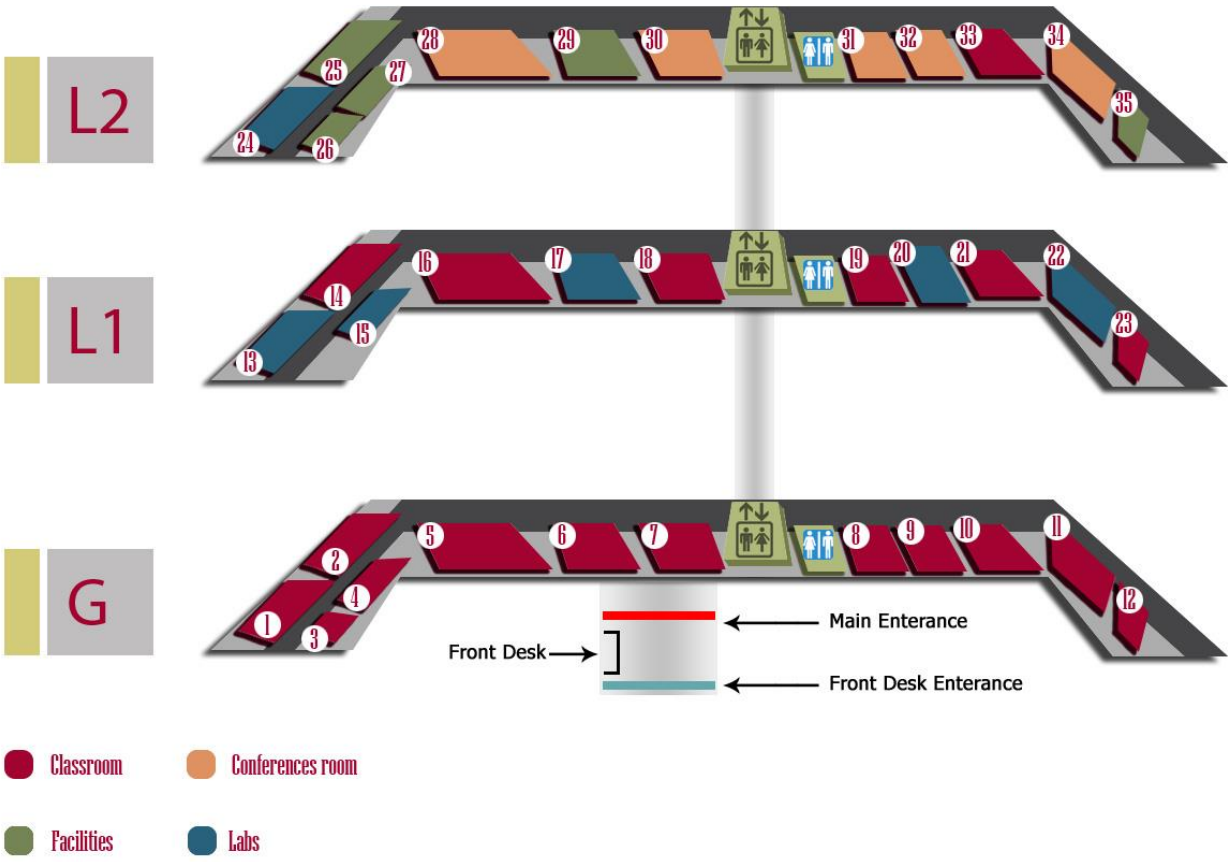


Figure 5.2: CAIV Environment Building Map

5.4 CAIV Environment System Design

The CAIV environment information system is designed with five essential classes which are user class, calendar class, history log class, user preferences class and ambient object class, as shown in Figure 5.3. There are two more subclasses linked with these five classes, which are the

authentication and location classes. Authentication class has a list of authentication credentials. The credential description (type) can be a password or RFID, while the value is the password (numbers) or RFID number (16-digits). Calendar class includes the information about an event such as event name, date, type, starting time and ending time which is assumed a one hour slot in the simulator. History log class includes all the user's attempts with time, location, user ID, log status, and parameters4; parameters4 used by experts to monitor the parameters behaviour and rectify the rules in the future. User preferences class contains the information of the preferable places of the user, such as location ID, description and the percentage of likelihood of the user to attend this facility (probabilityRatio). Ambient Object class illustrated the smart objects which have description, type, ID, user ID and location ID; the ID can be the physical or MAC address of the object.

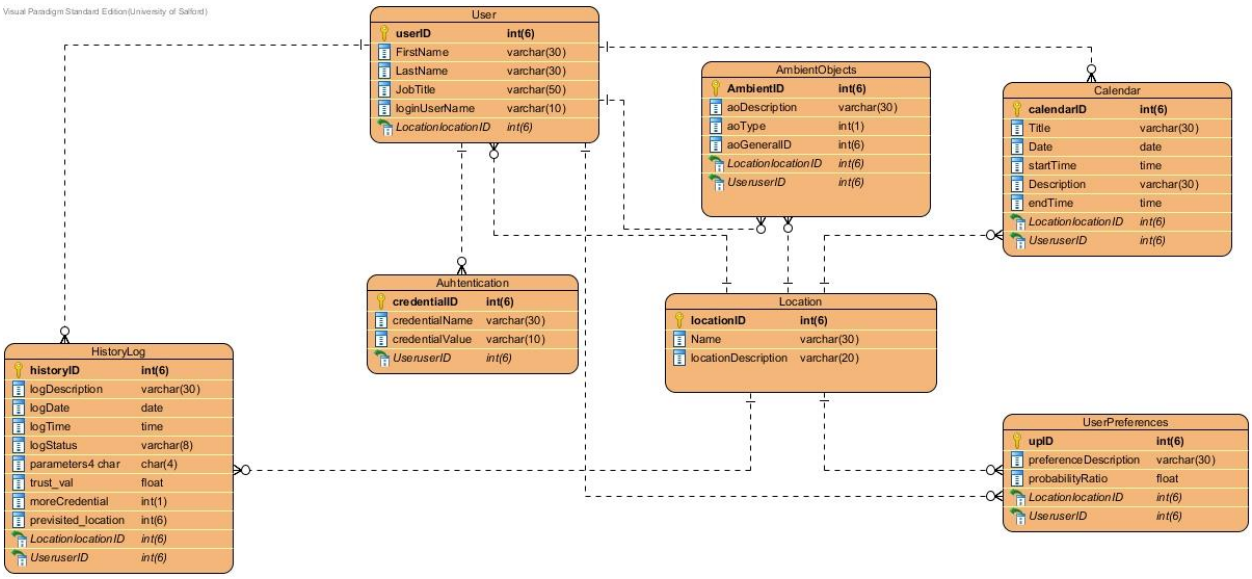


Figure 5.3: CAIV ERD

The state machine diagram in Figure 5.4 shows the interaction between the user and the CAIV module. It starts by swipe the RFID card of the user to the user interface (UI) which is the CAIV terminal. The control entity which represented by reasoning engine, aggregating the related data from the four context parameters and calculate the trust value (the output). Eventually, send that output to the user interface to trigger an action, for example, the UI returns the status as a trusted user and send a signal to the electronic lock to open it.

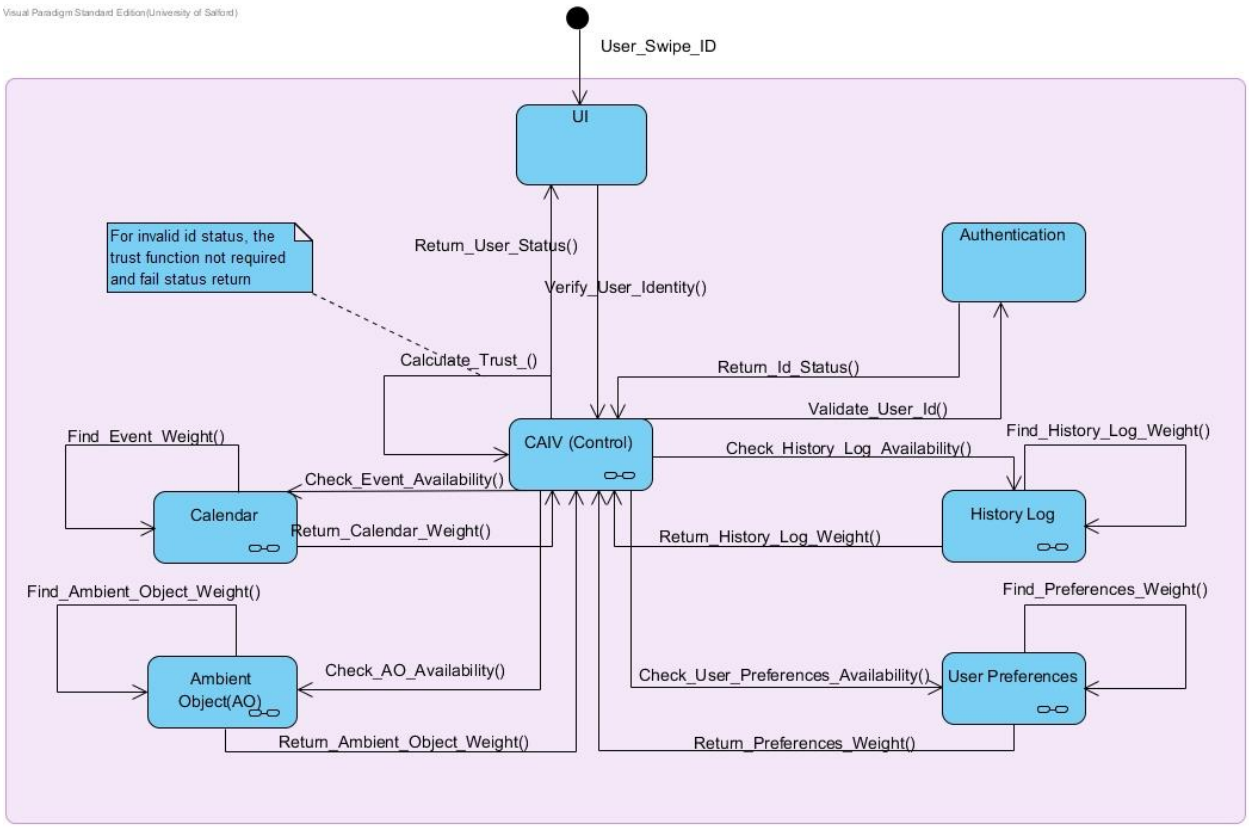


Figure 5.4: State Machine Diagram of CAIV

5.5 CAIV Simulation Module

It is necessary to design a simulator to evaluate the potential performance of the CAIV method. This section reviews the specification of the simulation module which is used as a case study to test and evaluate the CAIV performance. It aims to investigate the performance of the CAIV approach though mimicking the potential real-life scenarios of users by simulating a university information system using the CAIV method as the main method to verify user identities in an everyday life scenario. The CAIV simulator development has two main phases. The first phase is to generate the dataset and the second phase is to build up the simulation engine which can be run based on the generated dataset.

The dataset was generated randomly for 79 users and stored in the information system database. The information system has seven main MySQL tables (user, calendar, history log, ambient objects, user preferences, locations and authentication). It represents an information system used in the university. Python is the programming language platform of the CAIV dataset generator, CAIV simulator and CAIV prototype.

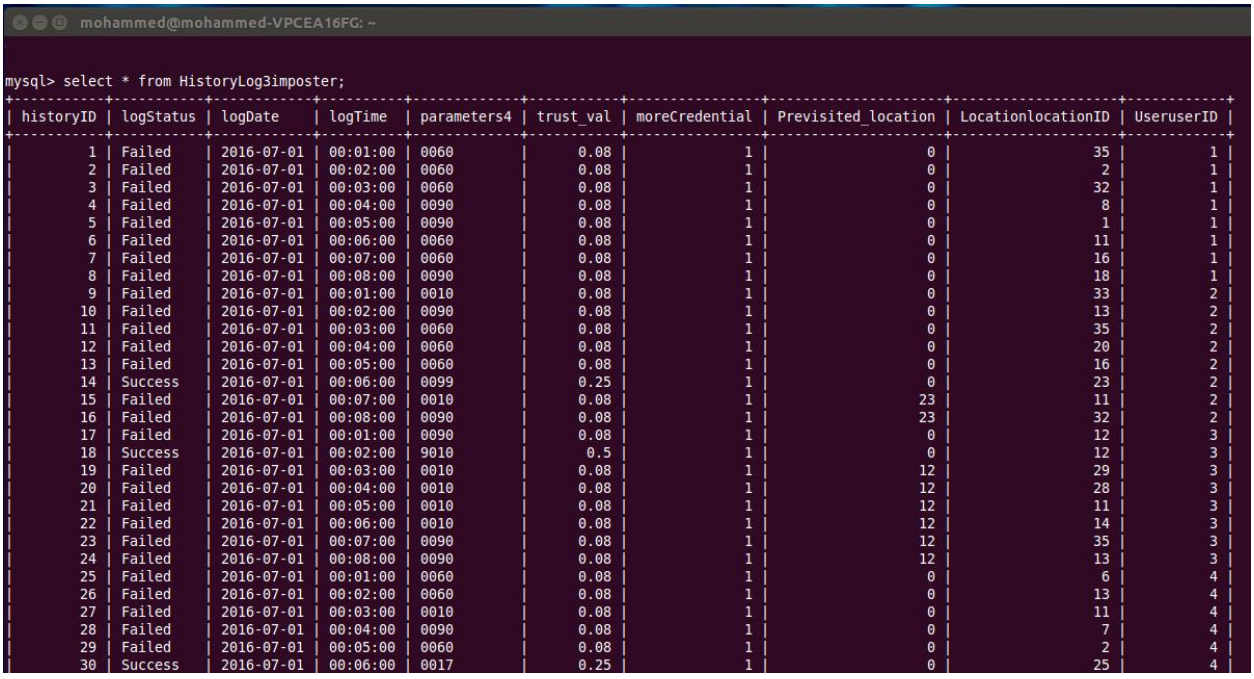
The second phase is to design an environment to mimic the user's activities during the working day. The CAIV simulator is built to achieve this goal. The CAIV simulator assumes that the user has a profile which includes the basic components which are shown above in Figure 5.3; the user is committed to attend any noted event on their calendar; the user should move to a location when they have free time on their calendar that reflects user behaviour in real life. Every single attempt is stored in the history log table. Figure 5.5 below shows a set of stored history logs; the trust_val column represents the calculated value of trust of each attempt and the logStatus column describes

whether the attempt succeeded or failed. The parameters4 column contains the input values (the rate of each parameter of an event) aggregated in one value; for example, let us take the first tuple with historyID (1); parameters4 has 2090 value; it means that the parameter weights of the user at this activity are 0.2 for calendar, 0.0 for history and 0.9 for ambient objects and 0.0 for user preferences. It can be used later to validate the fuzzy inference rules or to understand the behaviour of the system. It is mainly stored to help an expert or the administrator to maintain (rectify) system rules in the future.

The fuzzy rules system is designed and run by Matlab as shown in Figure 4.7. A short program was written to run the whole possible input combination on Matlab and store the whole tuples with the trust value in the rules_heap array; below is a sample of this array:

```
ruls_heap=[[0,0,0,0,0.08],
           [0,0,0,0.1,0.159885057],
           [0,0,0,0.2,0.203076923],
           [0,0,0,0.3,0.212989045],
           [0,0,0,0.4,0.20729443],
           [0,0,0,0.5,0.204974811],
           [0,0,0,0.6,0.20729443],
           [0,0,0,0.7,0.212989045],
           [0,0,0,0.8,0.223406593],
           [0,0,0,0.9,0.235208333],
```

