INTELLIGENT HOME AUTOMATION SECURITY SYSTEM BASED ON NOVEL LOGICAL SENSING AND BEHAVIOUR PREDICTION

by

Arun Cyril Jose

Submitted in partial fulfilment of the requirements for
PhD (Computer Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

September 2017

SUMMARY

INTELLIGENT HOME AUTOMATION SECURITY SYSTEM BASED ON NOVEL LOGICAL SENSING AND BEHAVIOUR PREDICTION

by

Arun Cyril Jose

Supervisor: Prof. Reza Malekian

Department: Electrical, Electronic and Computer Engineering

University: University of Pretoria

Degree: PhD (Computer Engineering)

Keywords: Home automation, Smart homes, Identity management systems, Security, Access control, Wireless sensor networks, ZigBee, Data security and Intrusion detection.

The work analyses various researches in existing home automation security or smart home security and addresses its vulnerabilities and shortcomings. The work proposes device fingerprinting techniques to successfully identify user devices accessing the home over the internet. The proposed algorithm is an improvement on existing device fingerprinting algorithms and its device identification capability.

The work proposes a novel logical sensing algorithm to identify and distinguish between normal and attack behaviour at primary and secondary access points in a home. It also developed logic based defensive strategies against tech savvy intruders. The work goes on to propose novel behaviour prediction security algorithm to identify the legitimacy of the user entering the home in a timely manner based on his behaviour inside the home. The work also demonstrates the efficiency, effectiveness and feasibility of the developed algorithm by developing the hardware, testing and implementing them in a home.

PEER REVIEWED JOURNAL ARTICLES

1. A.C Jose, R. Malekian, “Smart Home Automation Security: A Literature Review”, ***Smart Computing Review***, Vol. 5, No. 4, pp. 269-285, August 31, 2015.

2. A.C Jose, R. Malekian, N. Ye, "Improving Home Automation Security; Integrating Device Fingerprinting Into Smart Home", ***IEEE Access***, vol. 4, October 2016.

3. A. C. Jose, R. Malekian, "Improving Smart Home Security: Integrating Logical Sensing Into Smart Home," in ***IEEE Sensors Journal***, vol. 17, no. 13, pp. 4269-4286, July1, 2017.

4. A. C. Jose, R. Malekian, "Improving Smart Home Security: Integrating Behaviour Prediction Into Smart Home," International Journal of Sensor Networks, [Under Review].

LIST OF ABBREVIATIONS

APT	Advanced Packaging Tool
BTLE	Bluetooth Low Energy
CFL	Compact Fluorescent Lamp
CPU	Central Processing Unit
DTMF	Dual Tone Multi Frequency
FSR	Force Sensor Resistance
GoogleAPI	Google Application Program Interface
GPRS	General Packet Radio Service
GPU	Graphics Processing Unit
GPS	Global Positioning System
GSM	Global System for Mobile
HACS	Home Appliance Control System
HAP	Home Automation Protocol
HMMs	Hidden Markov Models
HSSIN	Home Security System on Intelligent Network
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HTML 5	Hypertext Markup Language 5
ICMP	Internet Control Message Protocol
IDE	Interactive Development Environment
IDS	Intrusion Detection System
IHASS	Intelligent Home Automation Security System
ICSP	In-Circuit Serial Programming
IOT	Internet of Things
IR	Infra-Red
IVM	Identity Verification Mechanism

JDK	Java Development Kit
LDR	Light Dependent Resistor
LED	Light Emitting Diode
M2M	Machine-to-Machine
MD5	Message-Digest algorithm 5
MIME	Multi-Purpose Internet Mail Extensions
NFC	Near Field Communication
OS	Operating System
OTP	One Time Password
PC	Personal Computer
PDA	Personal Digital Assistant
PIR	Passive Infrared
PTF	Polymer Thick Film
PWM	Pulse Width Modulation
RF	Radio Frequency
RFID	Radio Frequency Identification
SD Card	Secure Digital Card
SIM	Subscriber Identity Module
SMS	Short Messaging Service
SSH	Secure Shell
TCP	Transmission Control Protocol
UA	User Agent
UI	User Interface
WebGL	Web Graphics Library
WSN	Wireless Sensor Networks
XSS	Cross Site Scripting
ZC	ZigBee Coordinator

ZED	ZigBee End Device

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 PROBLEM STATEMENT

1.1.1 Context of the problem

The concept of Home Automation was a topic of interest in the Academic arena since the late 1970s, with time and advancement of technology people's expectations about Home Automation and how they should access their home has dramatically changed. The affordability and popularity of electronic devices and internet were contributing factors to this change. The modern Home Automation System is a delicate balance of Ubiquitous Computing Devices and Wireless Sensor/Actor Networks. The added expectations and 'Convenience of Access' has brought new security challenges to the Home Automation front.

Nowadays most smart homes are connected to the internet; it is one of the main features which adds to the convenience of the smart home users. Connecting the home to the internet allows a home owner to control their home from anywhere in the world but it also opens up the system to attackers from around the globe who otherwise had to be in the range of the home's wireless network to launch an attack. From an attacker's perspective, compromising a home automation system from the comforts of their home is far more attractive than physically being near the house and trying to break in.

Most obvious way to improve security would be to deny access to a home over the internet, but that significantly inconveniences the home inhabitants and the way they access their home and services, this defeats the purpose of Home Automation Systems. So, securing access to a Home over the internet is a vital part in Home Automation Security. This could be established by limiting access to a home over the internet; Access should be limited to a fixed number of trusted people using a fixed number of trusted electronic devices. To achieve this, we have to identify the user as well as the device accessing the home over the internet.

Home automation networks present a new set of security challenges unlike other wireless sensor networks. Here, the attacker needs devices in the home automation network to respond to their commands in order to achieve their goals. So unlike other sensor networks, simply disrupting the network communication and preventing individual devices from communicating with each other or with the controller is not the type of attack that is expected in a smart home environment. Home automation networks are an attractive target for an intruder, as most of the existing defensive methods in wireless sensor networks mainly focus on network and routing level attacks.

Ideal way to improve home security and defend against intrusion is to recognize a home's authorized inhabitants and identify their position inside a home at all times without inconveniencing its inhabitants. This is extremely challenging and complex, given the unpredictable nature of human behaviour and home being occupied by guests and other trusted people. Identifying access points to a home and regulating access to them is the next logical step towards securing a home. The work proposes that, normal user behaviour at access points to a home adhere to a set of predictable behaviours. These user behaviours when analysed by our novel logical sensing algorithms can differentiate between normal and attack behaviours. The work later goes on to propose a novel behaviour prediction algorithm to identify legitimate users accessing the home in a timely manner to react to unauthorized access, thus improving home security with very little inconvenience to the user.

Most advanced smart homes today are interested in predicting user behaviour inside the home to increase efficiency but little importance is given to incorporating logic or behaviour prediction into home security. Considering these issues and the ineffectiveness of the existing defensive techniques in the smart home environment, we propose a new defensive approach based on user behaviour and logic called Intelligent Home Automation Security System (IHASS).

1.1.2 Research gap

Researches [1] showed that wireless sensor network deployed in home is vulnerable to various routing attacks like Sinkhole attack, Selective forwarding attack, Sybil attack, and

Cloned ID attacks. The work of S. Zhong et al. [2] detected wormhole attack on wireless sensor networks in which the attacker records data packets in the network at one location and tunnels them to another location and retransmits them into the network. This attack can be carried out even if all communications in the network is done with confidentiality and integrity using IP sec [3] in 6LoWPAN [4]. In 2013, S. Raza et al. [5] proposed a real time Intrusion Detection System (IDS) called SVELTE for Internet of Things (IoT). SVELTE could defend against Sinkhole attacks, Selective forwarding attacks, Sybil attacks, Clone ID attacks and Wormhole attacks.

The SVELTE IDS proposed by S. Raza et al. [5] uses a distributed approach which use both signature based and anomaly based filtering to defend against most of the routing and network based intrusion attempts. But when we consider a home automation network, a distributed defensive approach may not always be applicable, as there are many devices with specific functionality and very limited processing power. Moreover all the manufacturers of the devices used in the home may not adopt the SVELTE IDS. In a home automation environment the chance of a routing based or network based attacks are less probable (but by no way impossible), as an attacker has to get within the proximity of the house to implement these attacks (which most cyber criminals try to avoid); from an attackers point of view he/she needs access to a person's house or personal information in order to achieve their goal. So simply disrupting a home automation system's network doesn't make any logical sense and doesn't serve the attacker's purpose. So the need for a different kind of defensive approach for home automation is imminent.

In 2011, Joshua Wright [6] showed how a ZigBee or 802.15.4 wireless networks can be hacked using replay attacks. During reflashing the new key is sent in plain text over the air. An attacker can take advantage of this and sniff (for encryption keys in plain text), inject, decode and alter data packets to manipulate a device's operations. Researchers [7] further demonstrated a vulnerability in Z-Wave door locks which gave the attacker full access without proper authorization.

T. Oluwafemi et al. [8] showed how a simple device in a home, such as Compact Fluorescent Lamp (CFL) which is connected to a home automation network or internet could be manipulated to cause physical harm (shattered glass, fire outbreak, mercury poisoning) to a home's inhabitants. Moreover, lights fluctuating at certain frequencies could be very dangerous for people with photosensitive epilepsy [9]. When the home automation network is connected to the internet there is a possibility that an attacker could gain control of switches and dimmers along with devices plugged into the power outlets. The researchers further discussed the presence of some well-known vulnerabilities in home automation systems, such as, Cross Site Scripting (XSS) [10], they were able to embed persistent Java Script in the log pages of one of the products. The researchers also observed that, in some home automation systems every communication between home owner and home automation system both from within the home network and over the internet is done in clear text (over the internet HTTP is used instead of HTTPS). This allows an attacker to eavesdrop on the communication and gather legitimate login credentials. In some home automation systems a user is authenticated using an authentication cookie which is not associated with any session ID or expiration time frame, so if an attacker could steal this authentication cookie from a legitimate user and include it in their browser session they could bypass the authentication page all together.

In their research, Tamara Denning [11] states why home automation systems are such a potentially attractive target for an adversary. When a home automation system is compromised the data available to an adversary is almost all the time very personal and intimate information about a home's inhabitants, home automation systems doesn't have a dedicated system administrator, the devices at home are usually used by all its inhabitants who may not be tech savvy or security conscious.

The above researches shows how different components of a home automation system can be exploited by a smart attacker and be used to gain access, gather information or cause physical harm to its inhabitants. The researchers also demonstrates how attractive targets smart homes can be to an attacker. All these works demonstrate the security flaws but doesn't provide any comprehensive or effective defensive approaches. Our proposed methodology takes into

account inhabitant behaviour and utilises logical sensing to identify and defend against intrusion attempts.

1.2 RESEARCH OBJECTIVE AND QUESTIONS

The objectives and research questions are discussed below.

- Successfully identify a device accessing the home over the internet using Device Fingerprinting. Successfully identify a user accessing the home over the internet using his/her login credentials.
- Identify legitimate user even when there are changes in location, browser or other browser specific features, which happens over time.
- Identify malicious devices and create a ‘blacklist’ consisting of fingerprints of those devices that will not be allowed access to home. Identify legitimate devices and develop a ‘whitelist’ consisting of fingerprints of devices that are allowed access to the home.
- Distinguish between primary and secondary access points in a home based on how they are used. Detect all user actions at these access points.
- Understand user behaviour after change in state of an access point.
- Identify and isolate attack behaviour by analysing the user behaviour at various access points using our logical sensing algorithm. Trigger warning or raise alarms depending on the situation.
- Identify patterns in user’s behaviour so parameters for behavioural prediction algorithm can be identified.
- Identify when and where behavioural prediction algorithms has to be implemented to successfully identify user behaviour.
- Differentiate legitimate user behaviour from intrusion attempts in a timely manner and activate defensive measures to defend against intrusions.

1.3 APPROACH

The first step was to identify the relevant literature corresponding to home automation security, device fingerprinting, and behaviour prediction, proceeded by analysing the literature and identifying shortcomings in the existing defensive approaches to smart home

security. The research progressed by recognising device fingerprinting parameters and developing device fingerprinting algorithm to improve smart home security.

As the next step we identified when and where behaviour prediction can be used and developed algorithms for predicting user's behaviour patterns from a security point of view. The work moved forward by choosing logical sensing parameters and developing the logical sensing algorithms. Next step is to check the efficiency, effectiveness and feasibility of our developed algorithms by developing the hardware and implementing the proposed algorithm in a home. Finally, a thesis was drafted based on the work.

1.4 RESEARCH GOALS

It is hypothesized that, logical sensing can be used to identify attack behaviours in a smart home environment; behaviour based defensive approaches developed by observing user habits can identify and prevent intrusion attempts in automated homes; device fingerprinting can be used to identify the device accessing the home over the internet.

To develop Device Fingerprinting algorithm to identify devices accessing the home over the internet: The objective here is to develop device fingerprinting algorithms to identify devices accessing the home over the internet. The proposed algorithm should be able to successfully identify and match the devices in a non-intrusive fashion preserving the privacy of the user even when browser updates the add-ons. The developed algorithm improves the identification accuracy of the existing device fingerprinting algorithms.

To develop Logical Sensing algorithms based on the identified parameters: Identify parameters for logic based sensing in smart homes after careful deliberations. Logical sensing means, analysing how probable a sensor value is at any given instant based on logic, when the expected values and the actual sensor values show significant disparity it means someone is manipulating the sensors. Different logical sensing algorithms for each of the parameters considered were developed.

To develop Behaviour Prediction algorithms: When we consider smart home in combination with the behavioural patterns of the people operating them from a security point of view, their operations becomes more or less predictable. People tend to follow routines when it comes to their daily life. Algorithms to observe, identify and learn human

behavioural patterns and routines were developed. Any sudden change from the expected behaviour pattern signifies a possible security threat. After careful considerations, 'where' our behaviour prediction algorithms can be used to generate user behaviour patterns and 'when' the generated behaviour pattern should be used to identify intrusion attempts in a timely manner was also identified.

Experiment, Testing and Evaluations: The device fingerprinting algorithm is verified and evaluated using a website in which the device fingerprinting algorithm was hosted. Logical sensing and behaviour prediction algorithms are implemented and tested using various sensors deployed thought out the home and the results are evaluated.

1.5 RESEARCH CONTRIBUTION

Contribution of the work are listed below.

- Improve the existing device fingerprinting algorithm to identify devices accessing the home over the internet.
- Design and develop novel logical sensing algorithms to identify attack behaviours in smart homes.
- Design and develop novel behaviour prediction algorithms to prevent intrusion attempts in smart homes.
- To the best of our knowledge, our research would be the first to combine the three aspects Device Fingerprinting, Behaviour Prediction and Logical Sensing to develop a comprehensive defensive approach dedicated to smart homes.

1.6 OVERVIEW OF STUDY

The work starts by understanding various approaches to smart home security and the literatures associated with it, the research focus is then shifted to identifying various shortcomings in the existing home automation security approaches, later researches in device fingerprinting, behaviour prediction and logical sensing were analysed in chapter 2. Chapter 3 identifies the parameters necessary for device fingerprinting, behaviour prediction and logical sensing. It also discusses the research methodology used to develop device fingerprinting, behaviour prediction and logical sensing algorithms. Chapter 4 explains the experiment setup and describes how the algorithms are tested in the experiment setup.

Chapter 5 shows the result of the experiment; Chapter 6 discusses the result and compares it with other similar works. The thesis concludes by indicating future directions our work could take.

CHAPTER 2 LITERATURE STUDY

2.1 CHAPTER OBJECTIVES

This chapter explains various literatures and approaches in smart home and home automation security. The chapter is divided into five sections, section 2.1 explains the chapter objectives, section 2.2 details various approaches to home automation security and their pitfalls. Section 2.3 discusses literatures on device fingerprinting and explains various parameters and techniques used by researchers, the chapter goes on to illustrate various logical sensing approaches in smart homes in section 2.4. Finally, section 2.5 discusses the existing behaviour prediction approaches in home automation. In short, this chapter discusses in details various literatures associated with home automation security along with various approaches to device fingerprinting, logical sensing and behaviour prediction.

2.2 LITERATURE ON SMART HOME SECURITY

In this section we discuss various Home Automation methodologies and techniques from a security perspective, we discuss each technology, their features and security pitfalls they have.

2.2.1 Context Aware Home Automation Systems

A modern home can be accessed by its inhabitants from outside through internet/GSM and wireless portable devices like mobile phones, tablets, laptops or through stationary devices like an office work station (PC). In other words, an average automated home user's computing environment keeps on changing; by computing environment we mean, user's type of network connectivity, type of network access at different places, cost of accessing the home over the internet, processing power of the devices and other hardware available to the user. We can't expect the user to be security conscious every time he/she accesses their home from the outside this brings in new security vulnerabilities to the home front. Understanding the context of a particular action by the user could go a long way in improving a home's security. The work of A.K. Dey [12] defines context as "*any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the*

user and applications themselves" In context aware Home Automation System the systems tries to be aware of the context in which a user makes a decision, predicting the location of the user inside the home adds to defining the context. The work of B.N. Schilit et al [13] discusses different techniques for identifying the location of a user at home; the work proposes the uses of Infra-Red Grid to accurately predict the position of the user in the home, thus contributing to improved security but this grid of Infra-Red Sensors are difficult to implement in a home environment; the work also discusses the use of badges to identify the location of inhabitants inside the home, this could significantly improve security but it is inconvenient for the user, moreover inhabitants can be careless in their home and misplace these badges which leads to confusion in the system. Another method mentioned in the work is, 'Static Object Checking'; it identifies an inhabitant's location by checking if someone is in the proximity of a static objects, this method limits the flexibility of the environment and if someone moves or changes the position of these static objects it will be very confusing to the system.

The work of V. Bellotti and K. Edwards [14] explain in detail why contextual sensing is difficult; human aspect or human behaviour is very difficult to predict or reason for, people tend to be unpredictable or unreasonable at times unlike computers they tend to make impulsive decisions about their context and they improvise. Context aware computing raises a serious question of user privacy; with context aware computing the system has even more intimate information about the user, in order to implement context aware computing, system will have to share this information, this raises the questions like; who the information is shared with? How the shared information is used? Who all has access to the shared information? Etc., in short the technology raises more privacy concerns than it solves. In his work S.S. Intille [15] states that, home of the future will not control the environment but instead will help its inhabitants to learn how to control the environment on their won. In other words, taking the power of decision making away from home inhabitants (what context aware computing does) will have a negative impact on their psyche. The author infer that, technologies in smart home should assist the home inhabitants to make energy saving or security conscious decisions by informing or reminding them when such opportunity arises rather than the system making a context aware decisions on its own.

In order for context aware systems to be successful when implemented at home:

- The system must disclose to the user what it knows? How it knows it and what it is going to do about it?
- System must make a user aware when its context based action affects others.
- Context Aware Computing considerably increases a system's complexity. Accurately interpreting the environment is hard enough without the added overhead of predicting the user's context (intention and reasoning for an action).
- Contextual computing increases a system's manufacturing, implementation and maintenance costs which make them unaffordable to the common man.
- In order for the system to be successful it requires constant user interference which most users find annoying. Moreover in case the system malfunctions only an expert or skilled personal can fix the system.

The concept of 'Context Aware' Systems looks good on paper but it is difficult to implement correctly and proposes some serious privacy issues.

2.2.2 Central Controller Based Home Security System

A Central Controller based home security system looks to improve the security of the homes in a locality by combining many homes to form a security network with a control node dedicated for each locality depending on the number of users. These control nodes are controlled by a few Central or Chief Control nodes with considerably high processing power. The security system described by S. Tsai et al [16] called Home Security System on Intelligent Network (HSSIN) uses such a central controller based approach. The proposed system lacks the modern security parameters.

The Central controller based security system has its challenges:

- All or most homes in the neighbourhood have to join in for the approach to be cost effective and successful.
- The main question we have to consider here is; who controls or has access to the central controller and its data? The Central controller will be able to know about a home's intimate and private information like, if a home's room heater is ON or if an

inhabitant in a home is taking a shower etc, from the data available at its disposal, this raises serious privacy concerns. We already know how people feel about government surveillance on the internet; the Central Controller based security systems provides an opportunity to do even more privacy violating surveillance on homes.

The work of K. Atukorala et al [17] proposes a Home Automation Security System called ‘SmartEye’ using GPRS. ‘SmartEye’ also uses a central controller to which many individual home controllers are connected. The system proposes a real time home automation and monitoring system. The system alerts the home owner on his/her mobile phone using GPRS and the user can view the home using the live camera feeds. The system uses ‘RabbitCore’ Module to connect an electrical appliance in the home to the home system (usually a PC); each home system is connected to a central server. ‘RabbitCore’ has an IP address so each device connected to it can be identified and operated via mobile phones using GPRS. The user sends the device management commands to the central server; the home system reads the command from the central server (called 'home polling') and makes the changes needed to a device. When a device changes state, the home system (usually a PC) sends the change of state of the device to the central server, the users mobile will read the change from the central server (called ‘mobile polling’). The mobile user is provided with the home plan and places where each equipment in kept in their home. The proposed research give importance to communication and network setup rather than security, it mentions intrusion detection but no concrete parameters identifying intrusions are mentioned. ‘SmartEye’ uses video cameras for security, its security issues are discussed below. Moreover, like all Centralized Home Security Systems the proposed system is also not ideal for securing a single homes but suits a group of homes, and the author’s claims of ‘increase in poll rate leads to increase in security’ is debatable and misleading.

In reality a Central controller based security system is difficult to implement and raises some very serious privacy concerns.

2.2.3 Bluetooth Based Home Automation System

The work of N. Sriskanthan et al [18] shows the implementation of a Home Automation System using Bluetooth. They use a Host Controller implemented on a PC which is connected to a microcontroller based sensor and device controllers. The researchers even built a new protocol on top of the Bluetooth software stack called Home Automation Protocol (HAP) to make the communication between devices possible. The device controller is connected to the electronic devices through I2C Bus. The system allows more than one device controller to be connected to the Host Controller.

The work of H. Kanma et al [19] also proposes a Home Automation System using Bluetooth which can be accessed remotely through General Packet Radio Service (GPRS). The researchers use a Cell Phone equipped with Bluetooth connectivity as a Host Controller (all home devices communicates with this phone via Bluetooth) and Global System for Mobile (GSM) Modem (provides internet connectivity). Home devices are fitted with Bluetooth communication adapters so that they can communicate with the Host Controller phone via Bluetooth. The paper discusses, remotely controlling and updating home devices along with their fault diagnostics/detection. The work also talks about providing an electronics user manual on the phone using Bluetooth and internet.

Issues of Using Bluetooth for Home Automation:

- Bluetooth has a limited range (maximum range of 100m in ideal conditions) of communication that is needed in a home environment.
- Bluetooth communication has comparatively high power consumption so batteries of devices need to be frequently recharged or replaced.
- Bluetooth technology has advanced and improved to Bluetooth Low Energy (BTLE) which provides the same range of communication (range of 100m in ideal conditions), it has serious security concerns such as ‘eavesdropping’ and breaking the encryption as discussed by M. Ryan [20].
- Bluetooth communication should only be used on occasions where there is a need for quick short-lived network communication with little concern about security.

Bluetooth looks like an attractive communication technology for implementing smart homes; it is cheap, easy and quick to setup, people are already familiar with the technology, hardware required for establishing Bluetooth communication is readily available, technology also provides necessary bandwidth for the operation in home. They also have serious flaws as discussed above.

2.2.4 GSM or Mobile based Home Automation System

Mobile based Home Automation is attractive to the researchers because of the popularity of Mobile phones and GSM technology. We mainly consider the three options for communication in GSM namely SMS (Short Messaging Service) based Home Automation, GPRS based Home Automation and DTMF (Dual Tone Multi Frequency) based Home Automation. Each of these three technologies is discussed below along with their shortcomings.

A. Alheraish [21] proposed the use of GSM module using a SIM (Subscriber Identity Module) to interact with home's sensors, electrical and mechanical devices. The system converts the machine functions into electrical signals through a transducer which goes into the microcontroller. A transducer converts physical quantities like sound, temperature, humidity etc. into some other quantity like voltage; here a sensor does that function. For electronic devices their reading goes directly into the micro controller. The microcontroller analyses these signals and converts into commands that can be understood by the GSM module. Based on the received commands GSM module selects the appropriate communication method (SMS, GPRS or DTMF).

2.2.5 SMS (Short Messaging Service) Based Home Automation System

The work of A. Alheraish [21] proposes a Home Automation System using SMS. The proposed system detects illegal intrusions at home and allows legitimate users to change the passkey for the door and control lights in the home. The illegal intrusion into the home is identified by monitoring the state of the home door, which is done using an LED (Light Emitting Diode) and IR (Infra-Red) sensors. The passkey to the door can be any 4 digits which can be set either by using the keypad or by using SMS from a registered user's mobile

number. A user can control the lights in their home remotely using SMS from their registered mobile number; by turning the lights ON at different rooms at random intervals of time, one can give the impression that the home is occupied even when it is not.

The work of M.S.H Khiyal et al [22] proposes an SMS based Home Security System called 'SMS Based Wireless Home Appliance Control System' (HACS). In their work a home owner can control their home using SMS messages from a pre-set registered mobile number. If the SMS is not from a legitimate user mobile number the system ignores the message. In case of an intrusion the appliance control subsystem and security subsystem in the proposed system informs the owner through SMS.

The work of U. Saeed et al [23] proposes an SMS based Home Automation System. The system has a Java application running on the phone, legitimate users can login to the application using their username and password and can select the building/floor/room/device that they wish to remotely control along with an appropriate action from the list of available user actions. The Java application will compose the appropriate SMS message and will sent it to the home's GSM modem. The GSM modem will receive the SMS message and decodes it and passes it to the home network to perform the action specified. The researchers use 4 digit passkey and facial recognition for security.

In the work of A.R Delgado et al [24] GPRS communication is used as a backup for internet based Home Automation System, this adds to the fault tolerance of the system. The home owner will be able to get alerts on their mobile phone about the unusual state changes in the sensors. The user could then react either by messaging (SMS) or using a web interface (internet). In any case there will be two possible ways to access the home, so if one fails the user can rely on the other.

Security concerns in SMS Based Home Security Systems:

- The 4 digit security passkey (used by A. Alheraish [21] and U. Saeed et al [23]), in itself proposes security vulnerability. An attacker could wait outside the home and peep through the window to learn the passkey; we can't expect the owner to be

careful every time he/she enters the passkey. The user punches in the passkey routinely so the probability of the user being careless is high.

- The passkey used in the work of U. Saeed et al [23] is different for each individual at home which improves the odds for hacking the keypad. Moreover, these passkeys are chosen by users who are vulnerable to 'Social Engineering' and other hacks.
- Most of the proposed systems doesn't consider sophisticated attackers or is no match for a sophisticated attacker; the systems don't consider any other entry points in the home apart from the front door. The LED and IR sensors used by A. Alheraish [21] to identify intrusions could easily be spoofed by a sophisticated attacker.
- Informing the home owner about an intrusion at home through an SMS message is never a good practice; users may not check their phones for SMS messages frequently or may not be near enough to the phone to hear the 'message received tone' so could easily miss the intrusion alert.
- Simple Facial Recognition systems could be hacked using a photograph of an authorized person, as the system can't distinguish between a picture and a real human.
- In today's world attackers are very sophisticated; one should always consider the possibility of an attacker cloning the SIM card of a legitimate device with which the attacker can do all the tasks a legitimate home owner does. Moreover, we can never rule out the possibility of the home owner misplacing his/her registered cell phone or the attacker stealing it. If that happens then 'breaking into a home can be as easy as stealing a cell phone'.
- The researchers A.R Delgado et al [24] point out the increase in security of the home because of remote access, the user can be made aware of an intrusion as soon as it happens so that he can view the home through various cameras installed at different parts of the home. The paper completely ignores the plethora of security vulnerabilities that exist in the devices used to connect and automate a home. Moreover, the chance of an attacker exploiting these vulnerabilities is increased significantly when the home is connected to the internet. In addition, the cameras used in this work have security issues which are discussed below.

2.2.6 GPRS (General Packet Radio Service) Based Home Automation System

There are a lot of home security systems implemented using GPRS. Most systems use the word security in the traditional sense and only address the threat put forth by old fashioned intruders in home.

Researchers M. Danaher and D. Nguyen [25] propose a home security system using GPRS. The work uses a webcam to stream video and pictures of the home to its owner's mobile through GPRS. The webcam detects movement by comparing frames for differences, including light intensity. Video streaming in the proposed work is done using the home internet connection not the GSM modem.

The work of B. Wu et al [26] describes a video camera (which can also take stills) surveillance using the GPRS facility in mobile phones. The camera is triggered when an Intrusion is detected or the door bell is ringed. The system identifies intrusions with an Infrared sensor. In case of a doorbell, the system calls the home owner and establishes a voice communication (along with the live video feed) between the visitor and home owner. When an intrusion is detected an email is sent to the user along with a picture probably of the intruder. Upon receiving this email the user can start monitoring the video feed on his phone.

Work of L. Yang et al [27] allows a user to read and change the status of the devices at home using a preregistered mobile number using GPRS. The proposed system doesn't allow external devices to connect directly to the home devices. When a legitimate device (identified using the device's phone number) tries to connect to the home environment, the connection is established between the virtual home which mirrors the current state of the home devices (acts like a honey-pot) and user. The commands issued by the user are analysed and if they don't pose any harm to the home devices then the command is applied to the real devices in home. When an emergency situation arises (intrusion is detected or Fire outbreak) the intelligent devices at home initiates a communication between the home and the user (via

telephone, text message, email) called 'phone-out-only' (Not the other way around i.e. user never initiates direct communication to home devices).

U. Ali et al [28] proposes another home and office automation system using GPRS in mobile phones. The user interacts with the home by a client/server architecture implemented at home using a PC and a micro Java application. Home devices are controlled by a device controller which is connected to the PC's parallel port. The proposed system allows users to remotely control and enquire the status of the devices that are connected to the device controller.