The first four digits are the combinations of inputs, while the bolded digit on the right is the trust value (the result of the FIS).



```
mohammed@mohammed-VPCEA16FG: ~

mysql> select * from HistoryLog3imposter;
+-----------+-----------+------------+----------+------------+-----------+---------------+--------------------+--------------------+------------+
| historyID | logStatus | logDate    | logTime  | parameters4 | trust_val | moreCredential | Previsited_location | LocationlocationID | UseruserID |
+-----------+-----------+------------+----------+------------+-----------+---------------+--------------------+--------------------+------------+
|         1 | Failed    | 2016-07-01 | 00:01:00 | 0060       |      0.08 |             1 |                  0 |                 35 |          1 |
|         2 | Failed    | 2016-07-01 | 00:02:00 | 0060       |      0.08 |             1 |                  0 |                  2 |          1 |
|         3 | Failed    | 2016-07-01 | 00:03:00 | 0060       |      0.08 |             1 |                  0 |                 32 |          1 |
|         4 | Failed    | 2016-07-01 | 00:04:00 | 0090       |      0.08 |             1 |                  0 |                  8 |          1 |
|         5 | Failed    | 2016-07-01 | 00:05:00 | 0090       |      0.08 |             1 |                  0 |                  1 |          1 |
|         6 | Failed    | 2016-07-01 | 00:06:00 | 0060       |      0.08 |             1 |                  0 |                 11 |          1 |
|         7 | Failed    | 2016-07-01 | 00:07:00 | 0060       |      0.08 |             1 |                  0 |                 16 |          1 |
|         8 | Failed    | 2016-07-01 | 00:08:00 | 0090       |      0.08 |             1 |                  0 |                 18 |          1 |
|         9 | Failed    | 2016-07-01 | 00:01:00 | 0010       |      0.08 |             1 |                  0 |                 33 |          2 |
|        10 | Failed    | 2016-07-01 | 00:02:00 | 0090       |      0.08 |             1 |                  0 |                 13 |          2 |
|        11 | Failed    | 2016-07-01 | 00:03:00 | 0060       |      0.08 |             1 |                  0 |                 35 |          2 |
|        12 | Failed    | 2016-07-01 | 00:04:00 | 0060       |      0.08 |             1 |                  0 |                 20 |          2 |
|        13 | Failed    | 2016-07-01 | 00:05:00 | 0060       |      0.08 |             1 |                  0 |                 16 |          2 |
|        14 | Success   | 2016-07-01 | 00:06:00 | 0099       |      0.25 |             1 |                  0 |                 23 |          2 |
|        15 | Failed    | 2016-07-01 | 00:07:00 | 0010       |      0.08 |             1 |                 23 |                 11 |          2 |
|        16 | Failed    | 2016-07-01 | 00:08:00 | 0090       |      0.08 |             1 |                 23 |                 32 |          2 |
|        17 | Failed    | 2016-07-01 | 00:01:00 | 0090       |      0.08 |             1 |                  0 |                 12 |          3 |
|        18 | Success   | 2016-07-01 | 00:02:00 | 9010       |       0.5 |             1 |                  0 |                 12 |          3 |
|        19 | Failed    | 2016-07-01 | 00:03:00 | 0010       |      0.08 |             1 |                 12 |                 29 |          3 |
|        20 | Failed    | 2016-07-01 | 00:04:00 | 0010       |      0.08 |             1 |                 12 |                 28 |          3 |
|        21 | Failed    | 2016-07-01 | 00:05:00 | 0010       |      0.08 |             1 |                 12 |                 11 |          3 |
|        22 | Failed    | 2016-07-01 | 00:06:00 | 0010       |      0.08 |             1 |                 12 |                 14 |          3 |
|        23 | Failed    | 2016-07-01 | 00:07:00 | 0090       |      0.08 |             1 |                 12 |                 35 |          3 |
|        24 | Failed    | 2016-07-01 | 00:08:00 | 0090       |      0.08 |             1 |                 12 |                 13 |          3 |
|        25 | Failed    | 2016-07-01 | 00:01:00 | 0060       |      0.08 |             1 |                  0 |                  6 |          4 |
|        26 | Failed    | 2016-07-01 | 00:02:00 | 0060       |      0.08 |             1 |                  0 |                 13 |          4 |
|        27 | Failed    | 2016-07-01 | 00:03:00 | 0010       |      0.08 |             1 |                  0 |                 11 |          4 |
|        28 | Failed    | 2016-07-01 | 00:04:00 | 0090       |      0.08 |             1 |                  0 |                  7 |          4 |
|        29 | Failed    | 2016-07-01 | 00:05:00 | 0060       |      0.08 |             1 |                  0 |                  2 |          4 |
|        30 | Success   | 2016-07-01 | 00:06:00 | 0017       |      0.25 |             1 |                  0 |                 25 |          4 |
```

Figure 5.5 : Snapshot of History Log Table in the CAIV simulator database

5.6 CAIV Prototype

The aim of building a CAIV prototype is to test and validate the CAIV method over certain scenarios and to prove the soundness, deployability, reliability and cost-effectiveness of the CAIV solution. The CAIV prototype was developed by using two principal components: the hardware component and the software component. The hardware is chosen based on performance and availability. Raspberry Pi3 as a main computing unit**.** There is a set of peripherals that are connected to the main computing unit which are the RFID module 13.56mhz Mfrc-rc522, a 4x4 keypad, an OPEN-SMART I2C / IIC LCD 1602 Display Module, a PIR motion sensor and a BLE dongle and Wi-Fi dongle as shown in Figure 5.6. The software is represented by choosing the

operating system platform with the RASPBIAN 4.2, Python V3.0 is the programming language and MySQL is the DBMS.

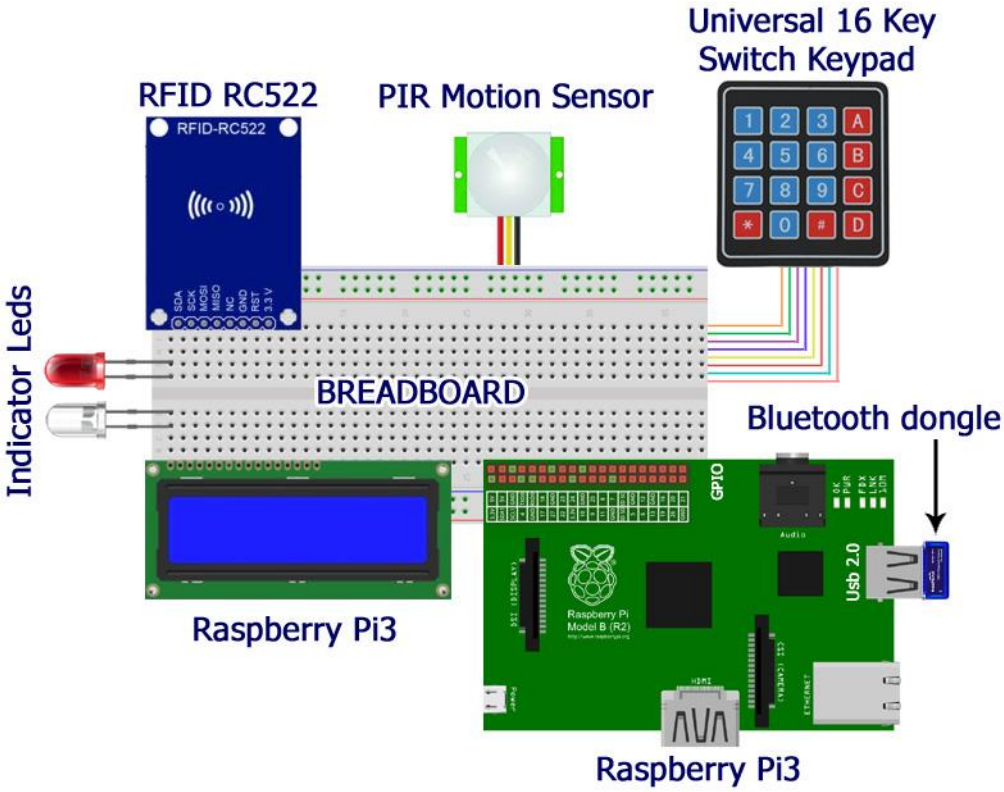


Figure 5.6: CAIV Terminal Hardware Components

(a) LCD shows message of unauthorised user (b)LCD shows message asks of user's passcode

Figure 5.7 : Basic 16x2 Character LCD (a,b)

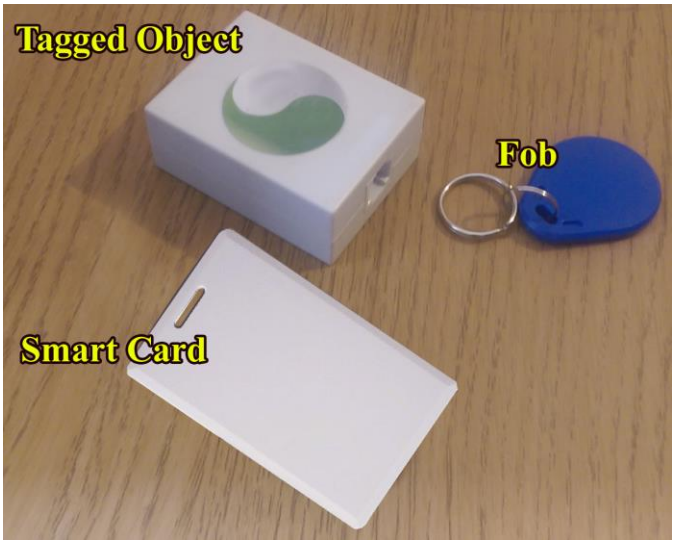


Figure 5.8: RFID Tags

The seven hardware components are assembled via a breadboard which is used to build in each terminal. Each component has a certain task to do, as shown in Table 5-1. Figure 5.9 shows the boxed CAIV terminal with the attached peripherals and two LED indicators.

Table 5-1: CAIV Hardware Components Tasks

ID	Component Name	Task(s)
1.	Raspberry PI 3	It is the main processing unit
2.	LCD	Output unit to display the system messages to the user
3.	Keypad	Input unit to key in the passcode
4.	Wi-Fi	To provide a hotspot to the user to get access to his/her profile
5.	BLE (Bluetooth Low Energy)	It used to discover any nearby devices (ambient objects)
6.	PIR motion sensor	To trigger a BLE scan to avoid any delay for the user
7.	RFID RC522	To read a user's RFID card to identify him/her

Figure 5.9: CAIV Terminal

The CAIV solution has two deployment options, centeralised and decenteralised. The centeralised version deploys a client-server approach as shown in Figure **5.10** while the decenteralised approach uses the local database rather than accessing a remote database server, as shown in Figure **5.11**. The hardware platform can handle both of these solutions. Choosing one of them depends on the environment infrastructure and user's requirements.

The software components are built on a Raspbian operating system (Debian Project, 2017). Python is used as the programming language and a MySQL database is used as a DBMS. Using an RFID reader and other physical sensors moved us forward to include some essential packages to communicate with these prephirals, such as mysql.connector, bluetooth, MFRC522, and RPi.GPIO as shown in Appendix B, the sample python code of CAIV terminal. The terminal provides a wifi hotspot to the users. The user can access the CAIV web-based portal to maintain his/her profile. First, the user needs to login through the homepage as shown in Figure 5.12, then they can modify their calendar as shown in Figure 5.13.

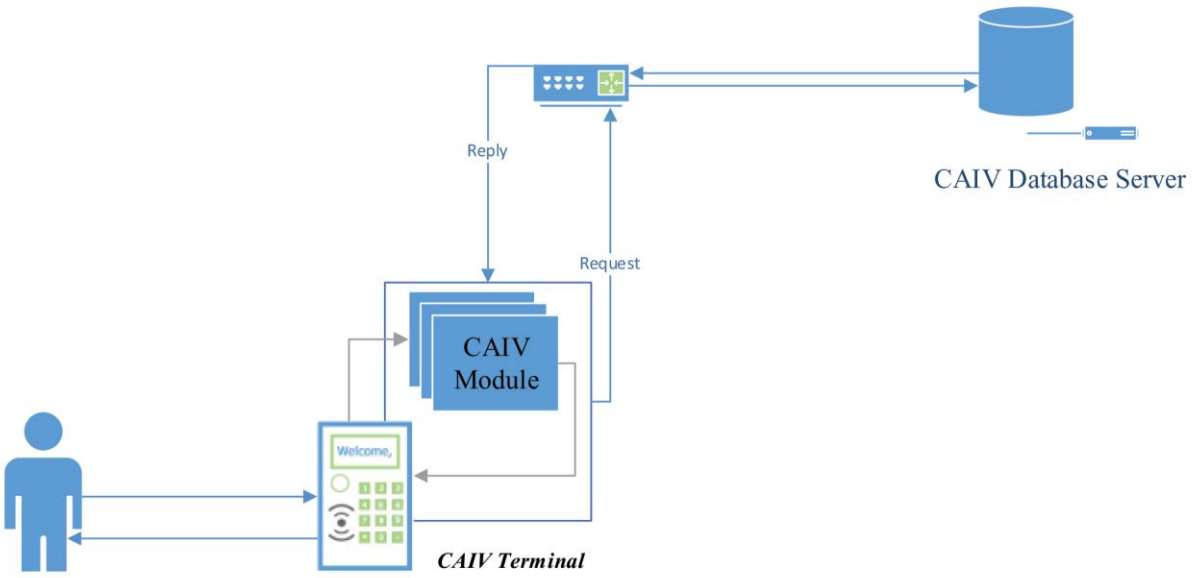


Figure 5.10 : CAIV Terminal (Centralised Approach)

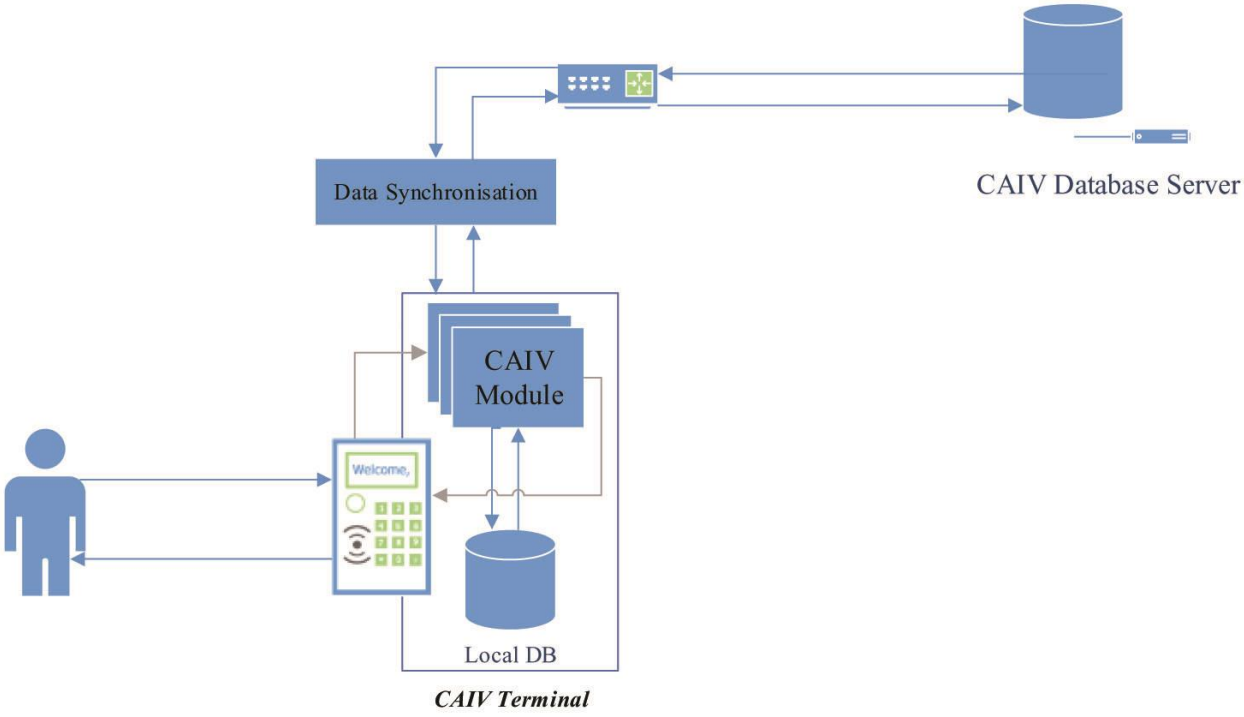


Figure 5.11: CAIV Terminal (Decentralised Approach)

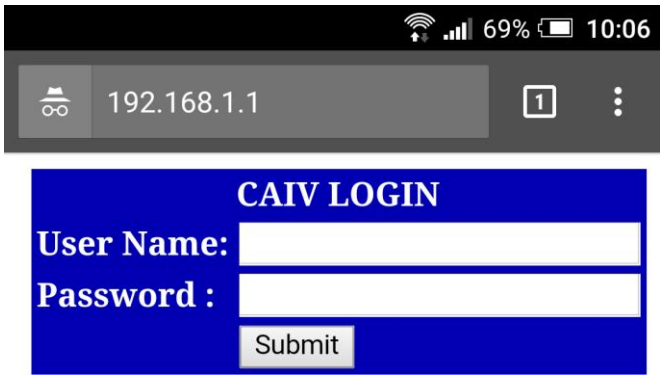


Figure 5.12: Profile Login Page

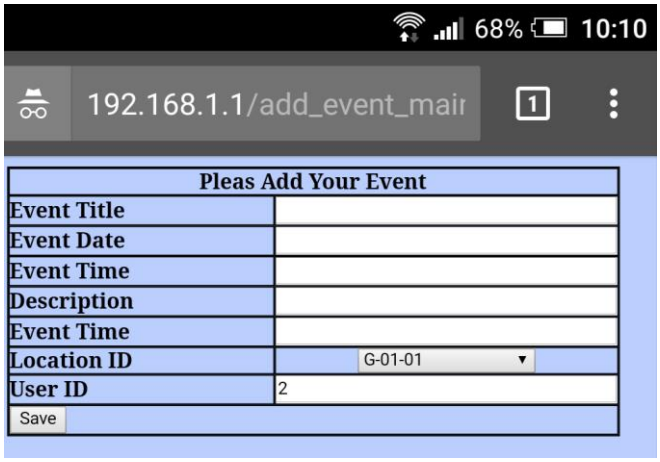


Figure 5.13: Add/Modify New Event Page

5.7 Summary

This chapter illustrated the test data structure, the simulator design and the prototype implementation. It was very challenging to prepare a virtual environment to test and implement the proposed method. Due to the time limitation, Python was chosen as a programming language. It hastens the process of evaluation since the required libraries to perform the method already existed. Additionally, it is available on Raspberry Pi so the transformation process of the method from simulator to prototype is achieved seamlessly. The scenarios helped us to brainstorm some possible cases that the system faces in the real environment. After this, the simulator result gave us a motive to proceed with building a pilot prototype of the virtual environment which proves the feasibility of the proposed method.

Chapter 6

Critical Evaluation

6.1 Overview

CAIV evaluation requires performing two main experiments. The first experiment is to find out if the system is resistant to any failure in identifying legitimate users, the second experiment is to test it against any counterfeit user attempts. The CAIV approach was tested in a simulated environment by using a CAIV simulator which based on CAIV method. The CAIV method aims to verify a user's identity in an unobtrusive way by relying on available context. The user uses only their RFID card to tell the system about the identity that they claim. The context parameters are aggregated from the pervasive environment. If the system (which adopts the CAIV method) fails to verify a user's identity, the system either denies the user's claim or asks them for more credentials as shown in Figure **5**.**7**-b when the system asks the user to enter the passcode to prove their identity.

Synthetic data have been generated by the dataset generator and stored in the database. The stored data have been processed by the CAIV simulator to obtain the CAIV method results. The simulator implementation is performed as described in Chapter 5. The following sections present the details of the synthetic data, the threshold selection and the result.