The researchers J. Jin et al [29] discuss a Home Automation System based on Wireless Sensor Networks (WSN) and GPRS. It allows it users to control equipments in their home, collect data about a device's status and weather conditions at home through their mobile devices. The authors' custom made the application for China as users receive information about home intrusions and fire through Chinese Instant Message Mobile Service. Unlike other GPRS based Home Automation the proposed system uses an embedded system based central controller.

Researchers S.R. Das et al [30] developed an iOS based Home Automation Security System using GPRS. The proposed system uses client/server model for communication. The authors develop an iOS application which runs on user's mobile phones acts as the client and the cloud to which the home devices are connected to acts as the server. The authors use video cameras, microphones, motion sensors for providing security at home. When a motion sensor is triggered the video cameras in the vicinity starts to record; a user can view these live feeds on the mobile device through GPRS. The proposed system can also be accessed using a web browser.

Security concerns in GPRS Based Home Security Systems:

- The works of M. Danaher and D. Nguyen [25], B. Wu et al [26], L. Yang et al [27], S.R. Das et al [30] implement cameras at home. Streaming live video feeds over the internet is never a good idea especially when it is from inside the home; if these

implemented cameras are compromised then the attacker will have an eye inside the home. A recent BBC report by L. Kelion [31] highlights the vulnerabilities in the wireless cameras. Moreover, people do not like to be shrivelled; it affects their normal behaviour and makes them uncomfortable.

- Video feeds could be looped by skilled attackers if the cameras and the system is not installed and maintained properly.
- In GPRS based intrusion detection system the user will have to monitor his/her phone constantly to successfully defend against intrusions.
- The IR sensor based intrusion detection system specified by B. Wu et al [26] can be spoofed by a skilled intruder so its ability to identify intruders can be questioned.
- The work of L. Yang et al [27] uses terms like 'safe', 'do not harm' which have a broader meaning and it is not clearly specified. Moreover, in their proposed system legitimate user mobile number could be spoofed by skilled attackers; so such attackers can manipulate the home devices or at least know the status of the status of the home devices; from the status of different home devices one can infer whether a home is occupied and which all rooms are occupied.
- Alerting a user about an intrusion attempt by email is never a good practice; users may not check their phones for email messages frequently or may not be near enough to the phone to hear the 'message received tone' so could easily miss the intrusion alert.
- Researchers S.R. Das et al [30] provides users access to home using a web browser which opens the home to a different set of browsing related security issues like session hijacking, cookie stealing, Cross-Site Scripting etc.
- The work of M. Danaher and D. Nguyen [25] provides limited security as they only uses cameras and no other security mechanisms.
- Almost all the GPRS based Home Security Systems uses one or more of these devices like, video cameras, motion sensors or Infra-Red sensors to identify intruders in a home environment. They rarely use any complicated techniques or algorithms to identify a more technically skilled attacker.

2.2.7 DTMF (Dual Tone Multi Frequency) Based Home Automation System

The work of L. Muhury and A.H.M.A Habib [32] describes the design and implementation of a DTMF based Home Automation system. The user calls the SIM number assigned to the Home and presses the digits on their phone's keypad to control the home devices by generating the DTMF tone. The tone is received and decoded at the GSM Module at home using a DTMF decoder. The decoded instructions are passed to the microcontroller so that user commands can be implemented at home.

Home Automation systems using DTMF are not very commonly implemented may be because there are other better options for communication available. Like all other systems DTMF based home security systems are also has their security flaws. They are vulnerable to 'Fuzzing Attacks' as described by R. Sasi [33]. In 'Fuzzing Attack' the user exploits the vulnerability in DTMF processing algorithms by giving unusual input data which results in triggering an exception. This could cause the entire Home Network to crash.

2.2.8 Internet Based Home Automation System

Internet or IP protocol based communication in Home Automation System is always a popular choice among researchers. Internet is: easily scalable, flexible when it comes to access and use, very popular as a communication method in today's world so the hardware and the network required for access is readily available, offers high bandwidth and very low communication cost, devices can connect and disconnect to-and-from the network easily. These are some of the features that make internet such an attractive choice for the researchers. Utilizing internet as a means to access and control the homes seems to be the next logical step in the way forward for Home Automation Systems.

From an end user's point of view, using internet to access their home is easy, convenient, cheap, flexible, no complication of an added technology to learn, user interface devices like laptops, Smartphone, PCs, tablets etc. are easily available in the market and these devices are already a part of people's daily life. So, incorporating Home Automation into these

already popular user devices seems to be the natural progression. The figure given below, Figure 3 shows the components of a typical Home Automation System using Internet.

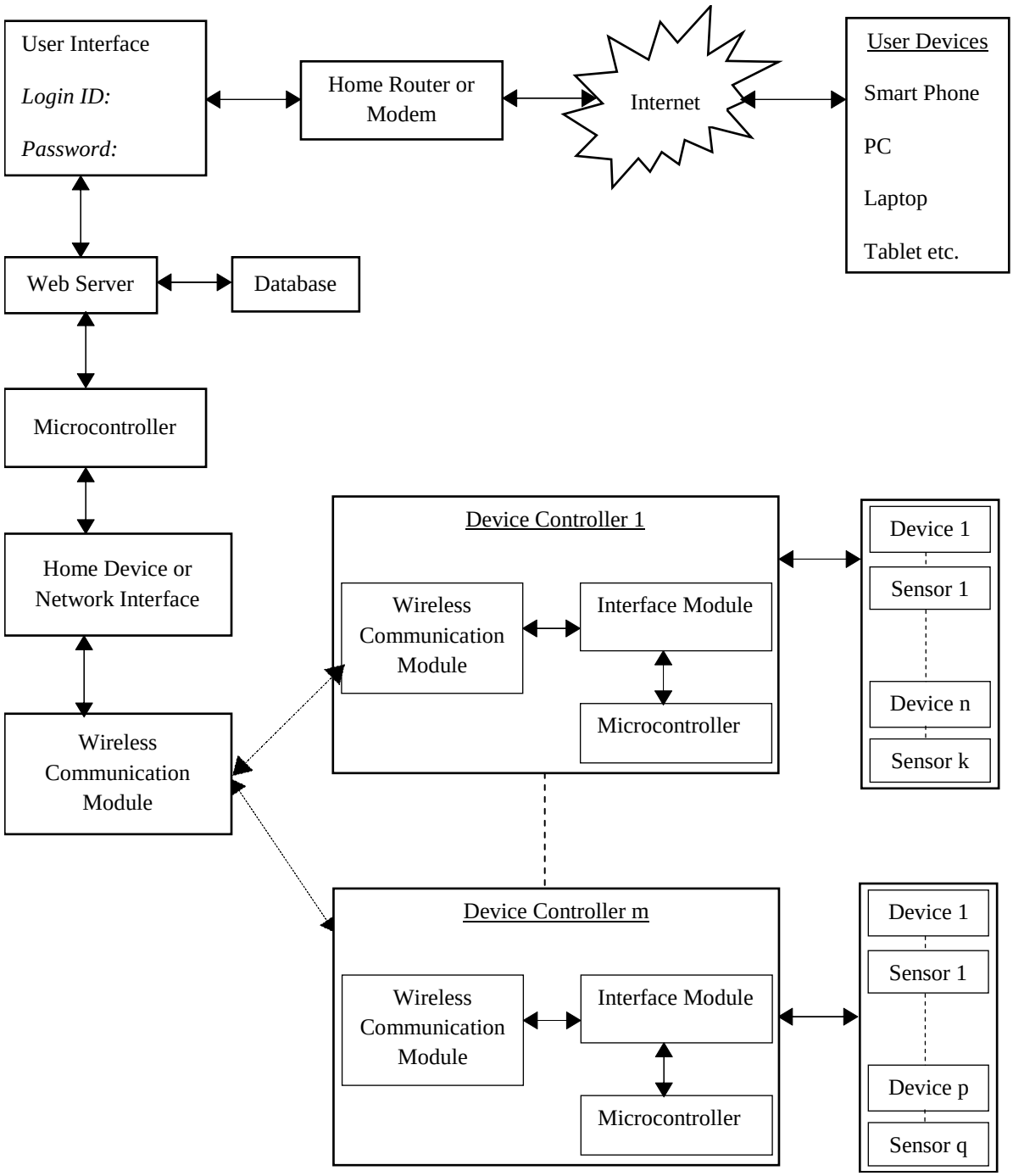


Figure 2.1. Logical Diagram of a typical Internet based Home Automation System

The User Interface (UI): User Interfaces are usually web pages or any Android/iOS/Windows applications developed by the researcher. A user can use these applications or a web browser to access their home from their portable devices using the internet. Most Home Automation Systems uses a ‘username’ and ‘password’ as a way of identifying legitimate users before granting access to the home.

In most Internet based Home Automation Systems username and password seems to be the only authentication method used, this raises some security concerns:

- People are generally careless in nature; they tend to write complicated passwords and usernames on paper near their workstations or underneath their keyboards thinking ‘who bothers to look there’.
- People often repeat the same passwords and usernames on different websites and forums, this behaviour makes them vulnerable to ‘phishing attacks’.
- During the course of time, a home owner will have to login to the home from different networks like from the office, from their friend’s house, from public Wi-Fi networks such as coffee shops, parks etc. sometimes using untrusted devices. The network chosen by the user to access the home may be vulnerable, this results the user being exposed to variety of attacks like ‘Man-in-the-Middle Attack’. Moreover, when accessing the home from a compromised device, legitimate user credentials could be stolen by the use of simple software tools such as ‘Keylogger’.
- Researchers should also be aware of the ‘Human factor’ when depending only on passwords for security. ‘Human factor’ means normal people tend to choose passwords which have some sort of significance to them like their pet’s names, name of their favourite movie, music artist, sports team etc. Moreover, we should never underestimate the most powerful hack of all ‘Social Engineering’ which could prove to be very effective when trying to obtain a person’s password and username.
- Accessing a home through a web browser opens the home up to a variety of browser related security issues mentioned earlier. Researchers have to assume that, when accessing their home over the internet people will choose convenience over security if given the choice.

Web Server, Database and the Microcontroller: The user interface is connected to the database via a web server. The database consists of details of all the home devices and their current status. A user remotely accessing their home can query the device status information from the database via the web server. A Microcontroller manages all the operations and communications in the home network as shown in figure 3. In reality, a PC can do all these tasks, so researchers replace these three components (web server, database and microcontroller) with a PC for convenience.

Network Interface Module: It manages the communication between the PC and the home device controllers. When a user issues commands to change the status of the devices at his home, these commands are transmitted through this interface to the device controllers. Upon completion of these commands the status of a device is relayed to the database through the interface.

Device Controller: A Device Controller consists on an interface module, a wireless communication module and a microcontroller to control its operations. A Device controller is connected to multiple home devices and sensors. User commands and status enquires to a home device are relayed through the device controller.

Communication Module: There are a few choices in technology when selecting the mode of communication between the devices within the home. Depending on the inhabitant's preference a Wired or Wireless connections can be used. For Wired communication X10 is the most commonly used communication protocol as it can be implemented using the existing wiring without a lot of drastic changes. For Wireless communication the choices include Infra-Red (IR), Bluetooth, Wi-Fi, Radio Frequency (RF). Each of these communication technologies has their own pros and cons. We discussed Bluetooth based Home Automation in the previous section. We also discuss a few works based on Infra-Red, Wi-Fi and Radio Frequency communication below.

The work of A.Z. Alkar and U. Buhur [34] implements a Home Automation system using internet for enabling remote home access and Infra-Red technology for device

communication within the home. The researchers use a PC to perform the task of a web server, database and the interface program. They use a RS232 module as a communication interface. The user interface is also developed by the researchers and made it accessible through the internet. The work proposes the use of SSL certificate to ensure the authenticity of the web server.

The SSL certificate proposed by the authors is relatively secure but there are still issues like SSL Certificate Stealing, Certificate Authority Hacking, and Fake Certificate Authority etc. User authenticity is ensured using a username and a password, which is an area of security concern as discussed before.

Wi-Fi communication technology has a lot of advantages; low installation cost, easy to deploy and install, has decent communication range, technology is scalable, high bandwidth, low power consumption, AES encryption offers good security, moreover, repeaters can be used to extend communication range. Wi-Fi is an ideal choice of communication for automating an already existing home without altering the existing architecture besides the communication is wireless so it improves the aesthetics of the home. All these factors make Wi-Fi an ideal choice for wireless communication among researchers. The work of A. ElShafee and K.A. Hamed [35] proposes an internet based Home Automation System. Their work uses Wi-Fi to enable communication between different devices and the server at home. A user can login using a username and password and control the devices at home. In their work, the researchers use a PC as a web server. The PC also has built-in Wi-Fi communication capabilities and a communication module and enables communication between the home devices and the Server (PC). The home devices are connected to the server via Wi-Fi by a hardware interface module. Security is improved in the proposed system by blocking access to a login page for some time after successive failed login attempt. This protects the system from Brute-force attacks, Dictionary based attacks. The system is still vulnerable to browser vulnerabilities and Social Engineering.

The work of J. Shah et al [36] proposes a Home Automation system using RF and SMS. Radio Frequency (RF) module discussed here is used for communication between the home

owner and the devices. The Home owner uses an RF remote to control their home devices; the RF remote used has a range of 20m-30m. When a device changes state an SMS is sent to the home owner alerting him about the change in state.

The proposed system doesn't allow home owners to access their home remotely from their mobile device; they can only control home devices through the RF remote. RF remote used here has very limited range to be successful in a home environment; furthermore the proposed system only allows 15 devices to be connected to the home network.

Internet: A home inhabitant can access and control his/her home from anywhere in the world if the Home Automation System is connected to the internet. It has its advantages as we discussed earlier but connecting the home to the internet opens it up to the world i.e. anyone with an internet connection can try to access the home. An attacker can search for known vulnerabilities and zero day exploits belonging to a particular device from a specific manufacturer from anywhere in the world. If the home was not connected to the internet attackers had to be in the proximity of the home to exploit the vulnerabilities; in other words the probability of attack decreases considerably but it defeats the whole purpose of automating the home.

End User and their Portable Devices: Advancement in electronics, processing power and considerable reduction in size and cost allows people to own and frequently use mobile portable devices for accessing the internet. So people uses these same portable devices like Smartphone, Laptops, and Tablets etc. to access their homes via the internet. This allows a flexible and convenient way for home inhabitants to access their home on the go. They uses the same device to access other applications and do their daily tasks like browsing, playing games, installing apps., watching movies etc. with the internet. This increases the probability of having a security risk in the mobile device, as we can't expect an average internet user to be security cautious all the time. When a user accesses their home from a compromised mobile device an attacker could easily steal legitimate user credential to access the home. This coupled with known and unknown vulnerabilities in the installed applications and services raises a serious question mark about the portable device's security. Furthermore,

user mobile devices can be stolen by an attacker, so if the user is already ‘logged in’ to his home via a Smartphone app., the attacker can then easily access the home. End user is vulnerable to ‘Social Engineering’ and their choice of user credentials like ‘username’ and ‘password’ can be questioned as discussed before.

In short connecting the Home Automation System to the internet has its advantages and disadvantages. Home Automation System users have to be aware of these security issues and operate their Home Automation System accordingly.