6.2 Synthetic Data

After the CAIV dataset generator generated the synthetic data, the CAIV simulator was run based on these data. It is an essential part of the evaluation process to observe the CAIV behaviour for two scenarios, for legitimate users and illegitimate users. The first observation obtained from the CAIV simulator assumed that the 79 users were attending whole day events without any intruder breach possibility; Figure 6.1 shows the overall distribution of 79 legitimate users who are moving between the 35 locations in the CAIV environment. The second scenario assumes that the intruder stole (cloned) the RFID card of the user to impersonate the user's identity and a breach test was performed against all 79 users. This attack was run randomly over the 35 locations; Figure 6.2 shows the distribution of 79 illegitimate users visits to the 35 locations within one working day (one day attack). Furthermore, samples of the results of both experiments are shown in Appendix C and Appendix D.

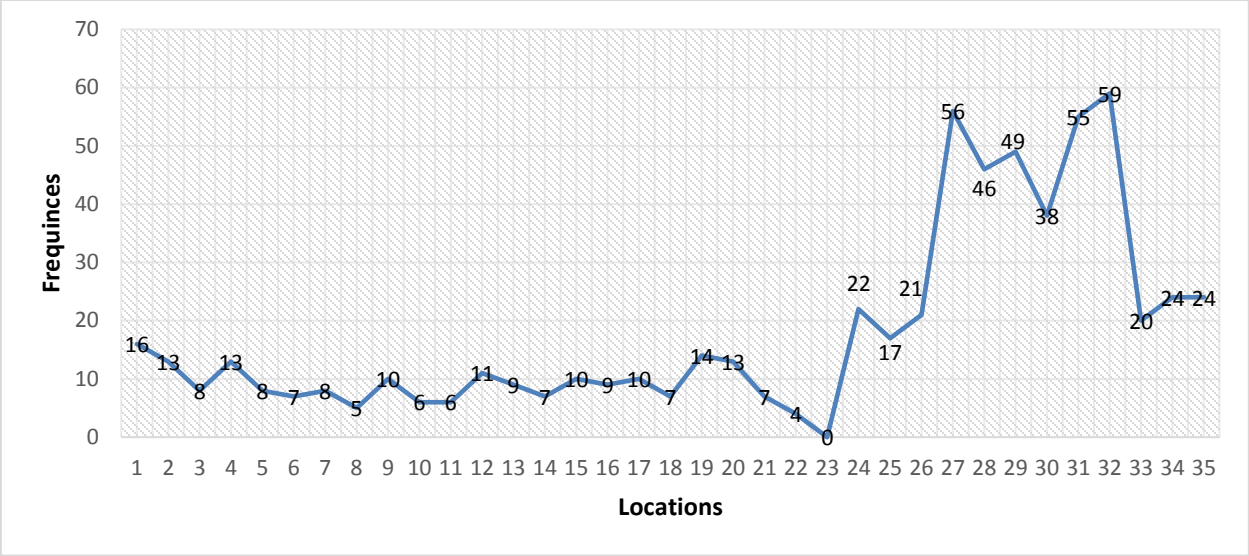


Figure 6.1: Destination Location Frequencies of CAIV Dataset of Legitimate users

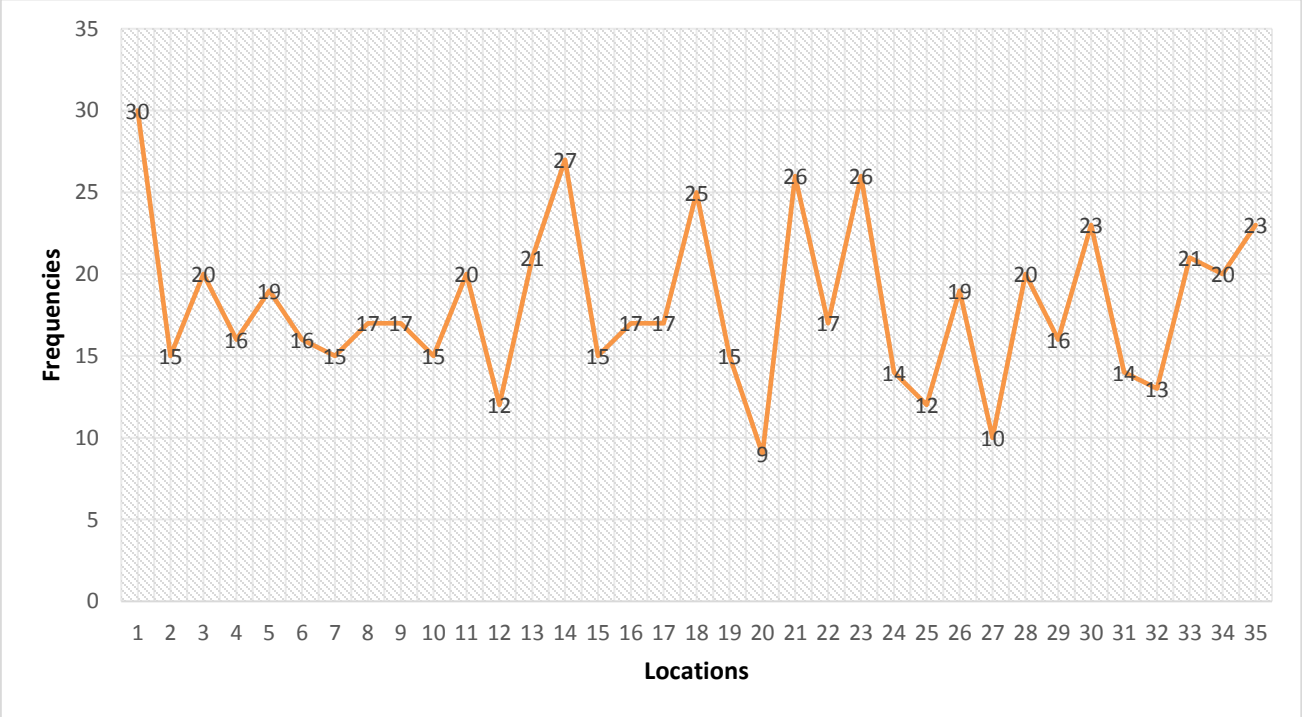


Figure 6.2: Destination Location Frequencies of CAIV Dataset of Illegitimate Users

6.3 Threshold Selection

Threshold selection is a crucial factor in any system. In order to find the optimum threshold value which is trade of False Negative Rate (FNR) and False Positive Rate (FPR). It is required to find the FNR, FPR, sensitivity, specificity and accuracy of each threshold to decide the proper one, which calculated these rates based on the two experiments. The next sections are illustrating the process of choosing the threshold value.

6.3.1 Experiment 1

The first experiment was run on the CAIV simulator for a legitimate user. The legitimate user experiment runs over the daily calendar events of a user with random location access for free time events and a commitment to attend any available calendar event of users; it is assumed that all the activities are coming from a legitimate user. The result data were stored in the local database of the machine. We examined the stored data of the experiment with different thresholds to find the False Negatives (FN) and True Positives (TP). The results were stored in tuples in the database, as shown in Figure 6.3 below; the logStatus field in the experiment database result represented whether the system was successful in verifying the user's identity or not. In this experiment, the failed status is regarded as a FN and a success status is regarded as a TP (De Luca *et al.*, 2012). Appendix E showing the source code of the legitimate user experiment.

```
mysql> select * from HistoryLog3;
+-----------+-----------+-----------+---------+------------+----------+--------------+--------------------+-------------------+-----------+
| historyID | logStatus | logDate   | logTime | parameters4 | trust_val | moreCredential | Previsited_location | LocationlocationID | UseruserID |
+-----------+-----------+-----------+---------+------------+----------+--------------+--------------------+-------------------+-----------+
|         1 | Success   | 2016-07-01 | 00:01:00 | 2090       |      0.25 |            1 |                  0 |                 0 |          1 |
|         2 | Success   | 2016-07-01 | 00:02:00 | 0090       |      0.25 |            1 |                  0 |                32 |          1 |
|         3 | Success   | 2016-07-01 | 00:03:00 | 0090       |      0.25 |            1 |                 32 |                31 |          1 |
|         4 | Success   | 2016-07-01 | 00:04:00 | 9060       |       0.5 |            1 |                 31 |                30 |          1 |
|         5 | Failed    | 2016-07-01 | 00:05:00 | 0010       | 0.0875525 |            1 |                 30 |                30 |          1 |
|         6 | Success   | 2016-07-01 | 00:06:00 | 0060       |       0.5 |            1 |                 30 |                27 |          1 |
|         7 | Failed    | 2016-07-01 | 00:07:00 | 2010       |  0.102964 |            1 |                 27 |                 5 |          1 |
|         8 | Success   | 2016-07-01 | 00:08:00 | 9060       |       0.5 |            1 |                 27 |                27 |          1 |
|         9 | Success   | 2016-07-01 | 00:01:00 | 0060       |       0.5 |            1 |                  0 |                27 |          2 |
|        10 | Success   | 2016-07-01 | 00:02:00 | 5090       |      0.25 |            1 |                 27 |                 9 |          2 |
|        11 | Success   | 2016-07-01 | 00:03:00 | 9090       |       0.5 |            1 |                  9 |                16 |          2 |
|        12 | Success   | 2016-07-01 | 00:04:00 | 0098       |      0.75 |            1 |                 16 |                34 |          2 |
|        13 | Success   | 2016-07-01 | 00:05:00 | 0090       |      0.25 |            1 |                 34 |                35 |          2 |
|        14 | Success   | 2016-07-01 | 00:06:00 | 9090       |       0.5 |            1 |                 35 |                 0 |          2 |
```

Figure 6.3: History Log Table

The experiment total login attempts were 632 attempts for one working day, since the number of subjects is 79 and there are eight time-slots for each business day. The next experiment has the same number of attempts. These two instances(FN, TP) are used later to find the accuracy, FNR and FPR.

6.3.2 Experiment 2

The second experiment was runs on the CAIV simulator for an illegitimate user to find the True Negatives (TN) and False Positives (TP). An illegitimate user experiment runs over the random locations access for all calendar daily events; it is assumed that all activities are coming from intruders (illegitimate users). The attack scenario assumed that the attacker has a user ID (RFID card which is used as user identifier in the system) and they intend to use it within the system's domain. For every attempt (attack), the simulator assigns a random location from the environment locations (35 locations) to test the system's robustness against the attack.

The experiment results are stored in the local database of the machine. The table structure is similar to the HistoryLog table in Figure 6.3 above but stored in different table in the database. The logStatus field of the experiment database represented whether the system had succeeded in verifying the user's identity or not. In this experiment, the failed status regarded as a True Negative

(TN) and a success status is regarded as a False Positive (FP) (De Luca *et al.*, 2012). These two instances are used later to find the accuracy, FNR and FPR.

6.3.3 Confusion Matrix

Given a classifier and an instance, there are four possible outcomes. If the instance is positive and it is classified as positive, it is counted as a **True Positive (TP)**; if the instance is classified as negative and it is positive, it is counted as a **False Negative(FN)**. If the instance is negative and it is classified as negative, it is counted as a **True Negative(TN)**; if the instance is classified as positive and it is negative, it is counted as a **False Positive(FP)**. The set of instances (the test set) can be represented by a two-by-two confusion matrix (also called a contingency table) as shown below in Table 6-1 (Marsland, 2015).

Table 6-1: Confusion Matrix and Common Performance Metrics

Legitimate User	**Illegitimate User**
True **P**ositives	**F**alse **P**ositives
False **N**egatives	**T**rue **N**egatives
P	**N**

6.3.4 False Negative Rate and False Positive Rate

The two equations below are used to find the FNR and FPR(Marsland, 2015). Equation 6-1 presents the formula to calculate the False Negative Rate (FNR). It divides the total of FN cases

where the CAIV system mistakenly identifies the legitimate user as an intruder over FN and TP. We regard the FNR as the False Rejection Rate (FRR).

$$FNR(FRR) = \frac{FN}{(FN+TP)}.100\% \quad \text{Equation 6-1}$$

Equation 6-2 is the formula that used to find the False Positive Rate. We regards the FPR as the False Acceptance Rate (FAR). It divide the FP where the CAIV system missidentified the intruder as a legitimate user over FP and TN cases.

$$\boldsymbol{FPR(FAR) = \frac{FP}{(FP+TN)}.100\%} \quad \text{Equation 6-2}$$

After implemented the two experiment, the stored data of the experiments results have a list of cases which includes (FNs, TPs, TNs, TPs), which can be used to find the FNR and PPR rates.

6.3.5 Accuracy

Accuracy determines the true value, the repeatability or reproducibility of the measurement and the proximity of measurement to the precision results. Marsland defined it as "the sum of the number of true positives and true negatives divided by the total number of examples".

$$\text{Accuracy} = \frac{TP+TN}{(FN+TN+FP+TP)} \quad \text{Equation 6-3 (Marsland, 2015)}$$

To find the optimum threshold, we need to find FAR, FRR, accuracy, sensitivity and specificity. Therefore, we need four essential types of classifications (TP, FP, TN and FN) to make the calculations for different thresholds. Classifications obtained by running the CAIV simulator based on two scenarios to get two experiments for both legitimate and illegitimate users as described in experiment 1 and experiment 2. To decide the optimum threshold, the experiments need to be

tested against different threshold values. After running the experiment for various thresholds, the resulted data can be used to decide the threshold value. The threshold values are [0.1,0. 12,…,0.98,1.0];

Table 6-2: FRR and FAR Rates of Different Thresholds

Threshold	FRR	FAR	Threshold	FRR	FAR	Threshold	FRR	FAR	Threshold	FRR	FAR
0.1	0.11	0.07	0.34	0.49	0.01	0.58	0.99	0	0.82	1	0
0.12	0.17	0.06	0.36	0.98	0.01	0.6	0.99	0	0.84	1	0
0.14	0.17	0.06	0.38	0.49	0.01	0.62	0.99	0	0.86	1	0
0.16	0.17	0.06	0.4	0.49	0.01	0.64	0.99	0	0.88	1	0
0.18	0.17	0.06	0.42	0.49	0.01	0.66	0.99	0	0.9	1	0
0.2	0.17	0.06	0.44	0.49	0.01	0.68	0.99	0	0.92	1	0
0.22	0.17	0.06	0.46	0.49	0.01	0.7	0.99	0	0.94	1	0
0.24	0.17	0.06	0.48	0.49	0.01	0.72	0.99	0	0.96	1	0
0.26	0.49	0.01	0.5	0.99	0	0.74	0.99	0	0.98	1	0
0.28	0.49	0.01	0.52	0.99	0	0.76	1	0	1	1	0
0.3	0.49	0.01	0.54	0.99	0	0.78	1	0			
0.32	0.49	0.01	0.56	0.99	0	0.8	1	0			

Figure 6.4 shows the presentation of the FRR values of the different thresholds, while Figure **6**.**5** shows the FAR values of the same thresholds. We conclude that the threshold value with 0.5 enhances the FAR percentage to 0% while get a high percentage of FRR with 99%. Increasing the threshold value affects the FAR positively and FRR negatively. However, the threshold value can be decided based on the institution (environment) policy.

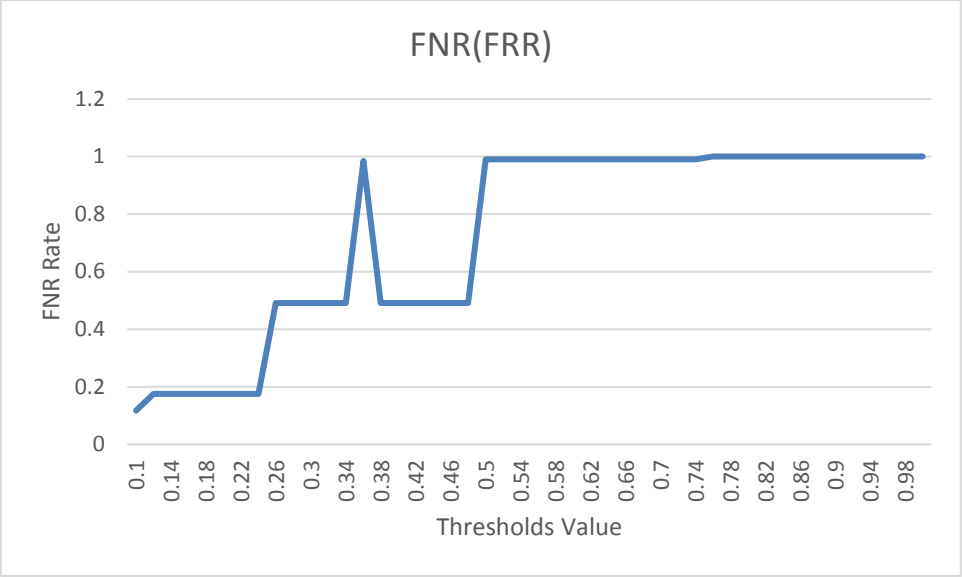


Figure 6.4: FRR Rates of Different Thresholds

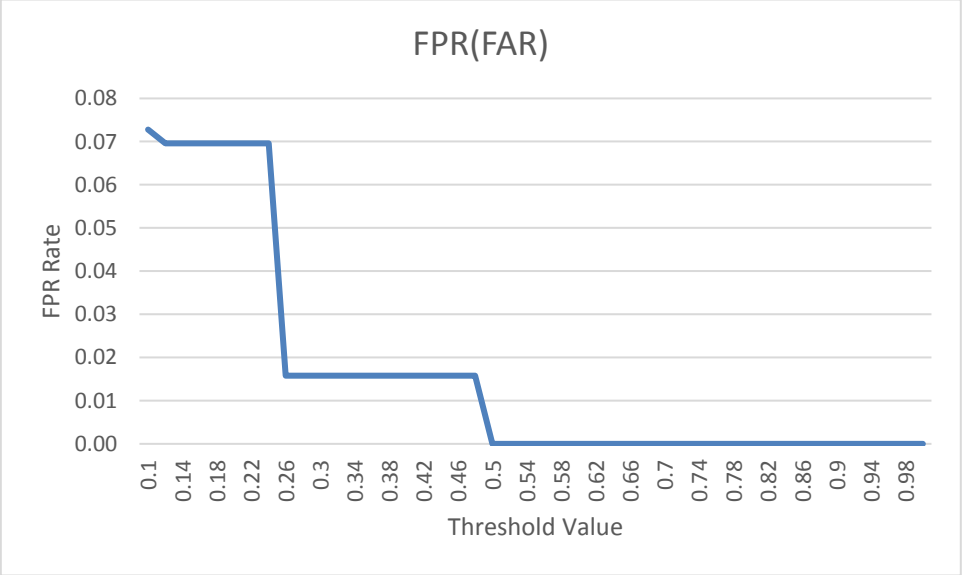


Figure 6.5: FAR Rates of Different Thresholds

To find the sensitivity which is the true positive rate Equation 6-4 is used. And to find the specificity which is the true negative rate Equation 6-5 is used.