2.2.9 Decentralized Approach to Home Automation Systems

Home Automation Systems discussed so far uses a central controller or centralized approach which has a single point of failure; in this section we discuss another approach. The work of M. Gauger et al [37] proposes a decentralized approach to Home Automation control. They implement the decentralized approach by integrating actuators into the Wireless Sensor Networks (WSN) of the home. The authors propose a distributed control or process architecture. The information from the sensors are received and processed by one or more control nodes, which in turn initiates the appropriate actuators to change or control the environment as previously specified by the user. The system thus eliminates the need for central controller.

Issues in Decentralized Approach for Home Automation Systems:

- A sophisticated attacker with prior knowledge of the network and actuator positions can simply disconnect them from the network. No alert mechanisms are implemented for such a case.
- The communications proposed here are done in clear text, so an attacker with the right hardware can eavesdrop on the communication.
- Home Automation System consist of complicated tasks which requires analysing and processing various values and inputs from different parts of the home during different points of time, it requires some processing power and storage which the actuator nodes can’t provide at present.

Decentralized approach to Home automation is an interesting concept but it requires a lot of work from the research to be efficiently and securely implemented in a home environment. In addition to this, the actuators discussed here require a significant increase in processing power and storage for it to be effective in a decentralized architecture.

2.3 LITERATURE ON DEVICE FINGERPRINTING

This section discusses the existing literatures related to device fingerprinting. The section also discusses various approaches, parameters and techniques used for identifying and retrieving device fingerprinting parameters.

The concept of cookie was introduced into the context of the web browser in 1994, by Lou Montulli [38] [39]. Cookie allowed webservers to store small amount of data on the visiting user's computer which is sent back to the server upon request. The concept of cookie was quickly embraced by browser manufacturers. Soon after, attackers began to take advantage of cookie's state-full nature. Third-party advertising sites used cookies to track users over multiple websites which encouraged behavioural advertising [40]. This privacy violating behaviour caught the attention of the research community [41 - 44], legal community [45] and was a cause of concern among the public [46] [47]. Moreover, cookies are vulnerable to Cross Site Scripting [48] and Cookie Stealing. The concept of cookie was further expanded to Flash cookies [49] and later to 'evercookie' [50] which is almost impossible to remove; this further enhanced the privacy concerns associated with cookies. A cookie-retention study [51] showed that one in three users deleted their first and third-party cookies within a month of visiting a website. The above researches illustrate the privacy, security and unavailability issues associated with using cookies to identify a user. So utilizing cookies in Home Automation to identify a user over the internet doesn't seem like a sensible decision.

The issues associated with cookies prompted researchers and internet advertisers to come up with a new way to tracking internet users. In 2009 Mayer [52] and in 2010 Eckersley [53] demonstrated how features of a web browser can be used to uniquely identify a user without cookies over the internet. Mayer [52] did a study on 1328 web clients. In his study, he hashed the combined contents of navigator, screen, navigator.plugins and navigator.mimeTypes.

Using this, he uniquely identified 96% of the web browsers in his study. Eckersley [53] conducted a study on 500,000 users and uniquely identified 94.2% of them. He combined various properties of the web browser and installed plugins, to uniquely identify users. He used Flash and JavaScript to collect the required information from the client machine. In 2012, Yen et al. [54] conducted a fingerprint study on month long logs of Bing and Hotmail by using User Agent (UA) string and client's IP address. The authors were able to identify 60 to 70% of users by just using UA string and the accuracy improved to 80% when IP prefix information was combined with the UA string. Eckersley [53] dismisses the use of IP address for fingerprinting as they are "not sufficiently stable". Now a days, internet users try to mask the UA string of their web browser to avoid identification, but the work of N. Nikiforakis [55] illustrated how counterproductive this is and demonstrated spoofing UA string aids in user identification which is contrary to the popular user belief. So, in our work IP prefix was avoided but UA string was utilized for device identification.

Mowery et al. [56] proposed a device fingerprinting method exploiting the difference in JavaScript performance profiles among different browser families. Each browser executes a set of predefined JavaScript bench marks and the completion time of each bench mark forms a part of the performance signature of the browser. Using this technique, the authors were able to successfully identify a browser's family 98.2% of the time; the identification process took over 3 minutes to completely execute. A study [57] shows that, an average user views a web page for about 33 seconds. So, fingerprinting based on JavaScript benchmark execution time may not be the solution as it takes too long to identify a client. Moreover, accuracy and detection rate of more specific device fingerprinting attributes such as, operating system, browser version and Central Processing Unit (CPU) architecture is significantly low. The work of Mowery [56], also demonstrated how selective enabling/disabling of JavaScript using browser plugins (like 'NoScript' in Firefox) for certain websites could aid in fingerprinting and subsequently helps user identification. Disabling JavaScript completely in the browser would be the way to preserve a user's online privacy but most websites needs JavaScript to function properly. The above research shows, contrary to the popular user belief selectively enabling of JavaScript helps in device identification. This gives a reason for even the most privacy concerned users not to disable

java script, so the fingerprinting algorithm discussed in this thesis utilizes JavaScript for device identification.

In 2012, Mowery and Shacham [58] proposed a device fingerprinting technique based on the hypothesis that different browsers display text and graphics in a different way. This difference raise from a combination of configuration differences in software, browser, driver, hardware and GPU. To exploit this, the authors rendered text and Web Graphics Library (WebGL) scenes into a HyperText Markup Language 5 (HTML 5) <canvas> element and measured the difference in the resulting pixel map of the canvas for different users. The proposed method cannot differentiate between two web clients with the exact same software and hardware configurations and will not work on older versions of a web browser.

T. Kohno et al. [59] proposed device fingerprinting using clock skew. The authors observed that, there is distinguishable clock skew difference between any two physical devices, and this unique clock skew difference between two devices will remain relatively stable over time. They exploited this clock skew feature to fingerprint a remote physical device by stealthily recording and analysing its Internet Control Message Protocol (ICMP) or Transmission Control Protocol (TCP) timestamps. Using ICMP and TCP timestamps has their limitation, ICMP timestamps are blocked by numerous firewalls, and some operating systems by default disable TCP timestamps. Later Zander and Murdoch [60] developed a device identification technique with synchronized sampling which significantly reduces the quantization error. It reduces the heavy network traffic which was necessary for previous identifications, their work was the first to calculate clock skew estimation through Hyper Text Transfer Protocol (HTTP) protocol. However, their approach could not be directly implemented at the server side for device identification. Inspired by this work, D.-J. Huang et al. [61] developed a client device identification in cloud computing scenario, which relay on JavaScript to send periodic timestamp back to the server for device fingerprinting. G. Nakibly et al. [62] proposed a device fingerprinting technique by exploiting the uniqueness of hardware features like, speaker/microphones, motion sensors, Global Positioning System (GPS) accuracy, battery charge and discharge time and Graphics Processing Unit (GPU)

clock skew. Most of their proposed techniques remain purely theoretical at the moment. Moreover, their fingerprinting approach requires constant user interactions, which is not ideal.

Other attempts in device fingerprinting include Operating System (OS) fingerprinting using popular tools like Nmap, Xprobe etc.; device fingerprinting approach discussed in this thesis did not implement OS fingerprinting as most of the firewalls and network administrators prevent this [63] and it requires manual interpretation [64].

Passwords are always vulnerable to brute force [65], dictionary [66] and rainbow-table attacks [67]. A study done by K. Kato et al. [68] among 262 University students revealed that, 80% of the passwords were not strong and 40 % of the passwords were reused for different accounts. This concurs with the work of D. Hart [69] who concludes that 30% of people reused their passwords 4 or more times. The work of J. Yan et al. [70] demonstrated that, average users have difficulty remembering random passwords and people are reluctant to use special characters in their passwords. Various password policies implemented by the administrator further complicates things [71]. So, people tends use grammatical structures in their passwords, the work of S. Rao [72] illustrates security issues associated with such passwords. Passwords are set and has to be remembered by humans, whose memory for sequences of items are temporally limited [73], with a short term capacity of around seven plus or minus two items [74]. Moreover, humans are vulnerable to social engineering [75]. These human errors, human memory limitations and social engineering compounds to the security issues associated with passwords [76] [77]. Moreover, well known password hacking tools such as ‘John the Ripper’ or ‘Hashcat’ also assists an attacker. So passwords alone are not enough to keep access to our homes secure over the internet. The thesis proposes a security system with two stage verification, which utilizes password and device fingerprinting before granting online access to home.

2.4 LITERATURE ON LOGICAL SENSING

This section narrates various literatures associated with logical sensing and their shortcomings.

The research of J. Choi [78] et al. utilized body temperature, pulse, facial expression, room temperature, time and location to predict and learn user context. Their work failed to take into account the fact that, user's body temperature, pulse or facial expression may vary depending on various other factors like state of mind, illness etc. Moreover, their work uses cameras to read facial expressions which when compromised [79] by a tech savvy attacker brings new security and privacy issues to the home.

O. Yurur et al. [80] proposed that context aware sensing vary depending upon user environment, prior knowledge of recent event patterns, user perception and context. In a home environment where people are relaxed, impulsive and unpredictable it is extremely challenging to predict the context of various user actions. This makes context aware computing difficult to implement unless the system has in-depth knowledge of the context, which requires sophisticated sensing techniques and high processing power. Such advanced sensing techniques and high processing power makes context aware sensing an expensive proposition for smart homes. In addition, during context aware computing the system handles very intimate and private information about a user and his habits, which has to be shared for the concept to be implemented successfully, this raises serious privacy issues. The work of S. Saponara [81] demonstrated how an attacker can determine what devices are active in a home by looking at the power consumption at any given time. So it is risky and expensive to implement a completely context based security system in smart homes. So, this thesis does not take into account the context in which the user makes a decision but focuses on user behaviours at various access points.

The work of D. M. Konidala et al. [82] suggested the use of Radio Frequency Identification (RFID) tags for successfully identifying various items inside a smart refrigerator. This technique could be extrapolated to improve home security but it requires most items inside

the home including home inhabitants fitted with RFID tags, which is inconvenient and difficult to implement considering forgetful human nature.

S. Lee et al. [83] used Infrared (IR) grid to identify inhabitant location inside a smart home. Their research used multiple IR sensors deployed on the ceiling to build an IR grid to predict inhabitant location inside a home. Later, H. H. Kim et al. [84] proposed the use of Bayesian classifier to improve the predicted inhabitant location accuracy inside the home. The work of P. Kumar and P. Kumar [85] utilized Arduino Uno microcontroller and IR motion sensor to identify intrusion attempts in a home, when an intrusion takes place the information is transmitted to the user using GPRS to their Personal Digital Assistant (PDA). In the proposed system, user has to be near their phone to be alerted to an intrusion attempt. Moreover, the IR sensors deployed by the researchers can be spoofed by skilled intruders. So, this thesis, not only depends on IR motion sensor but also implements ultra-sonic proximity sensors, force sensors, contact sensors and gas sensors for intrusion detection.

Y. Zhao and Z. Ye [86] proposed a low cost and flexible home security system which alerts the administrator of an intrusion using SMS. Their approach lacks any sophisticated intrusion detection algorithms to identify attack attempts. The work of S. Morsalin et al. [87] suggested a home security system utilizing Near Field Communication (NFC) tag, passwords and fingerprints. The system also has an embedded Global System for Mobile (GSM) module which communicates the logged password to a remote server using Machine-to-Machine (M2M) communication. Each time a user wants to access his home he has to enter the password and verify his fingerprints which is an inconvenience. The NFC tag mentioned in the work could be misplaced by a careless user or stolen by an attacker.
P. H. Huang et al [88] proposed a fire detection and identification method by analysing video, which is expensive to be used in a smart home scenario. It also requires setting up video cameras inside the home which when compromised proves to be a serious threat to inhabitant privacy. The fire detection system discussed in the thesis utilizes temperature and humidity sensors along with gas sensors to identify fire.

2.5 LITERATURE ON BEHAVIOUR PREDICTION

This section details various approaches to behaviour prediction in smart homes, most of literatures here are focused on improving the efficiency of the smart home rather than focusing on security.

Different user identification mechanisms were suggested over the years, it can vary from simple passwords, fingerprint verification [89], retinal scan verification [90], facial recognition [91] using cameras to more complicated vein recognition [92], biometric gait recognition [93] and voice recognition [94]. The user identity verification techniques proposed had some significant security issues [67] [95]. H. Park et al. [96] proposed the use of a sentry sensor to identify the occurrence of an event of interest which acts as the trigger to activate other sleeping sensors and lets them participate in event monitoring. The method was proposed as a way to improve energy efficiency of the sensors deployed in smart homes but it has single point of failure to be successfully implemented in a secure smart home environment. If attackers could identify and manipulate the sentry sensors all the other monitoring sensors will remain inactive and the whole smart home could be compromised.

Inhabitant behaviour prediction was widely used in the assisted living environment for many years now. Researchers utilized vibration sensors to identify a person falling [97] [98] or interaction with various objects, pyroelectric infrared sensors were modified to detect stove and oven operations in the kitchen [97], multiple Ultrasonic sensors were deployed to identify inhabitant locations inside the home [99] and pressure sensors were deployed to detect the presence of a user, steps taken by the user and identify fall events [97] [100]. Light sensor or photo sensors were not extensively used for behaviour prediction but they were used to identify user presence at various parts of the home [100], [101] and as a means of direction measurement [102]. Some approaches used magnetic switches to identify the status of a door or a cupboard at various parts of the home [103]. A. Fleury et al. [104] used microphones to identify various user activities in a home like talking, door closing, walking, phone ringing, objects falling, and TV usage etc. Various researchers [105] – [107] proposed the use of Wattmeter to identify electricity usage in home, which today can be considered as

a major indicator of the wellbeing of an inhabitant. Most of these approaches identifies particular action of a user in a specific part of the home, so it will be inefficient to deploy them to identify intrusion attempts in a smart home. Intruders cannot be always expected to use the kitchen or some other specific part of the home, moreover, most of these proposed approaches takes time to distinguish between normal and attack behaviour which in a security scenario is at a minimal.

Different approaches for activity recognition using wearable sensors were proposed by the researchers over the years. Accelerometer data can be used for activity recognition [108] [109]. J. Wannenburg and R. Malekian [110] proposed the use of smartphone accelerometer data to identify user activities such as jogging, walking, standing, sitting, laying down etc. during observation the smartphone is kept in the user's trouser pocket. Later, they [111] designed optical pulse oximeter sensor and combined it with digital temperature sensor into a wearable unit to measure heart rate, saturation of oxygen, pulse transit time and skin temperature to identify medical stress. J. Merilahti et al. [112] proposed the use of wrist-worn activity detector to identify a user's sleep/awake activity. K. Van Laerhoven et al. [113] combined accelerometer, tilt switches and inertial sensors into a wrist-worn sensing unit to model user's rhythms and model user behaviour. T. Maekawa et al. [114] utilized hand-worn magnetic sensors to identify and distinguish between various magnetic fields emitted by different electrical devices at home and recognized user activity. Implementing wearable sensors to identify user behaviour at home increases the user inconvenience as they have to wear the sensors all the time. Moreover, it will be difficult to successfully implement given the forgetful and careless human nature.