SN=TP/P Equation 6-4: Sensitivity (True Positive Rate)

SP=TN/N Equation 6-5: Specificity (True Negative Rate)

The optimum threshold value is 0.24. As shown in Table 6-3, the number of attacks was 632, only 44 of them was FP while the number of FN less than the number of FNs of 0.26 threshold value. From the observed data, the FP cases divided between the Calendar and User preferences while the history log did not record any FP case, Figure 6.6 shows the percentage for each parameter.

$$Accuracy(0.24) = \frac{TP+TN}{(FN+TN+FP+TP)} = 87\%$$

$$Accuracy(0.26) = \frac{TP+TN}{(FN+TN+FP+TP)} = 74\%$$

Table 6-3: The Optimum Threshold Results

Threshold	**TP**	**FN**	**TN**	**FP**	**FRR**	**FAR**	**Accuracy**	**Sensitivity**	**Specificity**
0.26	322	310	622	10	17%	1%	75%	51%	98%
0.24	521	111	588	44	17%	6%	88%	82%	93%

Figure 6.6 showing the percentage of exploiting the intruder for each parameter, when the CAIV misidentified the identity of the user. It shows that the calendar and preferences are sharing the

percentage with 60% of the Calendar and 40% of the Preferences. While the history did not exploit by the intruder with 0%. The Ambient Object excluded since the attack scenario assumed that the intruder has only the RFID card of the user.

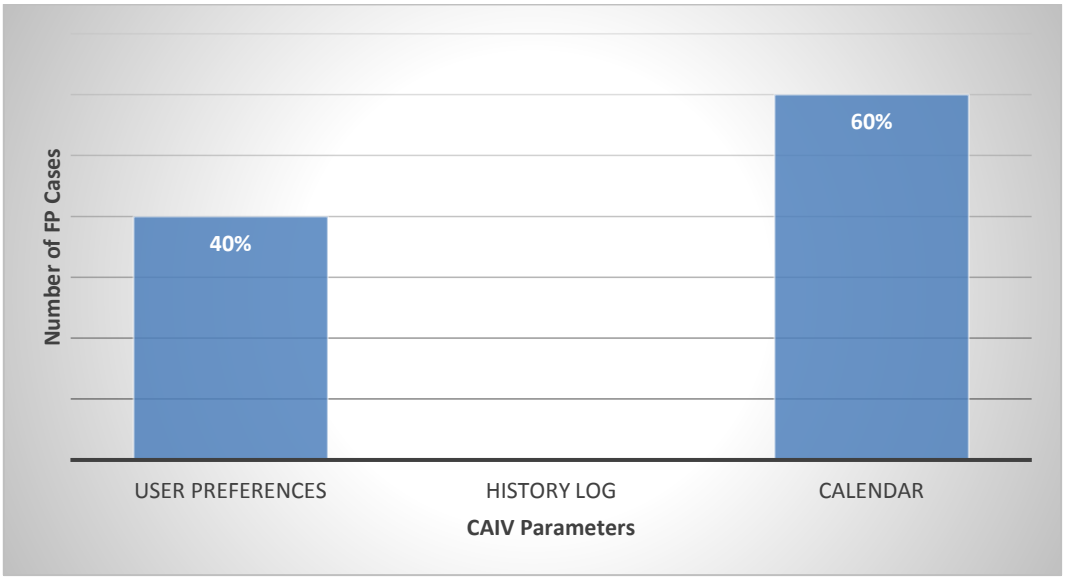


Figure 6.6: False Positive Cases of Illegitimate User Experiment, 0.4 Threshold Value

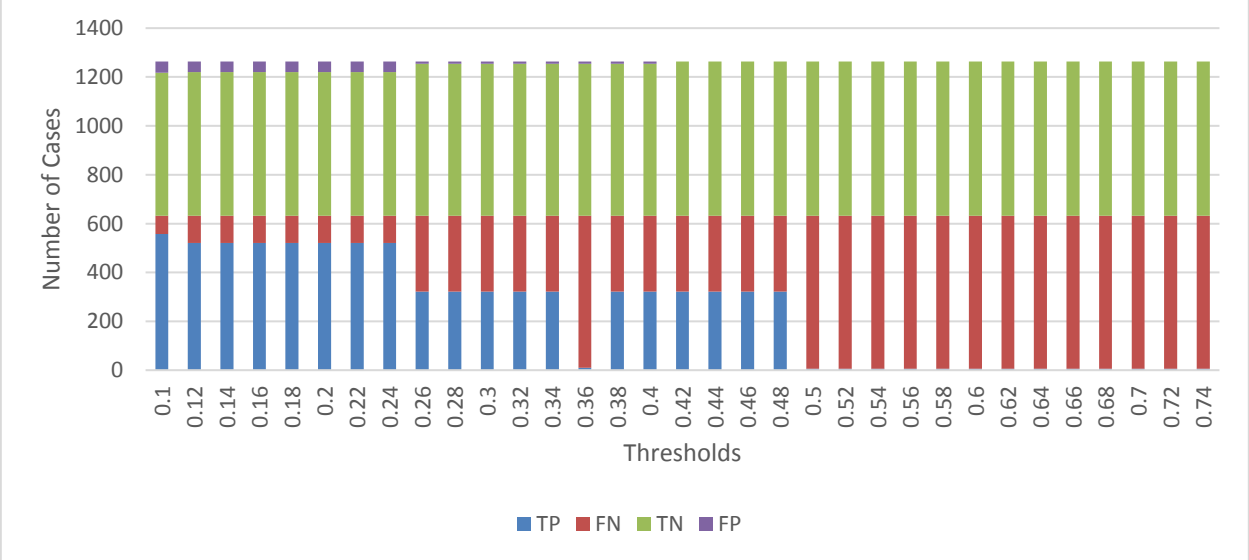


Figure 6.7: TP, FN, TN and FP of Different Thresholds

Figure 6.7 shows that the FP cases are very low at 0.26-0.48 thresholds. While TP dropdown to zero at 0.5 threshold. FNs increased exponentially until 0.5 threshold, after that still at the same level, while TN has a stable performance over the different thresholds.

6.4 Results

After running the experiment with the optimum threshold, 0.24, the result of sensitivity, specificity and accuracy values are as follows: 82%, 93% and 88%, respectively. The findings show that the CAIV model can identify and verify a user's identity in a pervasive environment with a high level of precision. Appendix F showing the results of these three rates of different thresholds.

Moreover, the performance of CAIV can be enhanced in the future by reviewing and refining CAIV inference rules. The maintenance of these rules is achievable because the history log of any misidentified identity is stored in the history log database; the experts can retrieve it for auditing purposes.

Table **6-3** above shows the accuracy percentages of the FRR and FAR of the optimum threshold, 0.24. Since, selecting the threshold value is a trade-off between FAR, FRR and accuracy, the 0.26 threshold value can be selected as well. However, we get less accuracy and sensitivity rate and quite high number of FN, yet the FPs are only 10 cases.

6.5 Summary

CAIV is a context-based model which employs four context parameters to verify user identity in a pervasive environment. The CAIV method can ascertain the identity of the user in this environment based on the available context (different combinations of context parameters). These parameters go through a fuzzy inference system, which includes a set of rules. These rules have

been discussed with experts and presented and refined to become 81 rules. We perform an evaluation test by using two experiments; one for legitimate users and the other for illegitimate users. The results in Table **6-3** above shows that the CAIV method is reliable enough to be considered as a standalone verification method of user identity in the pervasive computing environment. Moreover, due to the CAIV design which is illustrated in Chapters 4 and 5, CAIV can integrate with some information systems in the pervasive environment to facilitate the verification process (no further credentials required, less intrusive) and improve the security of the system while preserving the user's privacy at the same time.

Chapter 7

Conclusion and Future Work

In this research, an unobtrusive method is proposed to verify a user's identity in a pervasive computing environment based on four context parameters. This chapter includes the study's conclusion along with future work discussion, presented in the following sections.

7.1 Conclusion

The pervasive computing environment includes a variety of services, devices and secure locations which are restricted to use by legitimate users. Using pervasive computing environment facilities requires occasional verification of users' identities. Traditional authentication methods (HAK) are quite distracting for the user and need more effort (time, cost) to deploy them. Since these methods were not designed for the pervasive environment, the need has emerged for a new verification method tailored to this environment. This method needs to be less intrusive to the user and capable of verifying a user's identity at the same time. The CAIV method is well placed to solve this issue. It provides a seamless verification service for user identity in the pervasive computing environment which is the main objective of this research.

We presented the most important scenarios to evaluate the proposed method and depict the system's functionalities. Briefly, an RFID card is used to identify the user in the pervasive environment. Consequently, the CAIV module can identify the user based on the available context, the CAIV aggregates the available context and sends these data to the CAIV reasoning engine. The CAIV reasoning engine is a fuzzy rule based model; it processes these data to produce a trust value. The trust value is compared against the threshold value to enable the system to decide whether the user satisfies a certain level of trust. Then the system declares the user as a legitimate user if they reach that level of trust; otherwise, the user is either rejected and declared as an illegitimate user or the system requests more credentials to verify the user's identity.

A fuzzy rule based model is used in the CAIV method to infer a user's identity. It uses a set of rules which were discussed with experts and it was decided to use 81 rules. The fuzzy logic is selected to overcome data shortages, learning delay and cloning the expert experience into the CAIV model. This cloning brings the expertise of the security expert into the CAIV method and makes the system behave like an experienced security person.

The essential principles of the proposed system are to infer a user's identity based on the available context. That context is classified based on the existence of activity. This activity is divided into two parts: location and time. Then, the activity classification is mapped into each parameter of the four CAIV parameters. Eventually, the decision (classification) makes the decision of experts about the parameters' significance easier.

To implement, analyse and test the CAIV system behaviour against different scenarios which face the system and the users every day, we did two experiments by using the CAIV simulator that were

conducted to monitor and collect synthetic data about a potential user's behaviours. Finally, a CAIV prototype was developed to give users the experience of using the CAIV model in real-life.

The two experiments above were conducted in a discrete event simulator (CAIV simulator) which we developed to find a suitable environment for the CAIV method to be tested. The simulation results of those two experiments helped to select the proper threshold value and show the behaviour of the CAIV method with various scenarios.

The CAIV prototype indicates that our approach is achievable and deployable. It provides evidence of how the CAIV terminal will perform in a real system environment. It gives a chance to the user to use the system not only by relying on simulator results but to try it in a real environment.

The threshold value has been chosen based on the balance between the FAR and FRR. The selected value can be changed according to the policy in the deployed environment (organisation, university, institute, etc).

Results show that the new method tackles the issues faced by traditional verification methods and previous work. The result shows that the CAIV method successfully achieved the research aim and objectives. CAIV nominated four context parameters of pervasive computing to verify user’s identity by relying on the literature review. We developed the framework based on these parameters, tests and evaluates that framework by using the designated simulator. It overcome the learning process time of other approaches, such as machine learning by relying on fuzzy logic; the fuzzy logic rules is maitainable and easy to understand by expert. The results shows that the CAIV method is achieved 88% of accuracy of verifiying user identity in the pervasive computing envrinment.

7.2 Future work

The CAIV method maintains four parameters. The main goal of the research is to prove the concept thoroughly. This concept claims the ability to use multi-context parameters to verify a user's identity. Consequently, there are some assumptions and limitations that can overcome them in the future such as extending the history parameter dependency from location only to time and location together. Other extensions are expanding the ambient object weight to include multi-objects rather than picking up an object with the highest weight and deploying the CAIV prototype in a real environment, by getting real data that will help to maintain the CAIV design to enhance the CAIV performance. Finally, from the real data, we can calculate the weight of an event's components (time and location) for each parameter.

PUBLICATIONS

During the past three years, the following publication have achieved:

1- Mohammed Al-Jawad, Adil Al-Yasiri. Context-Aware Framework for Inferring User Identity in Pervasive Computing Environment. University of Salford, Research Showcase, College of Science and Technology, Jun-2014

2- Mohammed Al-Jawad, Adil Al-Yasiri. A Context-Aware Method for Verifying User Identity in Pervasive Computing Environments. University of Salford, Annual PGR Symposium, March-2017

REFERENCES

Abowd, D., Dey, A. K., Orr, R. and Brotherton, J. (1998) 'Context-awareness in wearable and ubiquitous computing', *Virtual Reality*. Springer, 3(3), pp. 200–211.

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M. and Steggles, P. (1999) 'Towards a better understanding of context and context-awareness', in *Handheld and ubiquitous computing*. Springer, pp. 304–307.

Akhtar, Z., Micheloni, C. and Foresti, G. L. (2015) 'Biometric liveness detection: challenges and research opportunities', *IEEE Security & Privacy*. IEEE, 13(5), pp. 63–72.

Al-Karkhi, A., Al-Yasiri, A. and Linge, N. (2015) 'Discreet verification of user identity in pervasive computing environments using a non-intrusive technique', *Computers & Electrical Engineering*, 41, pp. 102–114. doi: 10.1016/j.compeleceng.2014.10.006.

Alexander, I. F. and Beus-Dukic, L. (2009) *Discovering requirements: how to specify products and services*. John Wiley & Sons.

Alexander, I. F. and Maiden, N. (2005) *Scenarios, stories, use cases: through the systems development life-cycle*. John Wiley & Sons.

Baldauf, M. (2007) 'A survey on context-aware systems Schahram Dustdar * and Florian Rosenberg', 2(4).

Behrends, E. (2000) *Introduction to Markov chains*. Springer.

Bělohlávek, R. and Klir, G. J. (2011) *Concepts and fuzzy logic*. MIT Press.

Bhatnagar, R. and Kanal, L. N. (1992) *Models of enquiry and formalisms for approximate reasoning*. John Wiley & Sons, Inc.

Bishop, M. A. (2002) ‘The art and science of computer security’. Addison-Wesley Longman Publishing Co., Inc.

Brainard, J., Juels, A., Rivest, R. L., Szydlo, M. and Yung, M. (2006) ‘Fourth-factor authentication: somebody you know’, in *ACM conference on computer and communications security*, pp. 168–178.

Brown, M. G. (1996) ‘Supporting user mobility’, in *Mobile Communications*. Springer, pp. 69–77.

Brown, P. J. (1995) ‘The stick-e document: a framework for creating context-aware applications’, *ELECTRONIC PUBLISHING-CHICHESTER-*. Citeseer, 8, pp. 259–272.

Brown, P. J., Bovey, J. D. and Chen, X. (1997) ‘Context-aware applications: from the laboratory to the marketplace’, *Personal Communications, IEEE*. IEEE, 4(5), pp. 58–64.

Van Bunningen, A. H., Feng, L. and Apers, P. M. G. (2005) ‘Context for ubiquitous data management’, in *Ubiquitous Data Management, 2005. UDM 2005. International Workshop on*, pp. 17–24.

Campbell, R., Al-Muhtadi, J., Naldurg, P., Sampemane, G. and Mickunas, M. D. (2003) ‘Towards Security and Privacy for Pervasive Computing’, in *Proceedings of the 2002 Mext-NSF-JSPS International Conference on Software Security: Theories and Systems*. Berlin, Heidelberg: Springer-Verlag (ISSS’02), pp. 1–15. Available at: http://dl.acm.org/citation.cfm?id=1765533.1765535.

Carroll, J. M. (2000) *Making use: scenario-based design of human-computer interactions*. MIT press.

Chalmers, D. (2011) *Sensing and systems in pervasive computing: Engineering context aware systems*. Springer.

Chen, G. and Kotz, D. (2000) *A survey of context-aware mobile computing research*. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.

Chen, H. (2004) *An Intelligent Broker Architecture For Pervasive Context-Aware Systems*. University of Maryland.

Chisholm, R. M. (1997) ‘Identity Through Time’, in Rea, M. C. (ed.) *Material Constitution*. Rowman & Littlefield, p. 209.

Choi, S. and Zage, D. (2012) ‘Addressing insider threat using “where you are” as fourth factor authentication’, in *Security Technology (ICCST), 2012 IEEE International Carnahan Conference on*. IEEE, pp. 147–153.

Choi, Y., Doh, I., Park, S.-S. and Chae, K.-J. (2013) ‘Security based semantic context awareness system for M2M ubiquitous healthcare service’, in *Ubiquitous Information Technologies and Applications*. Springer, pp. 187–196.

Chun-Li, L. I. N., Hung-Min, S. U. N. and Hwang, T. (2001) ‘Attacks and solutions on strong-password authentication’, *IEICE transactions on communications*. The Institute of Electronics, Information and Communication Engineers, 84(9), pp. 2622–2627.

Commission, S. S. (2006) ‘Guidance on Multi-factor Authentication’. Available at: http://ict.govt.nz/assets/Uploads/Documents/egif-authentication-multi-factor-guidance-june-2006.pdf.

Conti, M., Das, S. K., Bisdikian, C., Kumar, M., Ni, L. M., Passarella, A., Roussos, G., Tröster, G., Tsudik, G. and Zambonelli, F. (2012) ‘Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber--physical convergence’, *Pervasive and Mobile Computing*. Elsevier, 8(1), pp. 2–21.

Covington, M. J., Fogla, P., Zhan, Z. and Ahamad, M. (2002) ‘A context-aware security architecture for emerging applications’, in *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, pp. 249–258.

Craig, I. and Whitty, M. (2017) ‘Region Formation for Efficient Offline Location Prediction’, *IEEE Pervasive Computing*. IEEE, 16(1), pp. 66–73.

Cranor, L. F. and Garfinkel, S. (2008) *Security and Usability: Designing Secure Systems that People Can Use*. O'Reilly Media. Available at: http://books.google.co.uk/books?id=wDVhy9EyEAEC.

Cristofaro, E. De (2014) 'A Comparative Usability Study of Two-Factor Authentication'.

Dawson, F., Dawson, F., Stenerson, D. and Stenerson, D. (1998) *RFC 2445 - Internet Calendaring and Scheduling Core Object Specification (iCalendar)*, *IETF Request For Comments*.

Debian Project (2017) *Raspbian*, *Debian Project*.

Deng, F. M. (2011) *War of visions: Conflict of identities in the Sudan*. Brookings Institution Press.

Dey, A. K. (1998) 'Context-aware computing: The CyberDesk project', in *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, pp. 51–54.

Dourish, P. (2004) 'What we talk about when we talk about context', *Personal and Ubiquitous Computing*, 8(1), pp. 19–30. doi: 10.1007/s00779-003-0253-8.

Emmanouilidis, C., Koutsiamanis, R.-A. and Tasidou, A. (2013) 'Mobile guides: Taxonomy of architectures, context awareness, technologies and applications', *Journal of Network and Computer Applications*. Elsevier, 36(1), pp. 103–125. doi: 10.1016/j.jnca.2012.04.007.

Fearon, J. D. (1999) 'What is identity (as we now use the word)', *Unpublished manuscript, Stanford University, Stanford, Calif*.

Fickas, S., Kortuem, G. and Segall, Z. (1997) 'Software organization for dynamic and adaptable wearable systems', in *Wearable Computers, 1997. Digest of Papers., First International Symposium on*. IEEE, pp. 56–63.

Gray, A. R. and MacDonell, S. G. (1997) 'A comparison of techniques for developing predictive models of software metrics', *Information and software technology*. Elsevier, 39(6), pp. 425–437.

Greenfield, A. (2010) *Everyware: The dawning age of ubiquitous computing*. New Riders.

Hayashi, E., Hong, J., Das, S., Amini, S. and Oakley, I. (2013) 'CASA : Context - Aware Scalable Authentication', pp. 1–10.

Henricksen, K. (2003) *A framework for context-aware pervasive computing applications*. University of Queensland Queensland.

Hong, J., Suh, E. and Kim, S.-J. (2009) 'Context-aware systems: A literature review and classification', *Expert Systems with Applications*. Elsevier Ltd, 36(4), pp. 8509–8522. doi: 10.1016/j.eswa.2008.10.071.

Hong, T.-P. and Lee, C.-Y. (1996) 'Induction of fuzzy rules and membership functions from training examples', *Fuzzy sets and Systems*. Elsevier, 84(1), pp. 33–47.

Hughes, J. and Maler, E. (2005) 'Security Assertion Markup Language (SAML) V2. 0 Technical Overview', *OASIS SSTC Working Draft sstc-saml-tech-overview-2.0-draft-08*, pp. 29–38.

Hull, R., Neaves, P. and Bedford-Roberts, J. (1997) 'Towards situated computing', in *Wearable Computers, 1997. Digest of Papers., First International Symposium on*. IEEE, pp. 146–153.

Huynh, T. D., Jennings, N. R. and Shadbolt, N. R. (2006) 'An integrated trust and reputation model for open multi-agent systems', *Autonomous Agents and Multi-Agent Systems*. Springer, 13(2), pp. 119–154.

IL, Barry L. Nelson, Jerry Banks, J. S. C. (2009) *Discrete-Event System Simulation*. 5th edn. United States: Pearson Education (US).

Indulska, J. and Sutton, P. (2003) 'Location management in pervasive systems', in *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21*, pp. 143–151.

ITU-T (2009) *NGN identity management framework*, *Recommendation ITU-T Y.2720*. Available at: https://www.itu.int/rec/T-REC-Y.2720-200901-I.

Jang, B., Park, S., Chang, H., Ahn, H. and Choi, E. (2010) 'A study of context-awareness RBAC model using user profile on ubiquitous computing', *Communications in Computer and Information Science*, 120

CCIS, pp. 205–213. doi: 10.1007/978-3-642-17604-3_23.

Jaroucheh, Z., Liu, X. and Smith, S. (2010) 'CANDEL: Product Line Based Dynamic Context Management for Pervasive Applications', *2010 International Conference on Complex, Intelligent and Software Intensive Systems*. Ieee, pp. 209–216. doi: 10.1109/CISIS.2010.21.

Jia, X., Cao, H., Tian, J. and Liu, Y. (2014) 'CONTEXT-AWARE ADAPTIVE AUTHENTICATION METHOD AND APPARATUS'.

Jsang, A. and Ismail, R. (2002) 'The beta reputation system', in *Proceedings of the 15th bled electronic commerce conference*, pp. 2502–2511.

Kim, I. and Lee, Y. (2006) 'CAST: Design and implementation of secure context-awareness simulation toolkit for ubiquitous computing research environment', *Lecture Notes in Control and Information Sciences*, 344, pp. 1040–1055. doi: 10.1007/11816492_135.

Kortuem, G., Segall, Z. and Bauer, M. (1998) 'Context-aware, adaptive wearable computers as remote interfaces to'intelligent'environments', in *1998 16th International Symposium on Wearable Computers*. IEEE Computer Society, p. 58.

Kothari, C. R. (2004) *Research methodology: Methods and techniques*. New Age International.

Krumm, J. (2009) *Ubiquitous computing fundamentals*. CRC Press.

Laliwala, Z. and Chaudhary, S. (2008) 'Event-driven service-oriented architecture', in *Service Systems and Service Management, 2008 International Conference on*, pp. 1–6.

Van Leekwijck, W. and Kerre, E. E. (1999) 'Defuzzification: criteria and classification', *Fuzzy sets and systems*. Elsevier, 108(2), pp. 159–178.

Lesani, M. and Montazeri, N. (2009) 'Fuzzy trust aggregation and personalized trust inference in virtual social networks', *Computational Intelligence*. Wiley Online Library, 25(2), pp. 51–83.

Li, X., Martínez, J.-F. and Rubio, G. (2017) 'Towards a Hybrid Approach to Context Reasoning for Underwater Robots', *Applied Sciences*. Multidisciplinary Digital Publishing Institute, 7(2), p. 183.

Lin, Z.-H. and Fu, L.-C. (2007) 'Multi-user preference model and service provision in a smart home environment', in *Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on*, pp. 759–764.

Lo, H. Z. (2014) *Discrete Event Simulation*, *University of Massachusetts Boston*. Available at: http://www.cs.umb.edu/~henryzlo/cs310/notes/simulation.pdf.

De Luca, A., Hang, A., Brudy, F., Lindner, C. and Hussmann, H. (2012) 'Touch me once and i know it's you!: implicit authentication based on touch screen patterns', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 987–996.

Luger, G. F. (2005) *Artificial intelligence: structures and strategies for complex problem solving*. Pearson education.

Malek, B., Miri, A. and Karmouch, A. (2008) 'A framework for context-aware authentication'. IET.

Mamdani, E. H. and Assilian, S. (1975) 'An experiment in linguistic synthesis with a fuzzy logic controller', *International journal of man-machine studies*. Elsevier, 7(1), pp. 1–13.

Marsland, S. (2015) *Machine learning: an algorithmic perspective*. CRC press.

Morris, R. and Thompson, K. (1979) 'Password security: a case history', *Communications of the ACM*, 22(11), pp. 594–597. doi: 10.1145/359168.359172.

Neuman, B. C. and Ts'o, T. (1994) 'Kerberos: An authentication service for computer networks', *Communications Magazine, IEEE*. IEEE, 32(9), pp. 33–38.

Nishiki, K. and Tanaka, E. (2005) 'Authentication and Access Control Agent Framework for Context-Aware Services', *2005 Symposium on Applications and the Internet Workshops (SAINT 2005 Workshops)*,

pp. 5–8. doi: 10.1109/SAINTW.2005.1620011.

Novák, V. (1999) ‘Weighted inference systems’, in *Fuzzy Sets in Approximate Reasoning and Information Systems*. Springer, pp. 191–241.

O’Gorman, L. (2003) ‘Comparing passwords, tokens, and biometrics for user authentication’, *Proceedings of the IEEE*, 91(12), pp. 2021–2040. doi: 10.1109/JPROC.2003.819611.

Ostle, B. and and Mensing, R. (1975) *Statistics in Research*. 3rd edn. Ames Iowa: The Iowa State University Press.

Paradesi, S., Doshi, P. and Swaika, S. (2009) ‘Integrating behavioral trust in web service compositions’, in *Web Services, 2009. ICWS 2009. IEEE International Conference on*, pp. 453–460.

Pascoe, J. (1998) ‘Adding generic contextual capabilities to wearable computers’, in *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*. IEEE, pp. 92–99.

Perera, C., Zaslavsky, A., Christen, P. and Georgakopoulos, D. (2014) ‘Context aware computing for the internet of things: A survey’, *IEEE Communications Surveys and Tutorials*, 16(1), pp. 414–454. doi: 10.1109/SURV.2013.042313.00197.

Pfitzmann, A. and Hansen, M. (2010) ‘A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management’.

Potts, C. (1995) ‘Using schematic scenarios to understand user needs’, in *Proceedings of the 1st conference on Designing interactive systems: processes, practices, methods, & techniques*, pp. 247–256.

Pradhan, B. and Pirasteh, S. (2010) ‘Comparison between prediction capabilities of neural network and fuzzy logic techniques for L and slide susceptibility mapping.’, *Disaster Advances*, 3(3), pp. 26–34.

Reddy, Y. V. (2006) ‘Pervasive computing: Implications, opportunities and challenges for the society’, in *1st International Symposium on Pervasive Computing and Applications*.

Rekimoto, J., Ayatsuka, Y. and Hayashi, K. (1998) 'Augment-able reality: Situated communication through physical and digital spaces', in *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*. IEEE, pp. 68–75.

Richard Gran (2012) 'What is Simulation'. Available at: https://www.youtube.com/watch?v=OCMafswcNkY.

Da Rocha, R. C. A. and Endler, M. (2012) *Context management for distributed and dynamic context-aware computing*. Springer.

Ross, T. J. (2009) *Fuzzy logic with engineering applications*. John Wiley & Sons.

Rountree, D. (2012) *Federated identity primer*. Newnes.

Rundle, M., Blakley, B., Broberg, J., Nadalin, A., Olds, D., Ruddy, M., Marcelo Thompson, M., Mello Guimarães, M. and Trevithick, P. (2007) *At a crossroads:' Personhood' and the digital identity in the information society*. STI working papers Series. OECD. Available from http://www. oecd. org/sti/working-papers.

Ryan, N. S., Pascoe, J. and Morse, D. R. (1998) 'Enhanced reality fieldwork: the context-aware archaeological assistant', in *Computer applications in archaeology*. Tempus Reparatum.

Salber, D., Dey, A. K., Orr, R. J. and Abowd, G. D. (1999) 'Designing for ubiquitous computing: A case study in context sensing'. Georgia Institute of Technology.

Schilit, B., Adams, N. and Want, R. (1994) 'Context-aware computing applications', in *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pp. 85–90.

Schilit, B. N., Theimer, M. M. and Welch, B. B. (1993) 'Customizing mobile applications', in *Proceedings USENIX Symposium on Mobile & Location-indendent Computing*.

Schilit, B. and York, N. (1995) 'Context-Aware Computing Applications', pp. 85–90.

Schmidt, A. and Van Laerhoven, K. (2001) 'How to build smart appliances?', *IEEE Personal Communications*. IEEE, 8(4), pp. 66–71.

Schumacher, G. (2012) 'Second Generation Biometrics: The Ethical, Legal and Social Context'. Edited by E. Mordini and D. Tzovaras. Dordrecht: Springer Netherlands (The International Library of Ethics, Law and Technology), 11, pp. 215–227. doi: 10.1007/978-94-007-3892-8.

Takagi, T. and Sugeno, M. (1985) 'Fuzzy identification of systems and its applications to modeling and control', *Systems, Man and Cybernetics, IEEE Transactions on*. IEEE, (1), pp. 116–132.

Taylor, C. (1989) *Sources of the self: The making of the modern identity*. Harvard University Press.

Truong, K. N., Abowd, G. D. and Brotherton, J. a (2001) 'Who, What, When, Where, How: Design Issues of Capture & Access Applications', *3rd International Conference on Ubiquitous Computing*, pp. 209–224. doi: 10.1007/3-540-45427-6_17.

Wang, L.-X. and Mendel, J. M. (1992) 'Generating fuzzy rules by learning from examples', *Systems, Man and Cybernetics, IEEE Transactions on*. IEEE, 22(6), pp. 1414–1427.

Ward, A., Jones, A. and Hopper, A. (1997) 'A new location technique for the active office', *Personal Communications, IEEE*. IEEE, 4(5), pp. 42–47.

Weiser, M. (1991) 'The computer for the 21st century', *Scientific american*. Nature Publishing Group, 265(3), pp. 94–104.

Weiss, K. P. (1988) 'Method and apparatus for positively identifying an individual'. Google Patents.

Wichmann, B. A. and Hill, I. D. (1982) 'Algorithm AS 183: An efficient and portable pseudo-random number generator', *Journal of the Royal Statistical Society. Series C (Applied Statistics)*. JSTOR, 31(2), pp. 188–190.

Windley, P. J. (2005) *Digital Identity: Unmasking identity management architecture (IMA)*. ' O'Reilly

Media, Inc.'

Zadeh, L. A. (1965) 'Fuzzy sets', *Information and Control*, 8(3), pp. 338–353. doi: http://dx.doi.org/10.1016/S0019-9958(65)90241-X.

Abowd, D., Dey, A. K., Orr, R. and Brotherton, J. (1998) 'Context-awareness in wearable and ubiquitous computing', *Virtual Reality*. Springer, 3(3), pp. 200–211.

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M. and Steggles, P. (1999) 'Towards a better understanding of context and context-awareness', in *Handheld and ubiquitous computing*. Springer, pp. 304–307.

Akhtar, Z., Micheloni, C. and Foresti, G. L. (2015) 'Biometric liveness detection: challenges and research opportunities', *IEEE Security & Privacy*. IEEE, 13(5), pp. 63–72.

Al-Karkhi, A., Al-Yasiri, A. and Linge, N. (2015) 'Discreet verification of user identity in pervasive computing environments using a non-intrusive technique', *Computers & Electrical Engineering*, 41, pp. 102–114. doi: 10.1016/j.compeleceng.2014.10.006.

Alexander, I. F. and Beus-Dukic, L. (2009) *Discovering requirements: how to specify products and services*. John Wiley & Sons.

Alexander, I. F. and Maiden, N. (2005) *Scenarios, stories, use cases: through the systems development life-cycle*. John Wiley & Sons.

Baldauf, M. (2007) 'A survey on context-aware systems Schahram Dustdar * and Florian Rosenberg', 2(4).

Behrends, E. (2000) *Introduction to Markov chains*. Springer.

Bělohlávek, R. and Klir, G. J. (2011) *Concepts and fuzzy logic*. MIT Press.

Bhatnagar, R. and Kanal, L. N. (1992) *Models of enquiry and formalisms for approximate reasoning*.

John Wiley & Sons, Inc.

Bishop, M. A. (2002) 'The art and science of computer security'. Addison-Wesley Longman Publishing Co., Inc.

Brainard, J., Juels, A., Rivest, R. L., Szydlo, M. and Yung, M. (2006) 'Fourth-factor authentication: somebody you know', in *ACM conference on computer and communications security*, pp. 168–178.

Brown, M. G. (1996) 'Supporting user mobility', in *Mobile Communications*. Springer, pp. 69–77.

Brown, P. J. (1995) 'The stick-e document: a framework for creating context-aware applications', *ELECTRONIC PUBLISHING-CHICHESTER-*. Citeseer, 8, pp. 259–272.

Brown, P. J., Bovey, J. D. and Chen, X. (1997) 'Context-aware applications: from the laboratory to the marketplace', *Personal Communications, IEEE*. IEEE, 4(5), pp. 58–64.

Van Bunningen, A. H., Feng, L. and Apers, P. M. G. (2005) 'Context for ubiquitous data management', in *Ubiquitous Data Management, 2005. UDM 2005. International Workshop on*, pp. 17–24.

Campbell, R., Al-Muhtadi, J., Naldurg, P., Sampemane, G. and Mickunas, M. D. (2003) 'Towards Security and Privacy for Pervasive Computing', in *Proceedings of the 2002 Mext-NSF-JSPS International Conference on Software Security: Theories and Systems*. Berlin, Heidelberg: Springer-Verlag (ISSS'02), pp. 1–15. Available at: http://dl.acm.org/citation.cfm?id=1765533.1765535.

Carroll, J. M. (2000) *Making use: scenario-based design of human-computer interactions*. MIT press.

Chalmers, D. (2011) *Sensing and systems in pervasive computing: Engineering context aware systems*. Springer.

Chen, G. and Kotz, D. (2000) *A survey of context-aware mobile computing research*. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.

Chen, H. (2004) *An Intelligent Broker Architecture For Pervasive Context-Aware Systems*. University of

Maryland.

Chisholm, R. M. (1997) ‘Identity Through Time’, in Rea, M. C. (ed.) *Material Constitution*. Rowman & Littlefield, p. 209.

Choi, S. and Zage, D. (2012) ‘Addressing insider threat using “where you are” as fourth factor authentication’, in *Security Technology (ICCST), 2012 IEEE International Carnahan Conference on*. IEEE, pp. 147–153.

Choi, Y., Doh, I., Park, S.-S. and Chae, K.-J. (2013) ‘Security based semantic context awareness system for M2M ubiquitous healthcare service’, in *Ubiquitous Information Technologies and Applications*. Springer, pp. 187–196.