The work of P. N. Dawadi et al. [115] demonstrated that sensors implemented in smart homes can be used to predict user's behaviour inside a home and this behaviour data can be utilized to predict an individual's clinical scores. The paper hypothesized that, there is a relationship between a user's behaviour monitored by the smart home sensors and his cognitive/physical health. The researchers analysed sensor data to identify inhabitant's sleep duration, cook duration, eat duration, relax duration, personal hygiene duration, bed toilet transition duration to calculate the clinical scores. The work of Paavilainen et al. [116] used

IST Vivago WristCare system to monitor circadian activities of older adults living in nursing homes and compared the changes in activity rhythms with the clinical observations to identify their health status. Most of these researches concentrate on identifying health issues of a home's inhabitants with little significance to home security. In the health monitoring scenario, the behaviour of the user is compared with pre-established standard clinical values to identify health issues of a home's inhabitants over time. This approach cannot be used to identify intrusion attempts in a home as often intrusion attempts has to be identified swiftly and precisely. Moreover, there are often very little pre-established standard user behaviours to successfully enable comparisons in an efficient manner.

Chahuara et al. [117] scrutinized various parameters in the home like, sounds made during various chores, location of the home's inhabitants, inhabitant's speech, water consumption, light states, temperature etc. to identify user's daily activities such as sleeping, cooking, resting etc. The system consist of microphones spread throughout the home to pick-up sounds and receive voice commands, these picked up sounds are analysed and archived for learning user behaviour, which can be viewed as violation of home inhabitant's privacy. In their work on automated homes, Emmanuel Munguia Tapia et al. [118] mainly focused on identifying user's daily activities such as sleeping, having meals etc. The proposed system uses supervised algorithms, training data and user interactions to identify and predict user activities to improve a home's efficiency. The proposed approaches primarily focused on improving home efficiency and gave minimal importance to home security. Moreover, they require large computational overhead and requires expensive hardware.

Over the years the world has witnessed a significant improvement in technology which made electronic devices more affordable, popular and improved the efficiency of computer and sensor networks. The sensors and actuators became compact with improved processing power, so we witnessed significant advances in intelligent environments. A lot of research projects and a few field projects were published over the years, these include, University of Virginia's smart assisted living facility [119], University of Missouri's TigerPlace [120], HyperMedia Studio [121], GATOR Tech Smart House [122], assisted smart living by Staffordshire University/Chiang Mai University [123], Interactive Room iRoom [124],

CASAS Smart Home Project at Washington State University [125], the MavHome Smart Home Project at the University of Texas at Arlington [126], and the Smart Medical Home, located at the University of Rochester, [127], the PROACT proposed by Intel Research Seattle [128].

Human behaviour can be learned and predicted through supervised and unsupervised learning. Data annotation is necessary in unsupervised learning, which is a time consuming process. In their work, Kasteren et al. [129] utilized Hidden Markov models (HMMs) for user activity recognition. Doctor et al. [130] proposed the use of fuzzy logic to learn and model activities in the iDorm unsupervised while Rivera-Illingworth et al. [131] employed neural networks. Mihailidis et al. [132] analysed user's hand washing activity to detect their actions and behaviours. Similarly, Wu et al. [133] proposed the use of various tagged objects frequently used by the user to identify and model user behaviour. Aztiria et al. [134] used feedback from the user to improve performance. When behaviour prediction is utilized in a home security scenario to identify intrusion attempts user interference is not always possible. The algorithm proposed in this work utilizes minimal user interference to identify intrusion. D. N. Monekosso and P. Remagnino [135] analysed activities inside a home like cooking, eating, bathroom/ablutions, watching TV, sleeping/in bed, working at a desk or no detectable activity to identify patters in observed values using Hidden Markov Model.

The work of G. Mokhtari et al. [136] used pyroelectric infrared sensors to detect direction in which a user is moving inside a home and an ultrasound array is deployed to detect moving resident's height both these information is combined to calculate the velocity at which someone moves. The researchers utilized Bluetooth Low Energy (BTLE) communication modules for transmitting data to the data server. The work primarily focusses on identifying home inhabitants from each other depending on their physical attribute and the speed at which they move to various places inside the home, so the system only have to distinguish between a few home inhabitants. When this approach is implemented in home security the sample set is significantly large and velocity and physical attribute such as user height is not enough to successfully identify intrusion attempts into the home.

CHAPTER 3 METHODOLOGY

3.1 CHAPTER OBJECTIVES

The chapter explains the research methodology used in the thesis. This chapter is divided into four sections. Section 3.1 explains the chapter objectives while section 3.2 explains the parameters and the techniques used for successfully generating a device's fingerprint. The fingerprint is used to identify devices accessing the home over the internet. Section 3.3 illustrates the factors considered for behaviour prediction algorithm and explains how these factors are identified and how they vary depending on user actions. Section 3.4 details the behaviour prediction algorithm used in the thesis; the section explains the factors which helps the behaviour prediction algorithm to identify legitimate user behaviour and distinguish it from attack behaviour in a timely manner.

3.2 DEVICE FINGERPRINTING METHODOLOGY

From the Related Works in Chapter 2 Section 2.2, it is clear that there are well documented security issues associated with implementing just password based user authentication in the home automation scenario. The system is at its most vulnerable when the home is online. Our work utilizes device fingerprinting and legitimate login credentials as a part of double verification process for authorized user and their device identification. Various approaches for Remote Physical Device Fingerprinting were considered before we settled on fingerprinting using JavaScript, Flash and Geo-Location. Our reliance on JavaScript was justified by a study [137] which showed that, 98% of internet users had their JavaScript enabled when they visited Yahoo's homepage. According to Adobe, more than 1 billion devices were using Flash by the end of 2015. The algorithm implemented in this thesis avoided using Java Plugin for device fingerprinting because of their known security vulnerabilities. To the best of our knowledge, this is the first attempt that incorporates HTML 5's Geo-Location capability into device fingerprinting.

3.2.1 Parameters Considered for Device Fingerprinting

The tables given below, Table 3.1 and Table 3.2, shows all the JavaScript parameters used for Device Fingerprinting. JavaScript is used to identify browser specific and device specific parameters for device fingerprinting.

Browser Specific parameters given in Table 3.1 can be obtained from "navigator.userAgent", "navigator.javascriptEnabled", "navigator.flashEnabled", "navigator.mimeTypes" and "navigator.plugins". "navigator.userAgent" provides information about OS name, OS Bits, Browser Name and Version which can be utilized for fingerprinting and identifying a device, while lesser bit parameters like "navigator.javascriptEnabled", "navigator.flashEnabled" provides 'true' or 'false' values which provides less identifiable information. The algorithm determines whether 'cookies' are enabled by actually setting/retrieving and then deleting the set cookie. 'Local Storage' enabled is also checked in a similar way.

Table 3.1 Browser specific parameters using JavaScript.

Number	Parameters	Obtained From
1	Browser Name	navigator.userAgent
2	Browser Version	navigator.userAgent
3	JavaScript Enabled	navigator.javascriptEnabled
4	Flash Enabled	navigator.flashEnabled
5	Cookie Enabled	By actually setting/retrieving and deleting a cookie
6	Local Storage Enabled	By actually setting/retrieving and deleting an item in local storage

7	Mime Length	navigator.mimeTypes.length
8	Mime Type	[EachMimeObject].type
9	Suffix Associated with each Mime Type	[EachMimeObject].suffixes
10	Number of Plugins Associated with each Mime Type	[EachMimeObject]. enabledPlugin.length
11	Plugin Length	navigator.plugin.length
12	Plugin name	[EachPluginObject].name
13	Each Plugin's Version	[EachPluginObject].description
14	Number of Mime Type Associated with each plugin	[EachPluginObject].length

Table 3.2 Device specific parameters using JavaScript.

Number	**Parameters**	**Obtained From**
1	OS Name	navigator.userAgent
2	Browser Version	navigator.userAgent
3	Screen Maximum Width	navigator.screen.maxWidth
4	Screen Maximum Height	navigator.screen.maxHeight
5	Screen Current Width	navigator.screen.availWidth
6	Screen Current Height	navigator.screen.availHeight

7	Screen Colour Depth	navigator.screen.colorDepth
8	Screen Pixel Depth	navigator.screen.pixelDepth
9	Taskbar Position	Calculated from Maximum Height, Width and Current Height and Width
10	Taskbar Size	Calculated from Maximum Height, Width and Current Height and Width
11	Time zone	navigator.date
12	Country Name	navigator.date
13	Current time	navigator.date
14	Geographical Location (Latitude, Longitude)	navigator.geolocation. getCurrentPosition()

The browser specific parameters mentioned above OS name, OS Bits, browser name, browser version when combined adds to the uniqueness of the fingerprint thus improving the fingerprint accuracy. Other browser specific parameters like, Multi-Purpose Internet Mail Extensions (MIME) length, MIME type, MIME suffixes and their number of associated plugins provides highly identifiable information corresponding to a browser which are utilized for fingerprinting. Total number of installed plugins, plugin name, version of each installed plugins and number of mime types associated with each plugin also contribute to the high accuracy of our fingerprint. The order of the installed plugins retrieved depends on the installation time of each of the individual plugin, as demonstrated by J.R Mayer [44]. A combination of all these parameters are used to develop a client device's fingerprint.

Device specific parameters given in Table 3.2 can be obtained from "navigator.screen" and "navigator.date" object in JavaScript. Parameters like screen maximum width, screen maximum height, screen current width, screen current height, screen color depth, screen pixel depth does not change even if a user changes their web browser. Position of the taskbar

(top/bottom OR left/right) and taskbar size can be deduced from the screen parameters, these two parameters almost never changes. A device's current time, time zone and country name can be obtained from the "navigator.date" object in JavaScript. The OS Name and OS Bits obtained from UA string of the browser are also device specific parameters. Many other device fingerprinting parameters such as, "navigator.language, navigator.product, navigator.appVersion, navigator.appName" etc. can be obtained using JavaScript but these were ignored because they were mostly unreliable and gave inconsistent or false values across different browsers. Moreover some parameters like "screen.updateInterval, screen.buffer" were browser specific.

We utilized the geo-location feature available in the HTML 5 to improve the accuracy of our fingerprinting algorithm. A lot of finger printable parameters can be gathered from "navigator.geoLocation.getCurrentPosition ()"; they include, latitude, longitude, altitude, accuracy, altitude accuracy, heading, speed. We only utilized two of those parameters, namely latitude and longitude in our fingerprinting algorithm. During the course of our work, it was found that these two parameters are readily available in almost all machines which supports HTML5 as compared to other parameters, which requires constant monitoring and in some cases specific equipment at the client's side. After accurately determining the location, Google Application Program Interface (GoogleAPI) is used to identify the actual country name based on latitude and longitude. The country name from GoogleAPI is compared with that obtained from the date object. The country names must be same, but if there is a mismatch it means the client's "navigator.date" object is intentionally giving misinformation or the client's device is in another time zone, either way in case of country name mismatch, the date parameter from our fingerprinting algorithm is ignored.

A client's device specific screen parameters will remain constant over time, Moreover, a client device's time zone and country name is unlikely to change unless they travel outside the country or changes time zones. Even when that happens, by analysing their geo-location and date object the real country name and time-zone can be obtained and compared. So when device specific parameters are unavailable the security and device identification capability of the device fingerprinting algorithm decreases.

The table given below, Table 3.3, shows the Flash parameters used in our fingerprinting algorithm. There are about 38 Flash parameters considered excluding Regular and Non-Regular device fonts. All of the parameters except system fonts are obtained from the 'Capabilities' class in flash. Even though, most of these parameters returned Boolean values with less identifiable qualities, lion share of them were device specific parameters, which when considered as a whole provides reliable information about a device's configuration.

Table 3.3 Device fingerprinting parameters using Flash.

No.	Parameter	Obtained from flash.system.Capabilities	Return Type
1	AV Hardware Disabled	avHardwareDisable()	Boolean
2	CPU Architecture	cpuArchitecture()	String
3	Has Accessibility	hasAccessibility()	Boolean
4	Has Audio	hasAudio()	Boolean
5	Has Audio Encoder	hasAudioEncoder()	Boolean
6	Has Embedded Video	hasEmbeddedVideo()	Boolean
7	Has IME	hasIME()	Boolean
8	Has MP3	hasMP3()	Boolean
9	Has Printing	HasPrinting()	Boolean
10	Has Screen Broadcast	hasScreenBroadcast()	Boolean
11	Has Screen Playback	hasScreenPlayback()	Boolean
12	Has Streaming Audio	hasStreamingAudio()	Boolean
13	Has Streaming Video	hasStreamingVideo()	Boolean

14	Has TLS	hasTLS()	Boolean
15	Has Video Encoder	hasVideoEncoder()	Boolean
16	Is Debugger	isDebugger()	Boolean
17	Is Embedded in Acrobat	isEmbeddedInAcrobat()	Boolean
18	Language	Language()	String
19	Local File Read Disabled	localFileReadDisabled()	Boolean
20	Manufacturer	Manufacturer	String
21	Max Level IDC	maxLevelIDC()	String
22	Operating System	os()	String
23	Pixel Aspect Ratio	pixelAspectRatio()	Number
24	Player Type	playerType()	String
25	Screen Colour	screenColor()	String
26	Screen DPI	screenDPI()	Number
27	Screen Resolution X	screenResolutionX	Number
28	Screen Resolution Y	screenResolutionY	Number
29	Support 32 Bit Processes	Support32BitProcesses()	Boolean
30	Support 64 Bit Processes	Support64BitProcesses()	Boolean
31	Touch Screen Type	touchScreenType()	String

32	Flash Player Version	version()	String
33	Dolby Digital Audio Enabled	hasMultiChannelAudio(flash.media.AudioDecoder.DOLBY_DIGITAL)	Boolean
34	Dolby Digital Plus Audio Enabled	hasMultiChannelAudio (flash.media. AudioDecoder.DOLBY_DIGITAL_PLUS)	Boolean
35	DTS Audio Enabled	hasMultiChannelAudio (flash.media. AudioDecoder.DTS)	Boolean
36	DTS Express Audio Enabled	hasMultiChannelAudio (flash.media. AudioDecoder.DTS_EXPRESS)	Boolean
37	DTS HD High Resolution Audio Enabled	hasMultiChannelAudio (flash.media.AudioDecoder.DTS_HD_HIGH_RESOLUTION_AUDIO)	Boolean
38	DTS HD Master Audio Enabled	hasMultiChannelAudio (flash.media. AudioDecoderDTS_HD_MASTER_AUDIO)	Boolean
39	Regular Device Fonts	Using Action Script in Flash	Array
40	Non-Regular Device Fonts	Using Action Script in Flash	Array

The OS name obtained from flash is compared with those obtained from the "navigator.userAgent". Ideally, the two OS names should match but if they are different it implies the user is using some user agent spoofing techniques even though it is counterproductive in protecting user identity as demonstrated by N. Nikiforakis et al. [55]. So in such a case, user agent and its associated parameters are ignored from the device fingerprinting algorithm. Similarly screen maximum width and screen maximum height obtained from the "navigator.screen" object are compared with the screenResolutionX and screenResolutionY to determine the validity of the "navigator.screen" object. If they are

mismatched, it means screen parameters available from JavaScript are not reliable, so all the screen parameters obtained from “navigator.screen” object can be ignored in the algorithm.