Chun-Li, L. I. N., Hung-Min, S. U. N. and Hwang, T. (2001) ‘Attacks and solutions on strong-password authentication’, *IEICE transactions on communications*. The Institute of Electronics, Information and Communication Engineers, 84(9), pp. 2622–2627.

Commission, S. S. (2006) ‘Guidance on Multi-factor Authentication’. Available at: http://ict.govt.nz/assets/Uploads/Documents/egif-authentication-multi-factor-guidance-june-2006.pdf.

Conti, M., Das, S. K., Bisdikian, C., Kumar, M., Ni, L. M., Passarella, A., Roussos, G., Tröster, G., Tsudik, G. and Zambonelli, F. (2012) ‘Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber--physical convergence’, *Pervasive and Mobile Computing*. Elsevier, 8(1), pp. 2–21.

Covington, M. J., Fogla, P., Zhan, Z. and Ahamad, M. (2002) ‘A context-aware security architecture for emerging applications’, in *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, pp. 249–258.

Craig, I. and Whitty, M. (2017) ‘Region Formation for Efficient Offline Location Prediction’, *IEEE*

Pervasive Computing. IEEE, 16(1), pp. 66–73.

Cranor, L. F. and Garfinkel, S. (2008) *Security and Usability: Designing Secure Systems that People Can Use*. O'Reilly Media. Available at: http://books.google.co.uk/books?id=wDVhy9EyEAEC.

Cristofaro, E. De (2014) 'A Comparative Usability Study of Two-Factor Authentication'.

Dawson, F., Dawson, F., Stenerson, D. and Stenerson, D. (1998) *RFC 2445 - Internet Calendaring and Scheduling Core Object Specification (iCalendar)*, *IETF Request For Comments*.

Debian Project (2017) *Raspbian*, *Debian Project*.

Deng, F. M. (2011) *War of visions: Conflict of identities in the Sudan*. Brookings Institution Press.

Dey, A. K. (1998) 'Context-aware computing: The CyberDesk project', in *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, pp. 51–54.

Dourish, P. (2004) 'What we talk about when we talk about context', *Personal and Ubiquitous Computing*, 8(1), pp. 19–30. doi: 10.1007/s00779-003-0253-8.

Emmanouilidis, C., Koutsiamanis, R.-A. and Tasidou, A. (2013) 'Mobile guides: Taxonomy of architectures, context awareness, technologies and applications', *Journal of Network and Computer Applications*. Elsevier, 36(1), pp. 103–125. doi: 10.1016/j.jnca.2012.04.007.

Fearon, J. D. (1999) 'What is identity (as we now use the word)', *Unpublished manuscript, Stanford University, Stanford, Calif*.

Fickas, S., Kortuem, G. and Segall, Z. (1997) 'Software organization for dynamic and adaptable wearable systems', in *Wearable Computers, 1997. Digest of Papers., First International Symposium on*. IEEE, pp. 56–63.

Gray, A. R. and MacDonell, S. G. (1997) 'A comparison of techniques for developing predictive models of software metrics', *Information and software technology*. Elsevier, 39(6), pp. 425–437.

Greenfield, A. (2010) *Everyware: The dawning age of ubiquitous computing*. New Riders.

Hayashi, E., Hong, J., Das, S., Amini, S. and Oakley, I. (2013) 'CASA : Context - Aware Scalable Authentication', pp. 1–10.

Henricksen, K. (2003) *A framework for context-aware pervasive computing applications*. University of Queensland Queensland.

Hong, J., Suh, E. and Kim, S.-J. (2009) 'Context-aware systems: A literature review and classification', *Expert Systems with Applications*. Elsevier Ltd, 36(4), pp. 8509–8522. doi: 10.1016/j.eswa.2008.10.071.

Hong, T.-P. and Lee, C.-Y. (1996) 'Induction of fuzzy rules and membership functions from training examples', *Fuzzy sets and Systems*. Elsevier, 84(1), pp. 33–47.

Hughes, J. and Maler, E. (2005) 'Security Assertion Markup Language (SAML) V2. 0 Technical Overview', *OASIS SSTC Working Draft sstc-saml-tech-overview-2.0-draft-08*, pp. 29–38.

Hull, R., Neaves, P. and Bedford-Roberts, J. (1997) 'Towards situated computing', in *Wearable Computers, 1997. Digest of Papers., First International Symposium on*. IEEE, pp. 146–153.

Huynh, T. D., Jennings, N. R. and Shadbolt, N. R. (2006) 'An integrated trust and reputation model for open multi-agent systems', *Autonomous Agents and Multi-Agent Systems*. Springer, 13(2), pp. 119–154.

IL, Barry L. Nelson, Jerry Banks, J. S. C. (2009) *Discrete-Event System Simulation*. 5th edn. United States: Pearson Education (US).

Indulska, J. and Sutton, P. (2003) 'Location management in pervasive systems', in *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21*, pp. 143–151.

ITU-T (2009) *NGN identity management framework*, *Recommendation ITU-T Y.2720*. Available at: https://www.itu.int/rec/T-REC-Y.2720-200901-I.

Jang, B., Park, S., Chang, H., Ahn, H. and Choi, E. (2010) 'A study of context-awareness RBAC model

using user profile on ubiquitous computing', *Communications in Computer and Information Science*, 120 CCIS, pp. 205–213. doi: 10.1007/978-3-642-17604-3_23.

Jaroucheh, Z., Liu, X. and Smith, S. (2010) 'CANDEL: Product Line Based Dynamic Context Management for Pervasive Applications', *2010 International Conference on Complex, Intelligent and Software Intensive Systems*. Ieee, pp. 209–216. doi: 10.1109/CISIS.2010.21.

Jia, X., Cao, H., Tian, J. and Liu, Y. (2014) 'CONTEXT-AWARE ADAPTIVE AUTHENTICATION METHOD AND APPARATUS'.

Jsang, A. and Ismail, R. (2002) 'The beta reputation system', in *Proceedings of the 15th bled electronic commerce conference*, pp. 2502–2511.

Kim, I. and Lee, Y. (2006) 'CAST: Design and implementation of secure context-awareness simulation toolkit for ubiquitous computing research environment', *Lecture Notes in Control and Information Sciences*, 344, pp. 1040–1055. doi: 10.1007/11816492_135.

Kortuem, G., Segall, Z. and Bauer, M. (1998) 'Context-aware, adaptive wearable computers as remote interfaces to'intelligent'environments', in *1998 16th International Symposium on Wearable Computers*. IEEE Computer Society, p. 58.

Kothari, C. R. (2004) *Research methodology: Methods and techniques*. New Age International.

Krumm, J. (2009) *Ubiquitous computing fundamentals*. CRC Press.

Laliwala, Z. and Chaudhary, S. (2008) 'Event-driven service-oriented architecture', in *Service Systems and Service Management, 2008 International Conference on*, pp. 1–6.

Van Leekwijck, W. and Kerre, E. E. (1999) 'Defuzzification: criteria and classification', *Fuzzy sets and systems*. Elsevier, 108(2), pp. 159–178.

Lesani, M. and Montazeri, N. (2009) 'Fuzzy trust aggregation and personalized trust inference in virtual

social networks', *Computational Intelligence*. Wiley Online Library, 25(2), pp. 51–83.

Li, X., Martínez, J.-F. and Rubio, G. (2017) 'Towards a Hybrid Approach to Context Reasoning for Underwater Robots', *Applied Sciences*. Multidisciplinary Digital Publishing Institute, 7(2), p. 183.

Lin, Z.-H. and Fu, L.-C. (2007) 'Multi-user preference model and service provision in a smart home environment', in *Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on*, pp. 759–764.

Lo, H. Z. (2014) *Discrete Event Simulation*, *University of Massachusetts Boston*. Available at: http://www.cs.umb.edu/~henryzlo/cs310/notes/simulation.pdf.

De Luca, A., Hang, A., Brudy, F., Lindner, C. and Hussmann, H. (2012) 'Touch me once and i know it's you!: implicit authentication based on touch screen patterns', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 987–996.

Luger, G. F. (2005) *Artificial intelligence: structures and strategies for complex problem solving*. Pearson education.

Malek, B., Miri, A. and Karmouch, A. (2008) 'A framework for context-aware authentication'. IET.

Mamdani, E. H. and Assilian, S. (1975) 'An experiment in linguistic synthesis with a fuzzy logic controller', *International journal of man-machine studies*. Elsevier, 7(1), pp. 1–13.

Marsland, S. (2015) *Machine learning: an algorithmic perspective*. CRC press.

Morris, R. and Thompson, K. (1979) 'Password security: a case history', *Communications of the ACM*, 22(11), pp. 594–597. doi: 10.1145/359168.359172.

Neuman, B. C. and Ts'o, T. (1994) 'Kerberos: An authentication service for computer networks', *Communications Magazine, IEEE*. IEEE, 32(9), pp. 33–38.

Nishiki, K. and Tanaka, E. (2005) 'Authentication and Access Control Agent Framework for Context-

Aware Services', *2005 Symposium on Applications and the Internet Workshops (SAINT 2005 Workshops)*, pp. 5–8. doi: 10.1109/SAINTW.2005.1620011.

Novák, V. (1999) 'Weighted inference systems', in *Fuzzy Sets in Approximate Reasoning and Information Systems*. Springer, pp. 191–241.

O'Gorman, L. (2003) 'Comparing passwords, tokens, and biometrics for user authentication', *Proceedings of the IEEE*, 91(12), pp. 2021–2040. doi: 10.1109/JPROC.2003.819611.

Ostle, B. and and Mensing, R. (1975) *Statistics in Research*. 3rd edn. Ames Iowa: The Iowa State University Press.

Paradesi, S., Doshi, P. and Swaika, S. (2009) 'Integrating behavioral trust in web service compositions', in *Web Services, 2009. ICWS 2009. IEEE International Conference on*, pp. 453–460.

Pascoe, J. (1998) 'Adding generic contextual capabilities to wearable computers', in *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*. IEEE, pp. 92–99.

Perera, C., Zaslavsky, A., Christen, P. and Georgakopoulos, D. (2014) 'Context aware computing for the internet of things: A survey', *IEEE Communications Surveys and Tutorials*, 16(1), pp. 414–454. doi: 10.1109/SURV.2013.042313.00197.

Pfitzmann, A. and Hansen, M. (2010) 'A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management'.

Potts, C. (1995) 'Using schematic scenarios to understand user needs', in *Proceedings of the 1st conference on Designing interactive systems: processes, practices, methods, & techniques*, pp. 247–256.

Pradhan, B. and Pirasteh, S. (2010) 'Comparison between prediction capabilities of neural network and fuzzy logic techniques for L and slide susceptibility mapping.', *Disaster Advances*, 3(3), pp. 26–34.

Reddy, Y. V. (2006) 'Pervasive computing: Implications, opportunities and challenges for the society', in

1st International Symposium on Pervasive Computing and Applications.

Rekimoto, J., Ayatsuka, Y. and Hayashi, K. (1998) 'Augment-able reality: Situated communication through physical and digital spaces', in *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*. IEEE, pp. 68–75.

Richard Gran (2012) 'What is Simulation'. Available at: https://www.youtube.com/watch?v=OCMafswcNkY.

Da Rocha, R. C. A. and Endler, M. (2012) *Context management for distributed and dynamic context-aware computing*. Springer.

Ross, T. J. (2009) *Fuzzy logic with engineering applications*. John Wiley & Sons.

Rountree, D. (2012) *Federated identity primer*. Newnes.

Rundle, M., Blakley, B., Broberg, J., Nadalin, A., Olds, D., Ruddy, M., Marcelo Thompson, M., Mello Guimarães, M. and Trevithick, P. (2007) *At a crossroads:' Personhood' and the digital identity in the information society*. STI working papers Series. OECD. Available from http://www. oecd. org/sti/working-papers.

Ryan, N. S., Pascoe, J. and Morse, D. R. (1998) 'Enhanced reality fieldwork: the context-aware archaeological assistant', in *Computer applications in archaeology*. Tempus Reparatum.

Salber, D., Dey, A. K., Orr, R. J. and Abowd, G. D. (1999) 'Designing for ubiquitous computing: A case study in context sensing'. Georgia Institute of Technology.

Schilit, B., Adams, N. and Want, R. (1994) 'Context-aware computing applications', in *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pp. 85–90.

Schilit, B. N., Theimer, M. M. and Welch, B. B. (1993) 'Customizing mobile applications', in *Proceedings USENIX Symposium on Mobile & Location-indendent Computing*.

Schilit, B. and York, N. (1995) ‘Context-Aware Computing Applications’, pp. 85–90.

Schmidt, A. and Van Laerhoven, K. (2001) ‘How to build smart appliances?’, *IEEE Personal Communications*. IEEE, 8(4), pp. 66–71.

Schumacher, G. (2012) ‘Second Generation Biometrics: The Ethical, Legal and Social Context’. Edited by E. Mordini and D. Tzovaras. Dordrecht: Springer Netherlands (The International Library of Ethics, Law and Technology), 11, pp. 215–227. doi: 10.1007/978-94-007-3892-8.

Takagi, T. and Sugeno, M. (1985) ‘Fuzzy identification of systems and its applications to modeling and control’, *Systems, Man and Cybernetics, IEEE Transactions on*. IEEE, (1), pp. 116–132.

Taylor, C. (1989) *Sources of the self: The making of the modern identity*. Harvard University Press.

Truong, K. N., Abowd, G. D. and Brotherton, J. a (2001) ‘Who, What, When, Where, How: Design Issues of Capture & Access Applications’, *3rd International Conference on Ubiquitous Computing*, pp. 209–224. doi: 10.1007/3-540-45427-6_17.

Wang, L.-X. and Mendel, J. M. (1992) ‘Generating fuzzy rules by learning from examples’, *Systems, Man and Cybernetics, IEEE Transactions on*. IEEE, 22(6), pp. 1414–1427.

Ward, A., Jones, A. and Hopper, A. (1997) ‘A new location technique for the active office’, *Personal Communications, IEEE*. IEEE, 4(5), pp. 42–47.

Weiser, M. (1991) ‘The computer for the 21st century’, *Scientific american*. Nature Publishing Group, 265(3), pp. 94–104.

Weiss, K. P. (1988) ‘Method and apparatus for positively identifying an individual’. Google Patents.

Wichmann, B. A. and Hill, I. D. (1982) ‘Algorithm AS 183: An efficient and portable pseudo-random number generator’, *Journal of the Royal Statistical Society. Series C (Applied Statistics)*. JSTOR, 31(2), pp. 188–190.

Windley, P. J. (2005) *Digital Identity: Unmasking identity management architecture (IMA)*. ‘ O’Reilly Media, Inc.’

Zadeh, L. A. (1965) ‘Fuzzy sets’, *Information and Control*, 8(3), pp. 338–353. doi: http://dx.doi.org/10.1016/S0019-9958(65)90241-X.

APPENDICES

Appendix A
CAIV Inference Engine Rules

3- if (Calendar is Low) and (History Log is Low) and (User Preferences is Low) and (Ambient Object is High) Then (Trust is Medium Low)

4- if (Calendar is Low) and (History Log is Low) and (User Preferences is Medium) and (Ambient Object is Low) Then (Trust is Medium Low)

5- if (Calendar is Low) and (History Log is Low) and (User Preferences is Medium) and (Ambient Object is Medium) Then (Trust is Medium Low)

6- if (Calendar is Low) and (History Log is Low) and (User Preferences is Medium) and (Ambient Object is High) Then (Trust is Medium Low)

7- if (Calendar is Low) and (History Log is Low) and (User Preferences is High) and (Ambient Object is Low) Then (Trust is Medium Low)

8- if (Calendar is Low) and (History Log is Low) and (User Preferences is High) and (Ambient Object is Medium) Then (Trust is Medium)

9- if (Calendar is Low) and (History Log is Low) and (User Preferences is High) and (Ambient Object is High) Then (Trust is Medium High)

10- if (Calendar is Low) and (History Log is Medium) and (User Preferences is Low) and (Ambient Object is Low) Then (Trust is Medium Low)

11- if (Calendar is Low) and (History Log is Medium) and (User Preferences is Low) and (Ambient Object is Medium) Then (Trust is Medium Low)

12- if (Calendar is Low) and (History Log is Medium) and (User Preferences is Low) and (Ambient Object is High) Then (Trust is Medium Low)

13- if (Calendar is Low) and (History Log is Medium) and (User Preferences is Medium) and (Ambient Object is Low) Then (Trust is Medium Low)

14- if (Calendar is Low) and (History Log is Medium) and (User Preferences is Medium) and (Ambient Object is Medium) Then (Trust is Medium)

15- if (Calendar is Low) and (History Log is Medium) and (User Preferences is Medium) and (Ambient Object is High) Then (Trust is Medium)

16- if (Calendar is Low) and (History Log is Medium) and (User Preferences is High) and (Ambient Object is Low) Then (Trust is Medium Low)

17- if (Calendar is Low) and (History Log is Medium) and (User Preferences is High) and (Ambient Object is Medium) Then (Trust is Medium)

18- if (Calendar is Low) and (History Log is Medium) and (User Preferences is High) and (Ambient Object is High) Then (Trust is Medium High)

19- if (Calendar is Low) and (History Log is High) and (User Preferences is Low) and (Ambient Object is Low) Then (Trust is Medium Low)

20- if (Calendar is Low) and (History Log is High) and (User Preferences is Low) and (Ambient Object is Medium) Then (Trust is Medium)

21- if (Calendar is Low) and (History Log is High) and (User Preferences is Low) and (Ambient Object is High) Then (Trust is Medium High)

22- if (Calendar is Low) and (History Log is High) and (User Preferences is Medium) and (Ambient Object is Low) Then (Trust is Medium Low)

23- if (Calendar is Low) and (History Log is High) and (User Preferences is Medium) and (Ambient Object is Medium) Then (Trust is Medium)

24- if (Calendar is Low) and (History Log is High) and (User Preferences is Medium) and (Ambient Object is High) Then (Trust is Medium High)