The system fonts installed in a device is mostly unique, it depends on user preferences and the presence of different browser plug-ins and software; we consider both regular and irregular device fonts for our fingerprinting algorithm. These fonts and the order in which these font names are retrieved in flash provide highly identifiable information which aids our fingerprinting algorithm as demonstrated by P. Eckersley [53]. Another method for extracting device fonts is by using JavaScript side channel font detection, in this technique we include the names of the fonts to be checked into fingerprint script, this limits font checking to only well-known system fonts. Moreover, it won’t allow us to determine the order of fonts in a client’s device. So we are not implementing any side channel font detection in our algorithm.

We also tried to detect a client’s history to see if he/she has visited a particular URL through the vulnerability in CSS as exploited by D. Jang [138]. The URL we checked is not indexed in any search engines and will only be visited by a legitimate user, as he visits his home from a device; it is used as a mechanism to identify a legitimate returning user. This attempt failed and confirms the researcher’s notion that, history detection using CSS vulnerability in modern web browsers is not possible. We tried to exploit this vulnerability in Mozilla Firefox version 44 and 45, Google Chrome version 48 and 49 and Microsoft Internet Explorer version 11.

We classify the above mentioned fingerprinting parameters into 9 categories; Parameters from User Agent String, Screen Parameters, Lesser Bit Parameters (cookie enabled, java script enabled, local storage enabled, flash enabled), Mime Parameters, Plugin Parameters, Parameters from Date object, Geo-Location Parameters, Flash Parameters and System Fonts. Our device identification proceeds by identifying, verifying, comparing and analysing various device specific features associated with each of these 9 parameters.

3.2.2 Device Fingerprinting Process

The figure given below, Figure 3.1 shows the Device Fingerprinting process in the proposed system. When a user wishes to access the home over the internet, he requests the login page from the server, the server then returns the login page along with the fingerprint java script. The user provides the login credentials along with the fingerprint of the device he is using. The login credentials are verified, if the verification is passed, then the gathered device fingerprint is analysed to see if there are enough device fingerprinting parameters available to provide a comprehensive fingerprint of the user device. If not, the client is requested to enable his Flash, JavaScript and Geo-location for accurate fingerprinting at the login page again.

There are two fingerprint lists in our database, whose entries are accumulated over time. The 'whitelist' is a list of approved or authorized device fingerprints belonging to legitimate users. Client devices with fingerprints in the whitelist are allowed access to the home after login credential verification. The 'blacklist' is a list of unauthorized or malicious device fingerprints belonging to potential attackers who tried to gain access to the home. Client devices with fingerprints in the blacklist are denied access to the home even if their login credentials are correct.

If the login credential are matched and there are sufficient fingerprinting parameters and the Device Fingerprint is not in our '*whitelist*' and '*blacklist*', then the client should be verified by some other more direct method in order to assure legitimacy. A simple and safe method would be make contact with the client using a phone call to the registered mobile number of the client and verify it is him trying to login to his home. Another alternative is, the server generates a One Time Password (OTP) and sent it to the legitimate user's registered mobile number via Short Message Service (SMS), which the user enters in the website and thus the legitimacy of the user is verified. When a new device's legitimacy is verified, user is asked, if he wants to add the device's fingerprint into the whitelist. A user adds a device's fingerprint to the whitelist, if that device is his own or it is a trusted third party device which is often used to access the home like the clients office computer. Irrespective of the user's

choice to add/not add a fingerprint into the database, he is allowed access to the home after direct verification.

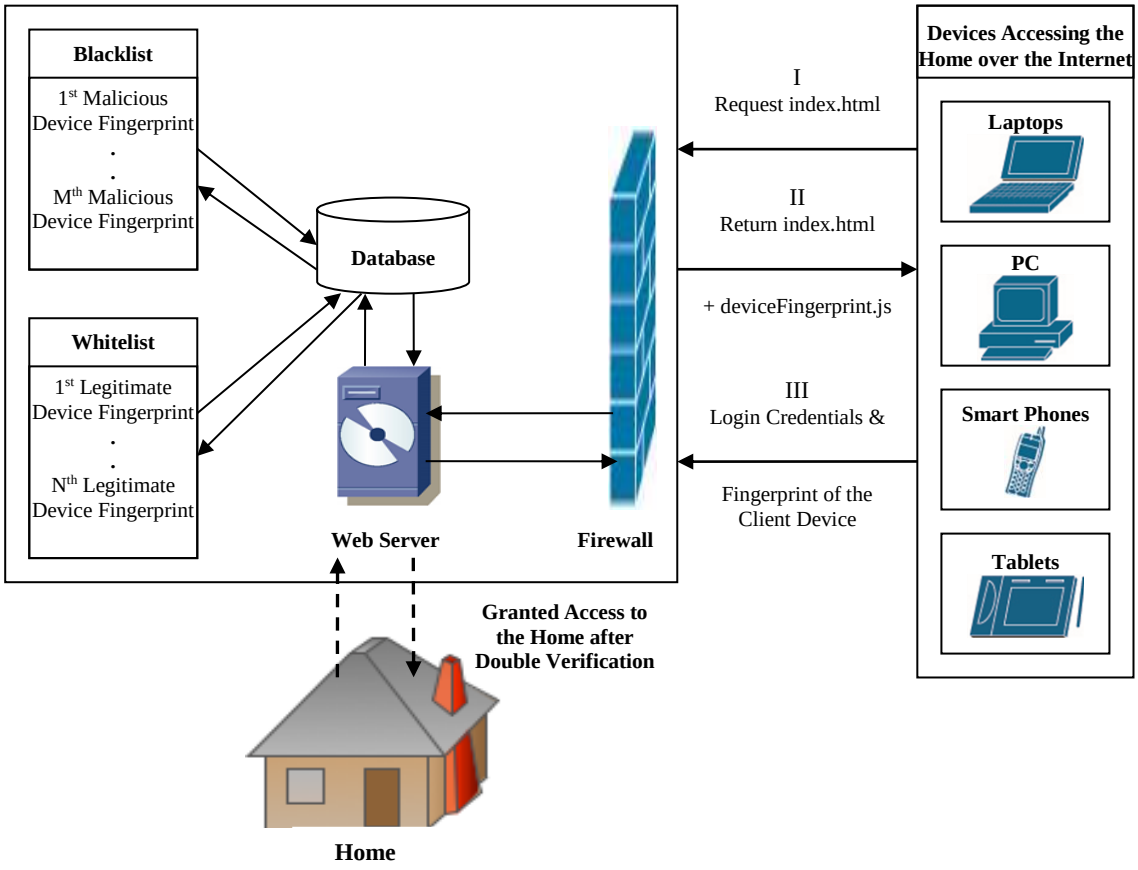


Figure 3.1. Logical Diagram of the Proposed System

During direct verification via phone or OTP, if the verification fails it means the user trying to access the home is not a legitimate user. So that device's fingerprint is added to the blacklist. It also means the login credentials of a legitimate user is compromised, so he is also asked to change them. User devices with continuous and repeated failed login attempts are also added to the blacklist as they are trying to guess the login credentials. When a device tries to access the home whose fingerprint is in the blacklist, it is immediately denied access to the home even if the login credential is correct. This way our proposed system identifies an attacker's device and denies access to the home without bothering the user. The flow chart of the two stage verification process is given in Figure 3.2.

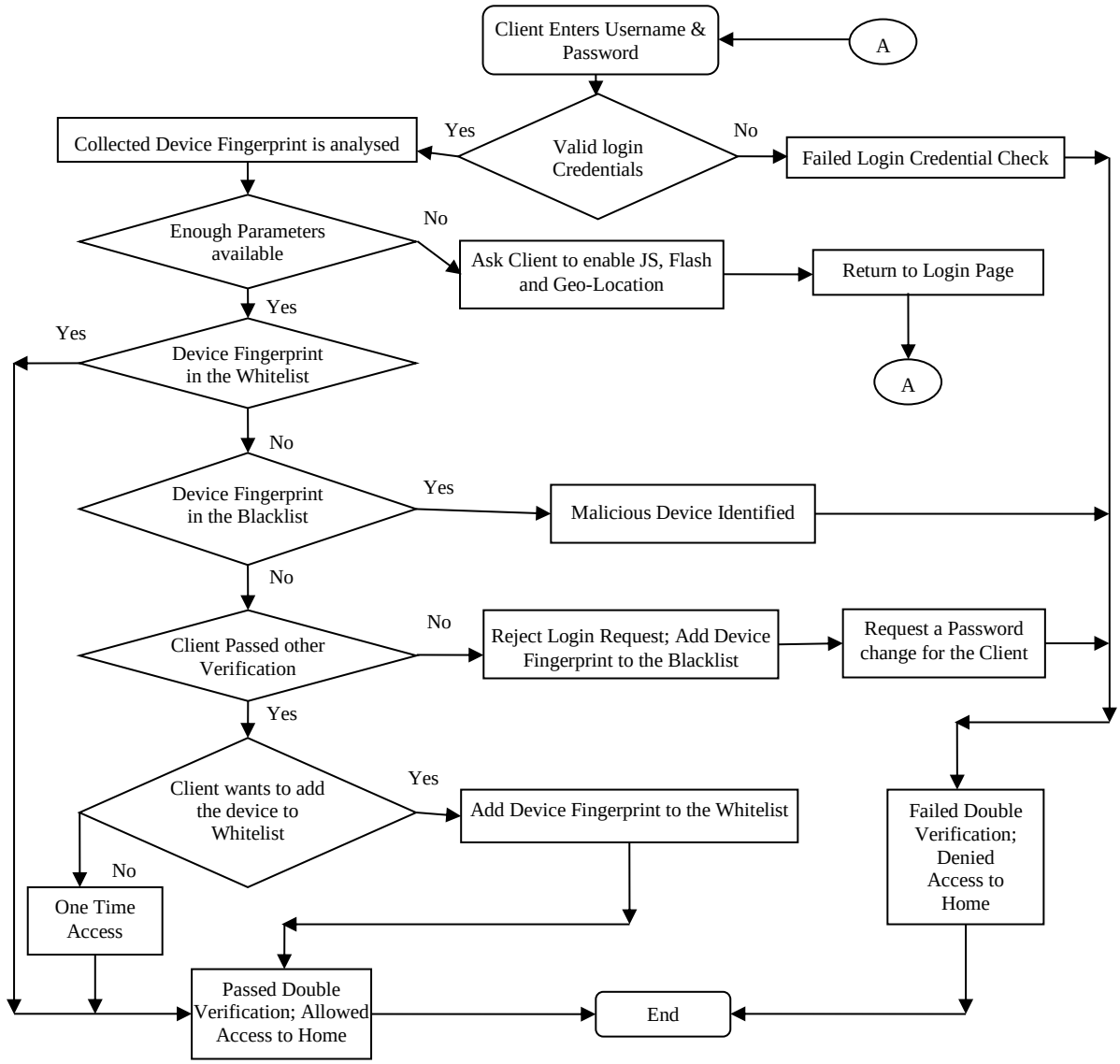


Figure 3.2. Flowchart of the double verification process

OTPs can be easily generated by the server and are short lived. OTPs can only be used once, so even if an attacker managed to record the OTP which is already used he will not be able to abuse it, thus defending against password replay attacks. If a legitimate user makes a mistake when entering the OTP the device's fingerprint is blacklisted and he is denied access to the home from that device for the time being. If the user wants to regain access to the home from a blacklisted device, he has to delete a device's fingerprint from the blacklist of the webserver's database. In order to prevent malicious behaviour and improve home

security this can only be done from inside the home where the webserver is physically located.

```
Start Device Fingerprint Verification Process.
flashParam = gather (client's Flash parameters)
javaScriptParam = gather (client's JavaScript parameters)
geolocationParam = gather (client's Geolocation parameters)
totalFP_ParamAvailable = analyseParamCount (flashParam, javaScriptParam, geolocationParam)
if (totalFP_ParamAvailable <= 7)
        Number of Fingerprint parameters lacking.
        Terminate Device Fingerprint Verification Process.
Else
        ClientDevice_FP = analyseAndDevelopFP(flashParam, javaScriptParam, geolocationParam)
        FPScore_BL = Compare_FP (ClientDevice_FP, eachFP_FromBlacklist)
        If (FPScore_BL > threshold value)
                Fingerprint found in Blacklist.
                Access Denied (home)
                Terminate Device Fingerprint Verification Process.
        Else
                FPScore_WL = Compare_FP (ClientDevice_FP, eachFP_FromWhitelist)
                If (FPScore_WL > threshold value)
                        Fingerprint found in whitelist.
                        Access Granted (home)
                        Terminate Device Fingerprint Verification Process.
                Else
                        Inform User (New device trying to access the home)
                        userResponse = getherUserResponse ()
                        if (userResponse = grant access to new device)
                                OTP = generate OTP()
                                SentOtp_ToUser (OTP)
                                If (userOtp_VerificationPassed())
                                        Access Granted (home)
                                        AmendWhiteList (ClientDevice_FP)
                                Else
                                        Access Denied (home)
                                        Terminate Device Fingerprint Verification Process.
Device Fingerprint Verification Process Completed.
```

Algorithm 3.1. Device Fingerprinting Algorithm

3.2.3 Device Fingerprint Algorithm

As discussed in Section 3.2.1, the 9 device fingerprinting parameters mainly considered in the device identification algorithm are —:

User Agent Parameters: These are parameters obtained from 'navigator.userAgent', they are namely Browser name, Browser version, OS name, OS Bits.

Screen Parameters: These are parameters obtained from 'navigator.screen' object, they are namely Screen maximum width, Screen maximum height, Screen available width, Screen available height, Screen color depth, Screen pixel depth, Taskbar position, Taskbar size.

Lesser Bit Parameters: Lesser bit parameters provide very little identifiable information about a client's device. They are namely Cookie enabled, Local storage enabled, Flash enabled, JavaScript enabled.

MIME Parameters: These parameters are obtained from 'navigator.mimeTypes', they are namely Mime length, Mime type, Suffixes associated with each mime type and Plugins associated with each mime type.

Plugin Parameters: Plugin parameters are obtained from 'navigator.plugin', they are namely Plugin length, Plugin name, Plugin name, Version of each plugin, Number of mime types associated with each plugin.

Date object Parameters: A client device's time zone, country name and current time can be obtained from the 'navigator.date' object.

Geo-Location Parameters: Geo-Location parameters can be obtained from 'navigator.geolocation.getCurrentPosition()', they include the client's current latitude and longitude and the country name corresponding to the latitude and longitude obtained from the Google API.

Flash Parameters: All the flash parameters are obtained from the 'flash.system.Capabilities' class when the client has Flash installed on their browser. The first 38 Parameters mentioned (excluding regular and non-regular device fonts) in Table 3.6 are the Flash parameters considered in the algorithm.

System Fonts: The name and number of fonts installed at a client's machine can be obtained from Flash using 'flash.system.Capabilities' class. This includes Regular device font length,

Non-Regular device font length, Regular device fonts installed and Non-Regular device fonts installed.

These parameters have different significance depending on the amount of information they have about a client device. Each of these 9 parameters are assigned scores depending on the varying degree of similarity with existing device fingerprints in our database. These computed scores will determine the probability match corresponding to each of these parameters. The algorithm determines if a device's fingerprint is present in the database if the total probability score is greater than or equal to the threshold probability score. Our Device Fingerprinting Algorithm is given above in Algorithm 3.1.

3.3 LOGICAL SENSING METHODOLOGY

In the thesis, we analysed various access points in a home to identify different improbable scenarios within a smart home during its operation. Access points are inherent in the structure of a home, which can be used for entering and exiting a home. In a typical home these natural access points are front door, back door, balcony doors and windows. Even though window is not a normal access point it can be used as one; most likely by an intruder depending on the situation. Physical access to a home is only possible through these access points unless serious structural alterations are made to a home. These serious structural alterations cannot be made without drawing attention to the act itself, like blasting or destroying a wall to create an entrance. So, managing access at these access points is crucial in securing a home. The thesis proposes that, irrespective of the number and type of access points in a home, the behaviour of a legitimate user at these access points can be broken down in to a set of possible events which can be predicted.

Based on the purpose of the access points, the thesis classifies access points into primary and secondary. In a home, when an access point is used by its inhabitants as a primary means to enter and exit from their home, it is categorized as primary access point like the front door, back door etc. On the other hand, secondary access points like the window, balcony door etc. also provide entry/exit to a home but they are rarely used for that purpose because there are other convenient ways in and out of a home for a legitimate user.

3.3.1 Primary Access Point

Front door is the primary access point to any home, inhabitants use this door as the main way in and out of their home. Depending upon the architecture and inhabitant needs, there can be one or more primary access points. This thesis proposes the use of motion and proximity sensors to detect user behaviour at primary access points. When a user leaves an occupied home, the motion and proximity sensors placed near the access point inside the home are triggered before the door is opened. Once the user stepped out and closes the door the motion and proximity sensors will not be triggered. When someone enters an empty home, they are entering from outside so, the motion and proximity sensors will not be triggered before the door is opened. Once the door is opened and the user enters the home the motion and proximity sensors placed inside the home will be triggered. Figure 3.3 shows the flowchart of the door state changes and sensor operations of the primary access point.

Table 3.4. Possible states a Main door could take.

State No	**Initial State → Intermediate State**	**Final State**	**Motion Sensor Trigger**		**Proximity Sensor Trigger**		**Home Empty**
			Before	**After**	**Before**	**After**	
1	C → O	O	✓	✓	✓	✓	F
2	C → O	O	×	✓	×	✓	F
3	C → O	O	×	×	×	×	F
4	C → O	C	✓	×	✓	×	F
5	C → O	C	×	×	×	×	F
6	C → O	C	✓	✓	✓	✓	F
7	O → C	C	✓	✓	✓	✓	F
8	O → C	C	✓	×	✓	×	F
9	O → C	C	×	×	×	×	F
10	O → C	O	×	×	×	×	F
11	O → C	O	✓	×	✓	×	F
12	O → C	O	✓	✓	✓	✓	F

13	C → O	O	✓	×	✓	×	F
14	O → C	C	×	✓	×	✓	F
15	C → O	C	×	✓	×	✓	F
16	O → C	O	×	✓	×	✓	F
17	C → O	C	×	✓	×	✓	T
18	C → O	O	✓	✓	✓	✓	T
19	C → O	O	×	✓	×	✓	T
20	C → O	O	×	×	×	×	T
21	C → O	C	✓	×	✓	×	T
22	C → O	C	×	×	×	×	T
23	C → O	C	✓	✓	✓	✓	T
24	O → C	C	✓	✓	✓	✓	T
25	O → C	C	✓	×	✓	×	T
26	O → C	C	×	×	×	×	T
27	O → C	O	×	×	×	×	T
28	O → C	O	✓	×	✓	×	T
29	O → C	O	✓	✓	✓	✓	T
30	C → O	O	✓	×	✓	×	T
31	O → C	C	×	✓	×	✓	T
32	O → C	O	×	✓	×	✓	T

Table 3.4, shows all the possible initial states, intermediate states (represented by '→'), final states and motion and proximity sensor triggers before and after the state changes of the main door when the home is occupied and empty. Table 3.5 represents the special cases the algorithm could take during its operation. States in Table 3.5 are only triggered from a few particular previous state which are explained in this section (section 3.3.1). Motion and proximity sensors should be strategically placed so that, they will only be triggered by someone from inside the home and not by the act of opening or closing of the door. The sensor placements should also ensure that, anyone using the door to enter and exit the home

cannot do so without triggering the motion and proximity sensors. The algorithm works by analysing multiple proximity and motion sensor values before and after the door is opened or closed. The time period between the initial and final states of the door, the number of sensor values considered before the initial state and after the final states in the algorithm can vary depending upon the structure of the home and positioning of the motion and proximity sensors from the door.

Once the door state is changed the algorithm considers number of proximity and motion sensor values before the door state is changed to identify if the door was opened from the inside or outside. After the initial state change the algorithm keeps observing the door for a specific interval of time called ‘door observation time’; the door state during this time is called intermediate state of the door. The algorithm observes the motion and proximity sensor values during the door observation time to identify user actions at an access point.

States 1 to 16 from Table 3.4 are triggered only when the home is occupied and states 17 to 32 occurs only when the home is empty. When the home is occupied and the user opens a closed door from the inside, the proximity and motion sensors are triggered on the way to open the door. After opening the door, the user can either:
(a) Leave the door open and come back into the house by triggering the motion and proximity sensors after opening the door (state 1 in Table 3.4).
(b) Leave the door open and step outside the home without triggering the motion and proximity sensors after opening the door (state 13 in Table 3.4). This leaves the home vulnerable to intruders, so after a fixed amount of time the home state is changed to empty, so a user will have to verify his identity upon re-entry into the home. The time taken by the algorithm to change the home state when the user steps out leaving the door open is called ‘state change time’. Before changing the state of the home the algorithm issues a warning, informing the user about the impending state change. Depending on the physical location of the home, proximity of the door to public areas and user preference the state change time of the algorithm can vary.

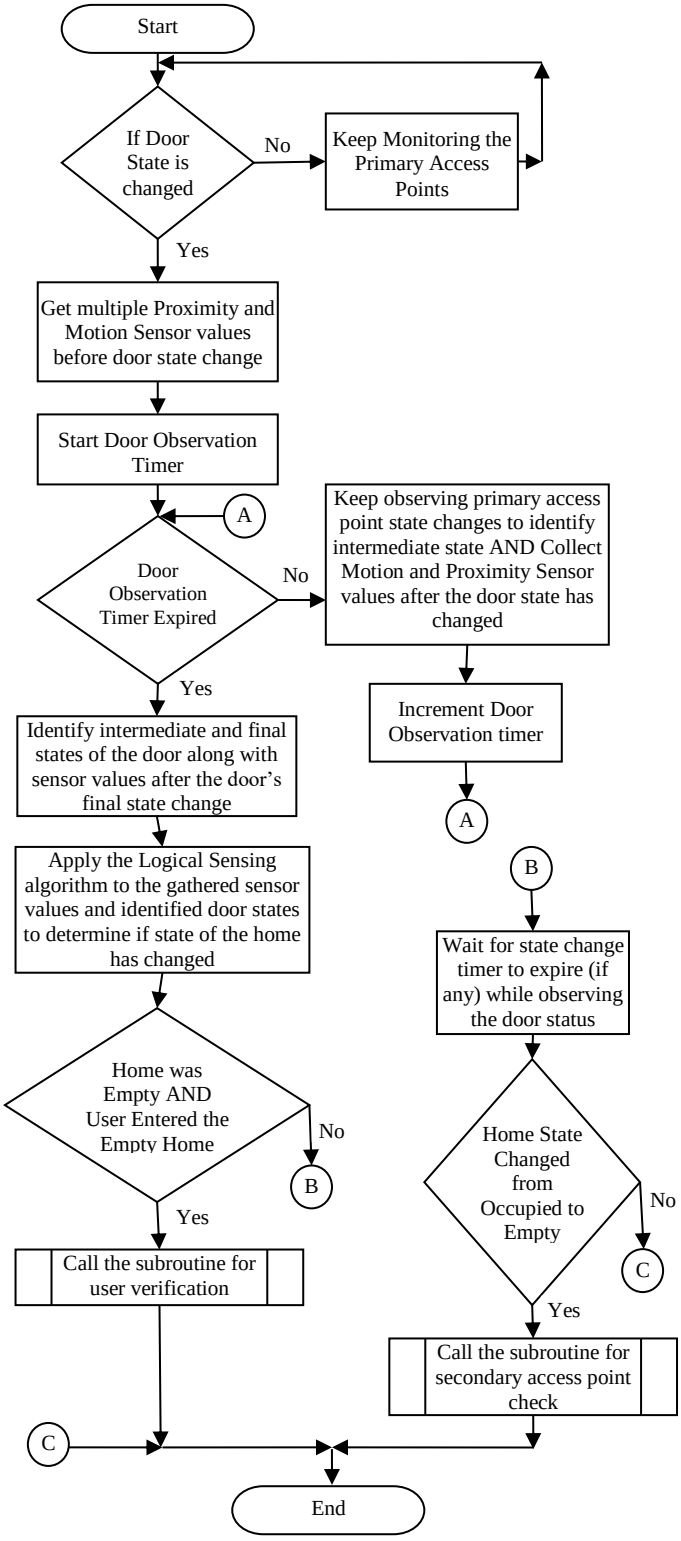


Figure 3.3. Flowchart showing door state changes and sensor operations of the primary access point.

Table 3.5. Special Cases.

State No	Initial State → Intermediate State	Final State	Motion Sensor Trigger	Proximity Sensor Trigger	Home Empty	Algorithm Action
33	O → O	O	✓	✓	F	Reset/Start State Change Timer
34	O → O	O	✓	✓	T	Activate IVM

(c) Step out and close the door behind him within door observation time allowed by the algorithm without triggering the motion and proximity sensors after the door is closed (state 4 in Table 3.4). When state 4 occurs in a single person occupied home the state of the home changes from occupied to empty after the door observation timer has expired.
(d) Close the door from the inside within the door observation time allowed by the algorithm and comes back in, triggering the motion and proximity sensors after the door is closed (state 6 in Table 3.4).

When the home is occupied and the door is open, it can be closed from the inside or from the outside. After closing the door depending on his position, user can either come back into the home or go out. These states are discussed below.
(e) User closes the door from the inside and comes back into the home. The motion and proximity sensors are triggered before and after the door is closed (state 7 in Table 3.4).
(f) User closes an open door coming from inside and steps out of the home leaving the home empty. Since the door is closed coming from the inside motion and proximity sensors are triggered before the door is closed but the sensors are not triggered after the door is closed as the user has stepped out (state 8 in Table 3.4). After triggering state 8, the state of the home changes from occupied to empty when the door observation timer expires.
(g) User closes an open door coming from the inside and opens it again within the door observation time and steps out of the home leaving the door open. The motion and proximity sensors are triggered before the door is closed as the user walks towards the door from inside

the home. The user then opens the door making the previous door state (closed) the intermediate state and the current state (open) the final state. Sensors are not triggered after the final state, as the user executes the final state from outside the home. Since the user has stepped outside the home leaving the door open the 'state change timer' is started, which upon expiry changes the home state to empty (state 11 in Table 3.4).

(h) The user closes an open door from the inside and opens it within the door observation time and comes back into the home. Like in state 11, the door is closed and opened within the door observation time making the former (closed) the intermediate state and the later (open) the final state. Motion and proximity sensors are triggered before the door is closed (initial state) and after the door is open (final state), as the user initially came from inside to close the door and went back into the home after leaving the door open (state 12 in Table 3.4).

The proposed algorithm keeps monitoring the door and sensor values for different state changes. Some states like state 2, 3, 5 and 15 in Table 3.4 are only triggered when their previous states are either state 5 or state 9. Figure 3.4 demonstrates the states the proposed system goes through before states 2, 3, 5 and 15 are triggered. State 2 is triggered when the user opens the door from the outside. Figure. 3.4 (a) shows state 2 transition; state 2 is triggered when the home is occupied and a closed door is opened without triggering the motion and proximity sensors before opening the door but the sensors are triggered after opening the door. In a single person occupied home, such an event will only take place, when the user steps out of a home leaving the door open (state 11 and state 13 in Table 3.4) and within the state change timer expiry period states 5 or 9 are triggered. When state 2 is triggered it means the user re-entered the home leaving the door open, so the state change timer can be reset. Figure. 3.4 (a) also shows the transition of states 9 and 10.

Figure. 3.4 (b) shows state 3 transition; state 3 happens when the home is occupied and a closed door is opened without triggering the motion and proximity sensors before and after opening the door. In a single person occupied home, like state 2, state 3 will only be triggered if states 5 or 9 are triggered before the state change timer has expired. Figure. 3.4 (c) shows state 5 transition; state 5 transpires when the home is occupied and a closed door is opened

and closed without triggering the motion and proximity sensors before or after closing the door. In a single person occupied home, similar to states 2 and 3, state 5 is only triggered if states 5 or 9 are triggered before the state change timer has expired. State 5 can be the previous state of state 5 or in other words state 5 can repeatedly happen until the state change timer has expired. Figure. 3.4 (d) shows the transition of the system to state 15; state 15 is triggered when the home is occupied and a closed door is opened and closed. The motion and proximity sensors are not triggered before opening the door but the sensors are triggered after closing the door which indicates the user stepped back into the home after closing the door. So after triggering state 15, in a single person occupied home the state change timer is reset.

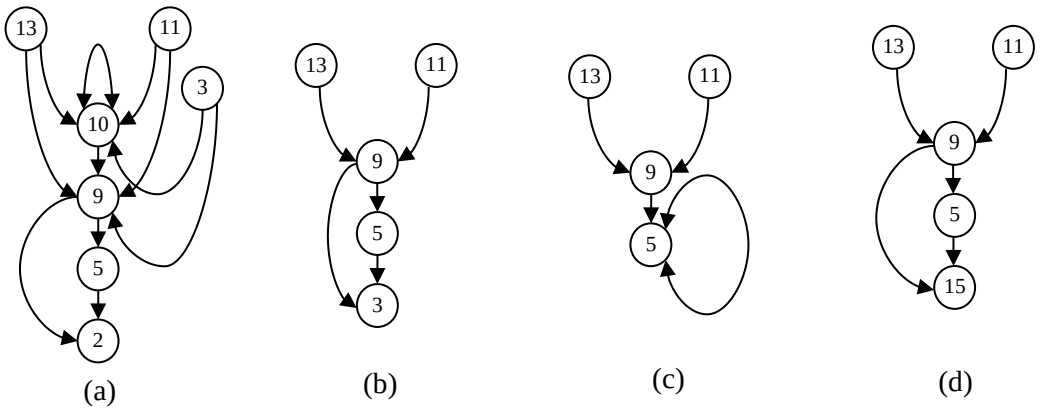


Figure 3.4. Showing various state Transitions.
(a) Shows State 2 transition along with state 9 and 10 transition; (b) State 3 transition; (c) State 5 transition (d) State 15 transition during the operation of the home.

A single person occupied home becomes empty when states 11, 13, 4 and 8 are triggered. When states 11 and 13 are triggered the algorithm utilizes a state change timer as mentioned above in (b) and (g). When states 4 and 8 are triggered the home state is changed when door observation timer expires as mentioned above in (c) and (e). Door observation time is a relatively short period of time compared to the state change time.