25- if (Calendar is Low) and (History Log is High) and (User Preferences is High) and (Ambient Object is Low) Then (Trust is Medium)

26- if (Calendar is Low) and (History Log is High) and (User Preferences is High) and (Ambient Object is Medium) Then (Trust is Medium)

27- if (Calendar is Low) and (History Log is High) and (User Preferences is High) and (Ambient Object is High) Then (Trust is Medium High)

28- if (Calendar is Medium) and (History Log is Low) and (User Preferences is Low) and (Ambient Object is Low) Then (Trust is Medium Low)

29- if (Calendar is Medium) and (History Log is Low) and (User Preferences is Low) and (Ambient Object is Medium) Then (Trust is Medium)

30- if (Calendar is Medium) and (History Log is Low) and (User Preferences is Low) and (Ambient Object is High) Then (Trust is Medium High)

31- if (Calendar is Medium) and (History Log is Low) and (User Preferences is Medium) and (Ambient Object is Low) Then (Trust is Medium Low)

32- if (Calendar is Medium) and (History Log is Low) and (User Preferences is Medium) and (Ambient Object is Medium) Then (Trust is Medium)

33- if (Calendar is Medium) and (History Log is Low) and (User Preferences is Medium) and (Ambient Object is High) Then (Trust is Medium High)

34- if (Calendar is Medium) and (History Log is Low) and (User Preferences is High) and (Ambient Object is Low) Then (Trust is Medium Low)

35- if (Calendar is Medium) and (History Log is Low) and (User Preferences is High) and (Ambient Object is Medium) Then (Trust is Medium)

36- if (Calendar is Medium) and (History Log is Low) and (User Preferences is High) and (Ambient Object is High) Then (Trust is Medium High)

37- if (Calendar is Medium) and (History Log is Medium) and (User Preferences is Low) and (Ambient Object is Low) Then (Trust is Medium Low)

38- if (Calendar is Medium) and (History Log is Medium) and (User Preferences is Low) and (Ambient Object is Medium) Then (Trust is Medium)

39- if (Calendar is Medium) and (History Log is Medium) and (User Preferences is Low) and (Ambient Object is High) Then (Trust is Medium High)

40- if (Calendar is Medium) and (History Log is Medium) and (User Preferences is Medium) and (Ambient Object is Low) Then (Trust is Medium Low)

41- if (Calendar is Medium) and (History Log is Medium) and (User Preferences is Medium) and (Ambient Object is Medium) Then (Trust is Medium)

42- if (Calendar is Medium) and (History Log is Medium) and (User Preferences is Medium) and (Ambient Object is High) Then (Trust is Medium High)

43- if (Calendar is Medium) and (History Log is Medium) and (User Preferences is High) and (Ambient Object is Low) Then (Trust is Medium)

44- if (Calendar is Medium) and (History Log is Medium) and (User Preferences is High) and (Ambient Object is Medium) Then (Trust is Medium High)

45- if (Calendar is Medium) and (History Log is Medium) and (User Preferences is High) and (Ambient Object is High) Then (Trust is Medium High)

46- if (Calendar is Medium) and (History Log is High) and (User Preferences is Low) and (Ambient Object is Low) Then (Trust is Medium)

47- if (Calendar is Medium) and (History Log is High) and (User Preferences is Low) and (Ambient Object is Medium) Then (Trust is Medium)

48- if (Calendar is Medium) and (History Log is High) and (User Preferences is Low) and (Ambient Object is High) Then (Trust is Medium High)

49- if (Calendar is Medium) and (History Log is High) and (User Preferences is Medium) and (Ambient Object is Low) Then (Trust is Medium)

50- if (Calendar is Medium) and (History Log is High) and (User Preferences is Medium) and (Ambient Object is Medium) Then (Trust is Medium High)

51- if (Calendar is Medium) and (History Log is High) and (User Preferences is Medium) and (Ambient Object is High) Then (Trust is Medium High)

52- if (Calendar is Medium) and (History Log is High) and (User Preferences is High) and (Ambient Object is Low) Then (Trust is Medium)

53- if (Calendar is Medium) and (History Log is High) and (User Preferences is High) and (Ambient Object is Medium) Then (Trust is Medium High)

54- if (Calendar is Medium) and (History Log is High) and (User Preferences is High) and (Ambient Object is High) Then (Trust is Medium High)

55- if (Calendar is High) and (History Log is Low) and (User Preferences is Low) and (Ambient Object is Low) Then (Trust is Medium)

56- if (Calendar is High) and (History Log is Low) and (User Preferences is Low) and (Ambient Object is Medium) Then (Trust is Medium)

57- if (Calendar is High) and (History Log is Low) and (User Preferences is Low) and (Ambient Object is High) Then (Trust is High)

58- if (Calendar is High) and (History Log is Low) and (User Preferences is Medium) and (Ambient Object is Low) Then (Trust is Medium)

59- if (Calendar is High) and (History Log is Low) and (User Preferences is Medium) and (Ambient Object is Medium) Then (Trust is Medium High)

60- if (Calendar is High) and (History Log is Low) and (User Preferences is Medium) and (Ambient Object is High) Then (Trust is Medium High)

61- if (Calendar is High) and (History Log is Low) and (User Preferences is High) and (Ambient Object is Low) Then (Trust is Medium)

62- if (Calendar is High) and (History Log is Low) and (User Preferences is High) and (Ambient Object is Medium) Then (Trust is Medium High)

63- if (Calendar is High) and (History Log is Low) and (User Preferences is High) and (Ambient Object is High) Then (Trust is Medium High)

64- if (Calendar is High) and (History Log is Medium) and (User Preferences is Low) and (Ambient Object is Low) Then (Trust is Medium)

65- if (Calendar is High) and (History Log is Medium) and (User Preferences is Low) and (Ambient Object is Medium) Then (Trust is Medium High)

66- if (Calendar is High) and (History Log is Medium) and (User Preferences is Low) and (Ambient Object is High) Then (Trust is Medium High)

67- if (Calendar is High) and (History Log is Medium) and (User Preferences is Medium) and (Ambient Object is Low) Then (Trust is Medium)

68- if (Calendar is High) and (History Log is Medium) and (User Preferences is Medium) and (Ambient Object is Medium) Then (Trust is Medium High)

69- if (Calendar is High) and (History Log is Medium) and (User Preferences is Medium) and (Ambient Object is High) Then (Trust is Medium High)

70- if (Calendar is High) and (History Log is Medium) and (User Preferences is High) and (Ambient Object is Low) Then (Trust is Medium)

71- if (Calendar is High) and (History Log is Medium) and (User Preferences is High) and (Ambient Object is Medium) Then (Trust is Medium High)

72- if (Calendar is High) and (History Log is Medium) and (User Preferences is High) and (Ambient Object is High) Then (Trust is High)

73- if (Calendar is High) and (History Log is High) and (User Preferences is Low) and (Ambient Object is Low) Then (Trust is Medium)

74- if (Calendar is High) and (History Log is High) and (User Preferences is Low) and (Ambient Object is Medium) Then (Trust is Medium High)

75- if (Calendar is High) and (History Log is High) and (User Preferences is Low) and (Ambient Object is High) Then (Trust is Medium High)

76- if (Calendar is High) and (History Log is High) and (User Preferences is Medium) and (Ambient Object is Low) Then (Trust is Medium)

77- if (Calendar is High) and (History Log is High) and (User Preferences is Medium) and (Ambient Object is Medium) Then (Trust is Medium High)

78- if (Calendar is High) and (History Log is High) and (User Preferences is Medium) and (Ambient Object is High) Then (Trust is High)

79- if (Calendar is High) and (History Log is High) and (User Preferences is High) and (Ambient Object is Low) Then (Trust is Medium High)

80- if (Calendar is High) and (History Log is High) and (User Preferences is High) and (Ambient Object is Medium) Then (Trust is High)

Appendix B
CAIV Prototype on Raspberry Pi 3 Code Sample

```
from array import *
import mysql.connector
import datetime
import bluetooth
import MFRC522
import signal
import RPi.GPIO as GPIO
from time import sleep
import time
from output import *
ble=[]
GPIO.setmode(GPIO.BOARD)
password=[0,0,0,0]
#initial values and constants
timenowsend=""
# RFID location can be changed to test the CAIV decision
rfid_location=28
# Enterance time tolerance (in minutes)
time_tolerance=55
# The assumed date ( it can be vary to test the system's flexibility )
sdate=datetime.datetime(2015, 5, 4)
# User ID
usid=2
# Trust Threshold Value
threshold=0.4
# Ambient Objects Value
global amb
amb=0.0
#########################################################################
#### Searching in the Fuzzy Inference Engine (inside the output.py)###
#########################################################################
def search (a,b,c,d):
 if b>0.9:
     b=0.9
 for i in range(0,len(ruls_heap)):
    temp=ruls_heap[i]
    if temp[0]==a and  temp[1]==b and  temp[2]==c and  temp[3]==d:
        return temp[4]
        break;
############################## Begin Table _ Preferences ####################
def view_preferences(uid,location):
    t=0.0
conn=mysql.connector.connect(user='root',password='root',host='localhost',dat
abase='uc')
    cursor=conn.cursor()
    # location     probability percentage
    query = ("SELECT c.LocationlocationID,c.probabilityRatio FROM
UserPreferences c WHERE UseruserID="+str(uid)+" AND
LocationlocationID="+str(location)+"; ")
```

```
    cursor.execute(query)
    tmp=[]
    for row in cursor.fetchall() :
        if row[1]>0:
            t=row[1]
        else:
            t=0.0
    conn.close()
    return(t)
########################## Begin_Calendar_Table_View#####################
```

Appendix C
Sample Data of Legitimate User History log on 1-7-2016 of Experiment 1

ID	logStatus	logTime	Weights	trust_value	P_Location	Location	User ID
1	Success	00:01:00	2090	0.25	0	0	1
2	Success	00:02:00	0090	0.25	0	32	1
3	Success	00:03:00	0090	0.25	32	31	1
4	Success	00:04:00	9060	0.5	31	30	1
5	Failed	00:05:00	0010	0.0875525	30	30	1
6	Success	00:06:00	0060	0.5	30	27	1
7	Failed	00:07:00	2010	0.102964	27	5	1
8	Success	00:08:00	9060	0.5	27	27	1
9	Success	00:01:00	0060	0.5	0	27	2
10	Success	00:02:00	5090	0.25	27	9	2
11	Success	00:03:00	9090	0.5	9	16	2
12	Success	00:04:00	0098	0.75	16	34	2
13	Success	00:05:00	0090	0.25	34	35	2
14	Success	00:06:00	9090	0.5	35	0	2
15	Success	00:07:00	9090	0.5	0	8	2
16	Success	00:08:00	9090	0.5	8	3	2
17	Success	00:01:00	5090	0.25	0	30	3
18	Success	00:02:00	9090	0.5	30	12	3
19	Success	00:03:00	0098	0.75	12	24	3
20	Success	00:04:00	2090	0.25	24	31	3
21	Success	00:05:00	0090	0.25	31	27	3
22	Success	00:06:00	5090	0.25	27	17	3
23	Success	00:07:00	0090	0.25	17	27	3
24	Failed	00:08:00	2010	0.102964	27	32	3
25	Success	00:01:00	5010	0.25	0	32	4
26	Success	00:02:00	0067	0.5	32	25	4
27	Success	00:03:00	0090	0.25	25	28	4
28	Success	00:04:00	5010	0.25	28	12	4
29	Success	00:05:00	5010	0.25	12	6	4
30	Failed	00:06:00	0010	0.0875525	6	28	4
31	Failed	00:07:00	0010	0.0875525	6	32	4
32	Failed	00:08:00	2010	0.102964	6	32	4
33	Success	00:01:00	0090	0.25	0	31	5

34	Success	00:02:00	5060	0.5	31	27	5
35	Success	00:03:00	5060	0.5	27	0	5
36	Success	00:04:00	0060	0.5	0	28	5
37	Success	00:05:00	5060	0.5	28	20	5
38	Success	00:06:00	0090	0.25	20	33	5
39	Success	00:07:00	5060	0.5	33	4	5
40	Success	00:08:00	0060	0.5	4	35	5
41	Failed	00:01:00	0010	0.0875525	0	27	6
42	Success	00:02:00	0066	0.5	0	34	6
43	Success	00:03:00	5060	0.5	34	4	6
44	Success	00:04:00	9060	0.5	4	29	6
45	Success	00:05:00	5010	0.25	29	12	6
46	Success	00:06:00	5060	0.5	12	32	6
47	Success	00:07:00	0060	0.5	32	27	6
48	Success	00:08:00	5010	0.25	27	27	6
49	Success	00:01:00	0060	0.5	0	28	7
50	Success	00:02:00	5060	0.5	28	7	7
51	Failed	00:03:00	0010	0.0875525	7	26	7
52	Failed	00:04:00	2010	0.102964	7	18	7
53	Success	00:05:00	2090	0.25	7	4	7
54	Failed	00:06:00	0010	0.0875525	4	26	7
55	Success	00:07:00	0090	0.25	4	34	7
56	Failed	00:08:00	0010	0.0875525	34	33	7
57	Success	00:01:00	5090	0.25	0	27	8
58	Success	00:02:00	5010	0.25	27	21	8
59	Success	00:03:00	0090	0.25	21	30	8
60	Success	00:04:00	0090	0.25	30	28	8
61	Success	00:05:00	2090	0.25	28	29	8
62	Success	00:06:00	9010	0.5	29	11	8
63	Failed	00:07:00	0010	0.0875525	11	28	8
64	Success	00:08:00	5090	0.25	11	28	8
65	Success	00:01:00	2060	0.5	0	29	9
66	Success	00:02:00	0019	0.25	29	34	9
67	Success	00:03:00	0090	0.25	34	33	9
68	Success	00:04:00	9090	0.5	33	27	9
69	Success	00:05:00	5060	0.5	27	13	9
70	Success	00:06:00	9060	0.5	13	17	9
71	Success	00:07:00	5060	0.5	17	8	9
72	Success	00:08:00	0060	0.5	8	24	9

73	Success	00:01:00	0060	0.5	0	29	10
74	Success	00:02:00	2090	0.25	29	9	10
75	Success	00:03:00	2090	0.25	9	5	10
76	Success	00:04:00	9060	0.5	5	6	10
77	Success	00:05:00	0090	0.25	6	28	10
78	Success	00:06:00	9060	0.5	28	28	10
79	Success	00:07:00	5060	0.5	28	14	10
80	Success	00:08:00	0090	0.25	14	35	10
81	Success	00:01:00	9010	0.5	0	30	11
82	Success	00:02:00	5010	0.25	30	10	11
83	Success	00:03:00	0060	0.5	10	29	11
84	Success	00:04:00	5010	0.25	29	19	11
85	Success	00:05:00	9060	0.5	19	31	11
86	Success	00:06:00	9060	0.5	31	29	11
87	Failed	00:07:00	0010	0.0875525	29	31	11
88	Success	00:08:00	0060	0.5	29	33	11
89	Success	00:01:00	5010	0.25	0	30	12
90	Success	00:02:00	0066	0.5	30	25	12
91	Success	00:03:00	9060	0.5	25	7	12
92	Success	00:04:00	0060	0.5	7	28	12
93	Success	00:05:00	2060	0.5	28	9	12
94	Success	00:06:00	0060	0.5	9	32	12
95	Success	00:07:00	9060	0.5	32	19	12
96	Success	00:08:00	0060	0.5	19	35	12

Appendix D
Sample Data of Illegitimate User History log on 1-7-2016 of Experiment 2

ID	logStatus	logTime	Weights	trust_value	P_Location	Location	User ID
1	Failed	00:01:00	0000	0.08	0	35	1
2	Failed	00:02:00	0000	0.08	0	2	1
3	Failed	00:03:00	0000	0.08	0	32	1
4	Failed	00:04:00	0000	0.08	0	8	1
5	Failed	00:05:00	0000	0.08	0	1	1
6	Failed	00:06:00	0000	0.08	0	11	1
7	Failed	00:07:00	0000	0.08	0	16	1
8	Failed	00:08:00	0000	0.08	0	18	1
9	Failed	00:01:00	0000	0.08	0	33	2
10	Failed	00:02:00	0000	0.08	0	13	2
11	Failed	00:03:00	0000	0.08	0	35	2
12	Failed	00:04:00	0000	0.08	0	20	2
13	Failed	00:05:00	0000	0.08	0	16	2
14	Success	00:06:00	0009	0.25	0	23	2
15	Failed	00:07:00	0000	0.08	23	11	2
16	Failed	00:08:00	0000	0.08	23	32	2
17	Failed	00:01:00	0000	0.08	0	12	3
18	Success	00:02:00	9000	0.5	0	12	3
19	Failed	00:03:00	0000	0.08	12	29	3
20	Failed	00:04:00	0000	0.08	12	28	3
21	Failed	00:05:00	0000	0.08	12	11	3
22	Failed	00:06:00	0000	0.08	12	14	3
23	Failed	00:07:00	0000	0.08	12	35	3
24	Failed	00:08:00	0000	0.08	12	13	3
25	Failed	00:01:00	0000	0.08	0	6	4
26	Failed	00:02:00	0000	0.08	0	13	4
27	Failed	00:03:00	0000	0.08	0	11	4
28	Failed	00:04:00	0000	0.08	0	7	4
29	Failed	00:05:00	0000	0.08	0	2	4
30	Success	00:06:00	0007	0.25	0	25	4
31	Failed	00:07:00	0000	0.08	25	12	4
32	Failed	00:08:00	0000	0.08	25	14	4
33	Failed	00:01:00	0000	0.08	0	19	5

34	Failed	00:02:00	0000	0.08	0	21	5
35	Failed	00:03:00	0000	0.08	0	12	5
36	Failed	00:04:00	0000	0.08	0	23	5
37	Failed	00:05:00	0000	0.08	0	23	5
38	Failed	00:06:00	0000	0.08	0	8	5
39	Success	00:07:00	0007	0.25	0	25	5
40	Failed	00:08:00	0005	0.0875525	25	34	5
41	Failed	00:01:00	0000	0.08	0	21	6
42	Failed	00:02:00	0000	0.08	0	7	6
43	Failed	00:03:00	0000	0.08	0	28	6
44	Failed	00:04:00	0000	0.08	0	28	6
45	Failed	00:05:00	0000	0.08	0	15	6
46	Failed	00:06:00	0000	0.08	0	3	6
47	Failed	00:07:00	0000	0.08	0	11	6
48	Failed	00:08:00	0000	0.08	0	30	6
49	Failed	00:01:00	0000	0.08	0	14	7
50	Failed	00:02:00	0000	0.08	0	9	7
51	Failed	00:03:00	0000	0.08	0	33	7
52	Failed	00:04:00	0000	0.08	0	28	7
53	Failed	00:05:00	0000	0.08	0	7	7
54	Success	00:06:00	0007	0.25	0	25	7
55	Failed	00:07:00	0000	0.08	25	31	7
56	Failed	00:08:00	0005	0.0875525	25	23	7
57	Success	00:01:00	0008	0.25	0	25	8
58	Failed	00:02:00	0000	0.08	25	17	8
59	Failed	00:03:00	0000	0.08	25	14	8
60	Failed	00:04:00	0000	0.08	25	34	8
61	Failed	00:05:00	0000	0.08	25	6	8
62	Failed	00:06:00	0000	0.08	25	35	8
63	Failed	00:07:00	0000	0.08	25	13	8
64	Failed	00:08:00	0000	0.08	25	14	8
65	Failed	00:01:00	0000	0.08	0	5	9
66	Success	00:02:00	0009	0.25	0	23	9
67	Failed	00:03:00	0000	0.08	23	4	9
68	Failed	00:04:00	0000	0.08	23	14	9
69	Failed	00:05:00	0000	0.08	23	5	9
70	Success	00:06:00	0009	0.25	23	23	9
71	Failed	00:07:00	0000	0.08	23	19	9
72	Failed	00:08:00	0000	0.08	23	21	9

73	Failed	00:01:00	0000	0.08	0	4	10
74	Failed	00:02:00	0000	0.08	0	4	10
75	Failed	00:03:00	0000	0.08	0	18	10
76	Failed	00:04:00	0000	0.08	0	12	10
77	Failed	00:05:00	0000	0.08	0	21	10
78	Failed	00:06:00	0000	0.08	0	1	10
79	Failed	00:07:00	0000	0.08	0	11	10
80	Failed	00:08:00	0000	0.08	0	29	10
81	Success	00:01:00	9000	0.5	0	30	11
82	Failed	00:02:00	0000	0.08	30	3	11
83	Failed	00:03:00	0000	0.08	30	1	11
84	Failed	00:04:00	0000	0.08	30	1	11
85	Failed	00:05:00	0000	0.08	30	24	11
86	Failed	00:06:00	0000	0.08	30	10	11
87	Failed	00:07:00	0000	0.08	30	26	11
88	Failed	00:08:00	0000	0.08	30	21	11
89	Failed	00:01:00	0000	0.08	0	21	12
90	Failed	00:02:00	0000	0.08	0	4	12
91	Failed	00:03:00	0000	0.08	0	3	12
92	Failed	00:04:00	0000	0.08	0	16	12
93	Failed	00:05:00	0000	0.08	0	17	12
94	Failed	00:06:00	0000	0.08	0	7	12
95	Failed	00:07:00	0000	0.08	0	27	12
96	Failed	00:08:00	0000	0.08	0	34	12

Appendix E
Experiment 1 Simulator - Python Source Code

```
import mysql.connector
import datetime
from array import *
import random
from output import *
############################## Markov Chain Function ########################
number_of_attack=0
decision=0
previsitedlocation=0
def fetch_location_pairs(id,x,y):
  location_rate =
[[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]]
conn=mysql.connector.connect(user='root',password='root',host='localhost',dat
abase='uc')
  cursor=conn.cursor()
  # Title    Time     Date     Location      [[TaskTime(1,23) iCalendar
Type]]
  query = ("SELECT h.Previsited_location,h.LocationlocationID FROM
HistoryLog3imposter h WHERE logStatus='Success' AND UseruserID="+str(id)+"
ORDER BY Previsited_location,LocationlocationID;")
  cursor.execute(query)
  tmp=[]
  for row in cursor.fetchall() :
      tmp.append ([row[0],row[1]])
  conn.close()
  for i in range (0,len(tmp)):
      location_rate[tmp[i][0]][tmp[i][1]]+=1
  return(location_rate)
#### Searching in the Fuzzy Inference Engine (inside the output.py)###
def search (a,b,c,d):
 if b>0.9:
     b=0.9
 for i in range(0,len(ruls_heap)):
    temp=ruls_heap[i]
    if temp[0]==a and  temp[1]==b and  temp[2]==c and  temp[3]==d:
        return temp[4]
        break;
############## Adds a tuple to the history log ######################
def
add_log(status,j,day,parameters_mrg,trustval,credential,previsited_location,d
esiredLocation,userid):

conn=mysql.connector.connect(user='root',password='root',host='localhost',dat
abase='uc')
    my_cursor=conn.cursor()
    #Add only the success location as a previsited location
    if status=="Success":
```

```
    previsitedlocation= desiredLocation
    add_historylog ="INSERT INTO HistoryLog3imposter
(logStatus,logDate,logTime,parameters4,trust_val,moreCredential,Previsited_lo
cation,LocationlocationID, UseruserID) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s)"
    tim=j+1
    tim='00:0'+str(tim)+':00'
    dat='2016-07-'+str(day+1)
    global previsitedlocation
    #print (trustval)

my_cursor.execute(add_historylog,(status,dat,tim,parameters_mrg,trustval,cred
ential,previsited_location,desiredLocation,userid))
    conn.commit()
timetemp_today_cal=array('i',[0,0,0,0,0,0,0,0])
timenowsend=""
desiredLocation=0
##########################Trust_Algorithm#####################################
def check_context(contx,current_time,timetemp_today_cal):
    t_flag=0
    if contx :
        for i in range(0,len(timetemp_today_cal)):
            #print(timetemp_today_cal[i])
            if timetemp_today_cal[i]==contx:
                if current_time==(i+1):
                    t_flag=1
    if t_flag==1:
        return (1)
    else:
        return (0)
##########################Trust_Algorithm#####################################
##########################Initial Trust Value#################################
trust=0
############################## Begin Table _ View ############################
def view_table(sdate,ididi):

conn=mysql.connector.connect(user='root',password='root',host='localhost',dat
abase='uc')
    cursor=conn.cursor()
    # Title    Time     Date     Location      [[TaskTime(1,23) iCalendar
Type]]
    query = ("SELECT c.Title,c.Time,c.Date, c.LocationlocationID, c.taskTime
FROM Calendar3 c WHERE Date ='"+sdate+"' AND UseruserID="+str(ididi)+"; ")
    cursor.execute(query)
    tmp=[]
    for row in cursor.fetchall() :
        tmp.append ([row[0],row[1],row[2],row[3],row[4]])
    conn.close()
    return(tmp)
############################## End Table View ################################
```

```
############################## Begin Table _ Preferences ##################
def view_preferences(uid):

conn=mysql.connector.connect(user='root',password='root',host='localhost',dat
abase='uc')
    cursor=conn.cursor()
    # location     probability percentage
    query = ("SELECT c.LocationlocationID,c.probabilityRatio FROM
UserPreferences c WHERE UseruserID="+str(uid)+"; ")
    cursor.execute(query)
    tmp=[]
    for row in cursor.fetchall() :
        tmp.append ([row[0],row[1]])
    conn.close()
    return(tmp)
############################## End Table Preferences ####################
############################## Begin Table _ AmbientObjects ##############
def view_AmbientObjects(uid):
    tmp2=[]

conn=mysql.connector.connect(user='root',password='root',host='localhost',dat
abase='uc')
    cursor=conn.cursor()
    query1 = ("select a.aoDescription,a.aoType,a.aoGeneralID from
AmbientObjects a where UseruserID="+str(uid)+"; ")
    cursor.execute(query1)
    for row in cursor.fetchall() :
        tmp2.append ([row[0],row[1],row[2]])
    conn.close()
    return(tmp2)
############################## End Table View ################################

########################## convert ###########################################
########################** Background Map Fucntion ***********################
########### Reduce number of characters of Event##############################
def convert_txt(x):
    output=str(x)
    output=output[:8]
    text22 = font.render(output, True, black)
    return (text22)
######################### convert ent ########################################
def date_convert(x):
    date_tmp=str(x)
    return date_tmp[:10]
#####################################################################
 ################### Starting The Main Iterations#####################
#####################################################################
for ididi in range (1,80):
 var=0
 sdate=datetime.datetime(2016, 7, 1)
 #vt=view_table(date_convert(sdate),ididi)
 #print("Date: ",date_tmp[:10])
 my_row=[]
```

```
 xm=0
 for day in range(0,1):
    #vt=view_table(date_tmp[:10],ididi)
    my_row.append(view_table(date_convert(sdate),ididi))
    sdate +=datetime.timedelta(days=1)
    #print("++++++++++++++++++++++++++++++"+"/n")
    #print("++++++++++++++++++++++++++++++"+"/n")
    #print(my_row[xm])
    xm=xm+1
    #print (vt[0],vt[1],vt[2],vt[3],vt[4],vt[5],vt[6],vt[7],"H"+"/n")
    ##!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!#
    #//////////////////////Threshold Value//////////////////////////////#
    ##!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!#
    threshold_value=0.24
 #This list contains all history log of the user with ID = x
 history=[]
 tot=0
 previsitedlocation=0
 htrust=0.0
 for day in range(0,1):
    my_row.append(view_table(date_convert(sdate),ididi))
    sdate +=datetime.timedelta(days=1)
    if my_row[day]:
        # Hour counter, initial value
        hr_local=1
        for j in range(0,8):
        #Print(my_row[day][j][1])   time, can be zero for name, and 3 for
date
        #Title    Time     Date     Location      [[TaskTime(1,23) iCalendar
Type]]
        #Print("@@@@@@@@@@",hr_local,"+++++++++",desiredLocation)
        desiredLocation=random.randint(1,35)
                #ctrust input value of calendar
                ctrust=0
    ############################################################
    ############ Parameter (1) = (Calendar) ###################
    ############################################################
                #if my_row[day][j][2].day == my_row and
my_row[day][j][3]==desiredLocation:
                if my_row[day][j][3]==desiredLocation:
            # set high value to ctrust
                    if my_row[day][j][4]==1 :
                        ctrust=0.9
            # set medium value to ctrust
                    elif my_row[day][j][4]==2 :
                        ctrust=0.5
            # set low value to ctrust
                    elif my_row[day][j][4]==3 :
                        ctrust=0.2
```

```
#########################################################
############ Parameter (2) = (User Preferences)#########
#########################################################
        # Secondly, Check the preferences and find the input value
            prf=view_preferences(ididi)
            #ptrust input value of preferences
            ptrust=0
            for k in range(0,len(prf)):
                if prf[k][0]== desiredLocation :
                    ptrust=prf[k][1]
#########################################################
############ Parameter (3) = (Ambient Objects)##########
#########################################################
            # Thirdly, Check ambient objects which will assign one
              object randomly.
            atrust=0
            #1
            ao= view_AmbientObjects(ididi)
            #print(ao)
            #2
            if len(ao)>2:
                #ao_in=random.sample(range(len(ao)), 2)
                valrnd=random.randint(0,len(ao)-1)
                #print valrnd
                ao_in=ao[valrnd]
            else:
                ao_in=ao[0][0]
            #ao_in=ao
            #3
            #print ("&&&&",ao_in[1])
            for ki in range(0,len(ao_in)):
                #print ("Hello",ao_in[0][1])
                if ao_in[1] == 1:
                    if atrust<0.9 : atrust=0.9
                elif  ao_in[1]==2:
                    if atrust<0.1 : atrust=0.1
                elif ao_in[1]==3:
                    if atrust<0.6 : atrust=0.6
```

```
    ##########################################################
    ############ Parameter (4) = (History Log) #############
    ##########################################################
                    total_history_weight=0

xarray=fetch_location_pairs(ididi,previsitedlocation,desiredLocation)
                #print
("Previsited_Location,",previsitedlocation,desiredLocation,xarray[previsitedl
ocation][desiredLocation])
                if xarray[previsitedlocation][desiredLocation]>0 :
                  for i in range (0,36):
total_history_weight=total_history_weight+xarray[previsitedlocation][i]
              #print ("Total weight",total_history_weight)
                     #print
(xarray[previsitedlocation][desiredLocation],total_history_weight)
                  if total_history_weight!=0:
htrust=float(xarray[previsitedlocation][desiredLocation])/float(total_history
_weight)
                       #print ("History
Trust",float(xarray[previsitedlocation][desiredLocation]),float(total_history
_weight),htrust)
                  else:
                      htrust=0
                else:
                   htrust=0

    ##########################################################
    ################## Rounding the parameters weights ######
    ##################  to be stored in the DB         ######
    ##########################################################
    ###################

                parameters_mrg=""
                cal_input=round(ctrust,1)
                his_input=round(htrust,1)
                amb_input=round(atrust,1)
                prf_input=round(ptrust,1)
                b1=str(cal_input)
                b2=str(his_input)
                b3=str(amb_input)
                b4=str(prf_input)
                parameters_mrg=b1[2]+b2[2]+b3[2]+b4[2]
```

```
    ##########################################################
    ################## Check the threshold value ############
    ##################+ The Attack Can Be Adjusted +##########
    ##########################################################
    ###/////////////////
                #print (parameters_mrg)

                random_passcode_value=random.randint(1,10)
                #**********************
                # ******30% attack******
                #**********************
        # It is disabled now
                if random_passcode_value>11:
                     # add new tuple to the history table
                     trustval=search(0,0,0,0)
                     if trustval>=threshold_value:

add_log('Success0',j,day,parameters_mrg,trustval,1,previsitedlocation,desired
Location,ididi)
                          decision=1
                     else:

add_log('Failed1',j,day,parameters_mrg,trustval,1,previsitedlocation,desiredL
ocation,ididi)
                          decision=0

                else:
                     # add new tuple to the history table with faild status
                     trustval=search(cal_input,his_input,0.0,prf_input)
             if trustval==null:
              trustval=0;
                     if trustval>=threshold_value:

                         add_log('Success',j,day,parameters_mrg,trustval,1,pre
                         visitedlocation,desiredLocation,ididi)
                     else:

                         add_log('Failed',j,day,parameters_mrg,trustval,1,prev
                         isitedlocation,desiredLocation,ididi)
                          decision=0

      # Instance tracing of the four parameters, trust value, and the status

                         print(round(ctrust,1),round(ptrust,1),round(atrust,1)
                         ,round(htrust,1),"*+-+-+-+-*/n",trustval,decision)

                # Set value to the output variable(status)
                #print ("e")
                hr_local = hr_local+1
      ####\\\\\\\\\\\\\\\\\\\***
```

Appendix F
Accuracy, Specificity and Sensitivity of Different Thresholds

Threshold	TP	FN	TN	FP	Accuracy	Sensitivity	Specificity
0.1	558	74	586	46	0.91	0.88	0.93
0.12	521	111	588	44	0.88	0.82	0.93
0.14	521	111	588	44	0.88	0.82	0.93
0.16	521	111	588	44	0.88	0.82	0.93
0.18	521	111	588	44	0.88	0.82	0.93
0.2	521	111	588	44	0.88	0.82	0.93
0.22	521	111	588	44	0.88	0.82	0.93
0.24	521	111	588	44	0.88	0.82	0.93
0.26	322	310	622	10	0.75	0.51	0.98
0.28	322	310	622	10	0.75	0.51	0.98
0.3	322	310	622	10	0.75	0.51	0.98
0.32	322	310	622	10	0.75	0.51	0.98
0.34	322	310	622	10	0.75	0.51	0.98
0.36	10	622	622	10	0.50	0.02	0.98
0.38	322	310	622	10	0.75	0.51	0.98
0.4	322	310	622	10	0.75	0.51	0.98
0.42	322	310	622	10	0.75	0.51	0.98
0.44	322	310	622	10	0.75	0.51	0.98
0.46	322	310	622	10	0.75	0.51	0.98
0.48	322	310	622	10	0.75	0.51	0.98
0.5	6	626	632	0	0.50	0.01	1.00
0.52	6	626	632	0	0.50	0.01	1.00
0.54	6	626	632	0	0.50	0.01	1.00
0.56	6	626	632	0	0.50	0.01	1.00
0.58	6	626	632	0	0.50	0.01	1.00
0.6	6	626	632	0	0.50	0.01	1.00
0.62	6	626	632	0	0.50	0.01	1.00
0.64	6	626	632	0	0.50	0.01	1.00
0.66	6	626	632	0	0.50	0.01	1.00
0.68	6	626	632	0	0.50	0.01	1.00
0.7	6	626	632	0	0.50	0.01	1.00
0.72	6	626	632	0	0.50	0.01	1.00
0.74	6	626	632	0	0.50	0.01	1.00

0.76	0	632	632	0	0.50	0.00	1.00
0.78	0	632	632	0	0.50	0.00	1.00
0.8	0	632	632	0	0.50	0.00	1.00
0.82	0	632	632	0	0.50	0.00	1.00
0.84	0	632	632	0	0.50	0.00	1.00
0.86	0	632	632	0	0.50	0.00	1.00
0.88	0	632	632	0	0.50	0.00	1.00
0.9	0	632	632	0	0.50	0.00	1.00
0.92	0	632	632	0	0.50	0.00	1.00
0.94	0	632	632	0	0.50	0.00	1.00
0.96	0	632	632	0	0.50	0.00	1.00
0.98	0	632	632	0	0.50	0.00	1.00
1	0	632	632	0	0.50	0.00	1.00