When a particular state is triggered the previous state of the door is also considered to accurately determine the possible next states and the occupied status of the home. Table 3.6 shows the possible previous states for a particular state and timer actions in the proposed system when the home is occupied and empty. Whenever a home changes state from

occupied to empty the algorithm checks if the secondary access points to the home are secure. If not it issues a warning to the user to secure the secondary access points. Figure 3.5 shows the flowchart of the secondary access point checking when the home becomes empty.

State 10 is triggered when an open door is first closed then opened from outside without triggering the motion and proximity sensors before or after the initial and final states. State 10 is triggered when the previous states are states11, 13 or 3 and with state change timer still running. State 10 can also be the previous state of state 10 i.e. state 10 can repeat itself until the state change timer has expired. Similarly, states 9, 14 and 16 are only triggered when their previous states are either are 3, 10, 11 or 13. In a single person occupied home, triggering states 14 and 16 means the user has re-entered the home so the state change timer is reset.

Table 3.6. Possible next state for a particular state.

State No	Possible Previous States	IVM Trigger	Timer Status
0	17, 19, 20, 27, 31, 32	No	None
1	0, 5, 6, 7, 9, 14, 15	No	None
2	5, 9	No	Reset State Change Timer
3	5, 9	No	Continue State Change Timer
4	0, 5, 6, 7, 9, 14, 15	No	None
5	5, 9	No	Continue State Change Timer
6	0, 5, 6, 7, 9, 14, 15	No	None
7	0, 1, 2, 10, 12, 16	No	None
8	0, 1, 2, 10, 12, 16	No	None
9	3, 10, 11, 13	No	Continue State Change Timer
10	3, 10, 11, 13	No	Continue State Change Timer
11	0, 1, 2, 10, 12, 16	No	Start State Change Timer
12	0, 1, 2, 10, 12, 16	No	None
13	0, 5, 6, 7, 9, 14, 15	No	Start State Change Timer
14	3, 10, 11, 13	No	Reset State Change Timer

15	5, 9	No	Reset State Change Timer
16	3, 10, 11, 13	No	Reset State Change Timer
17	4, 8, 17, 26, 22	Yes	Start Identity Verification Timer
18	Irrelevant	Alarm	Alarm triggered, so no Timer
19	4, 8, 22, 26	Yes	Start Identity Verification Timer
20	4, 8, 22, 26	No	None
21	Irrelevant	Alarm	Alarm triggered, so no Timer
22	4, 8, 22, 26	No	None
23	Irrelevant	Alarm	Alarm triggered, so no Timer
24	Irrelevant	Alarm	Alarm triggered, so no Timer
25	Irrelevant	Alarm	Alarm triggered, so no Timer
26	11, 13, 20, 27	No	None
27	11, 13, 20, 27	No	None
28	Irrelevant	Alarm	Alarm triggered, so no Timer
29	Irrelevant	Alarm	Alarm triggered, so no Timer
30	Irrelevant	Alarm	Alarm triggered, so no Timer
31	11, 13, 27	Yes	Start Identity Verification Timer
32	11, 13, 27	Yes	Start Identity Verification Timer
33	3, 10, 11, 13, 33	No	Reset/Start State Change Timer
34	20, 27	Yes	Start Identity Verification Timer

Whenever states 3, 10, 11 and 13 are triggered (the door remains open and the state change timer is running) the algorithm keeps monitoring the front door and the sensors until another door state is triggered or until the state change timer is expired. When the user re-enters the

home by triggering the motion and proximity sensors before the state change timer has expired without changing the door state (door still remains open), state 33 from Table 3.5 is triggered. As the user re-entered the home, the algorithm stops and resets the state change timer. Upon resetting the state change timer, the algorithm keeps observing the sensor values as the door is still open. When the user steps out of the home again without changing the state of the door (door still remains open), triggering the sensors state change timer is started, which on expiry changes the home state to empty. So state 33 can be its own previous state.

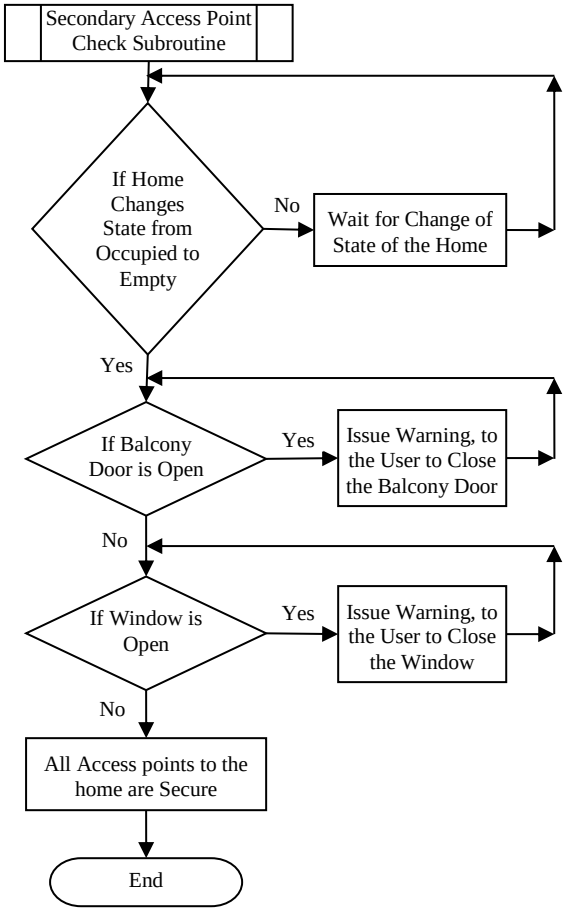


Figure 3.5. Flowchart demonstrating secondary access point check when home becomes empty.

When a home changes its state from empty to occupied, the identity of the person causing the state change has to be verified. The techniques used for verifying user identity [139] can vary from simple 4 digit pin, facial recognition, retinal scan verification, fingerprint verification to sophisticated biometric gait recognition, vein recognition and voice

recognition. There can be one or more Identity Verification Mechanism (IVM) depending on area secured, user preferences and security requirements. The user identification should be done within a fixed time period called the 'verification time period'. The verification timer stars once the door to an empty home is opened and someone enters, it stops when a valid user identity is confirmed. When the timer exceeds the verification time period an intruder alert is triggered. Depending upon the level of security and user preference the intruder alert can be an audible alarm to scare off intruders and alert the neighbours, silent alarm to alert the authorities or any other defensive measures. The 'verification time period' is subjective to user habits and location of the IVM. Figure 3.6 explains the algorithm when door to an empty home is opened. Figure 3.7 shows the identity verification process in the algorithm.

When the home is empty and the closed main door is opened from outside without triggering the motion and proximity sensors. After opening the door the user can either:

(i) Close the main door within the door observation time allowed by the algorithm and enter the home, triggering the motion and proximity sensors while walking into the home after closing the door (state 17 in Table I). When the home is empty and the user comes in after opening the door he will have to confirm his identity within the verification time period.

(j) Leave the door open and enter the home triggering the motion and proximity sensors while moving into the home (state 19 in Table I). The verification timer is triggered as soon as the door is opened so the user have to verify his identity using the IVM.

(k) Leave the door open and decides to stand at the door or go out of the house, without triggering the motion and proximity sensors (state 20 in Table I). After triggering state 20 the door remains open, so the algorithm keeps monitoring the motion and proximity sensors until another door state is triggered; whenever motion and proximity sensors are triggered immediately after state 20, it means someone entered the home through the open door triggering state 34 from Table II, so IVM is activated and user identity is verified.

(l) Closes the door within the sensing time allowed by the algorithm and steps out of the home without triggering the motion and proximity sensors after the door is closed (state 22 in Table I). So sensors are not triggered before the initial state or after the final state of the

door and no one entered the home, so user identity is not verified, home remains empty and the door is closed.

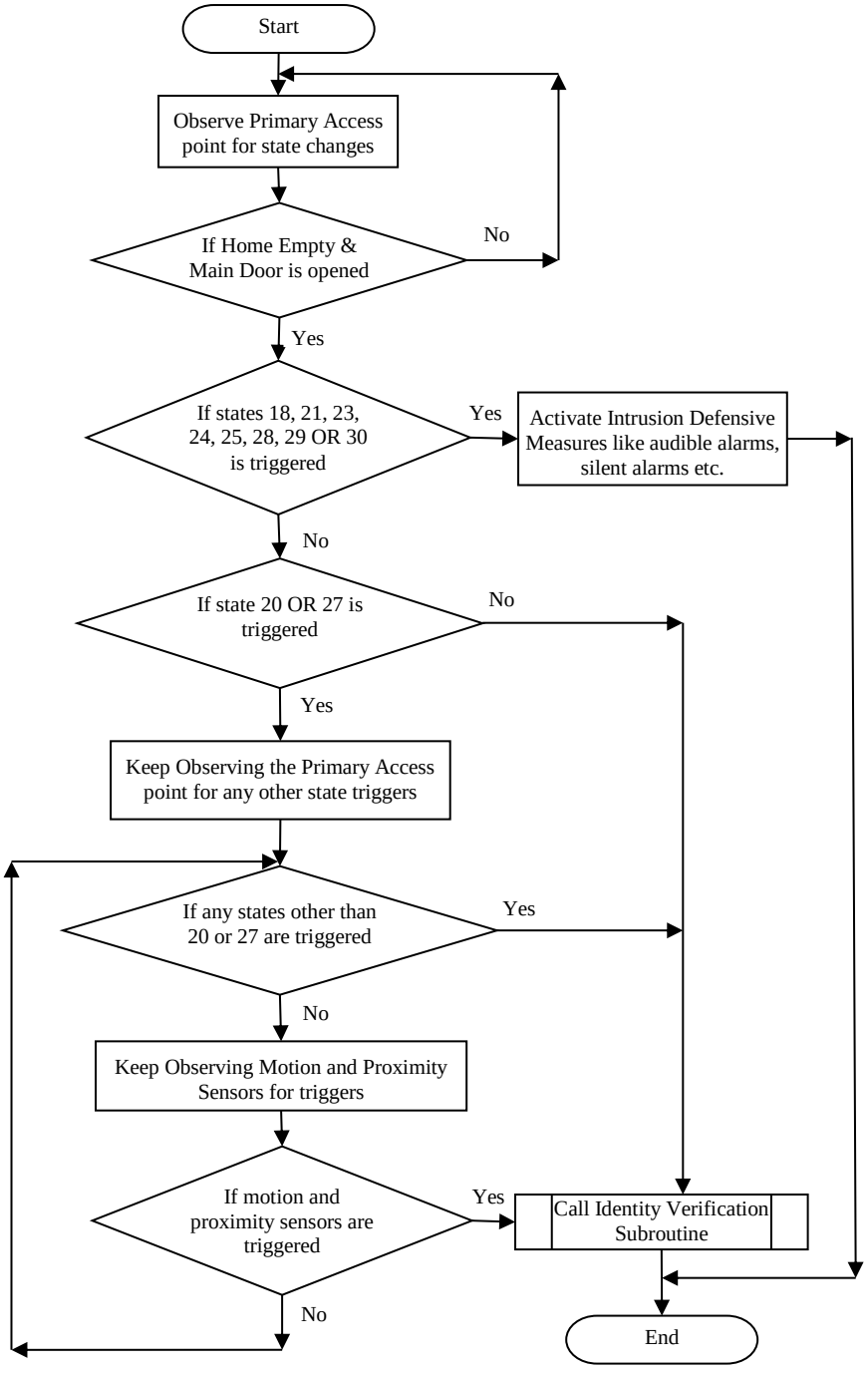


Figure 3.6. Flowchart explaining the algorithm when door to an empty home is opened

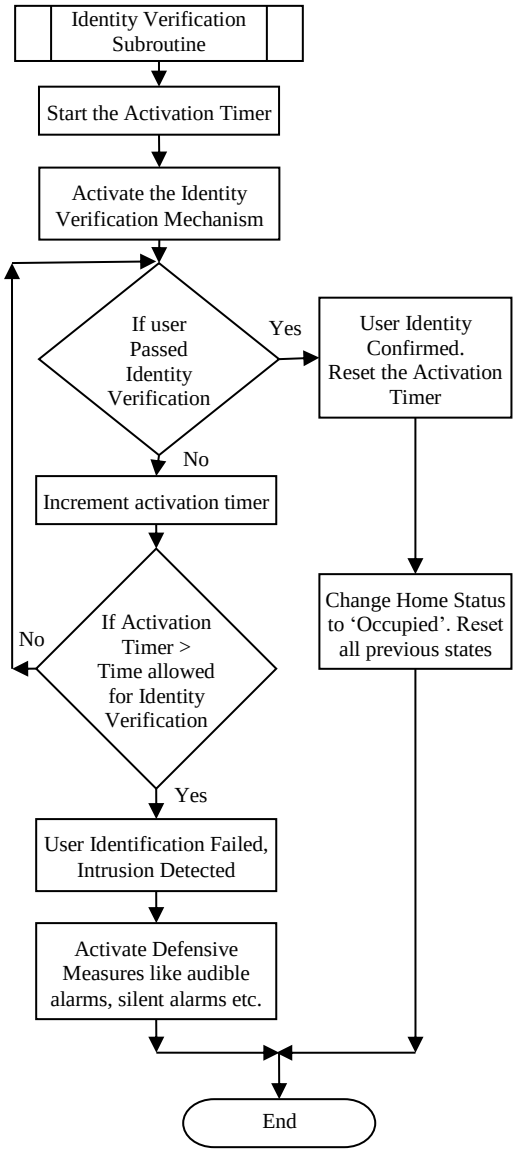


Figure 3.7. Flowchart showing Identity Verification Process.

State 18 is triggered in an empty home when the closed door is opened by triggering the motion and proximity sensors before and after the initial and final states. Similarly, state 23 happens when a closed door is opened and closed by triggering motion and proximity sensors before and after the initial and final states. State 24 is triggered when an open door is closed and the motion and proximity sensors are triggered before and after the initial and final states. Likewise, state 29 happens when an open door is closed and then opened triggering the sensors before and after the initial and final states. All the above states 18, 23, 24 and 29

occurs when the proximity and motion sensors are triggered before the door is opened, this will not happen in an empty home. So, when any of these states are triggered irrespective of the previous state, intrusion defence mechanisms are triggered without waiting for user identity confirmation.

States 21, 25, 28 and 30, occurs when the home is unoccupied and the closed door is opened from inside. In an empty home, this only happens when someone gains access to the home using a secondary access point or using some other means. So irrespective of previous states, intrusion defence mechanisms are triggered without activating IVM to confirm user identity.

When the state change timer expires after triggering states 11 or 13 the door is left open and the home becomes empty; the user can then trigger either states 26, 27, 31 or 32. State 26 is triggered when the user closes the open door without triggering the motion or proximity sensors, which means the door is closed from the outside and user stayed outside. Since there is no entry into the home the home status remains empty and there is no need for user identification. In state 27, the user closes and opens the door without triggering any sensors before or after the initial states, which indicates the door is closed and opened from the outside and no one has entered the empty home. Similar to state 26, in state 27 the home remains empty and no one enters the home so no user identification is necessary. State 27 can be its own previous state making it repetitive. After triggering state 27 the door remains open, so the algorithm keeps monitoring the motion and proximity sensors until another door state is triggered. Similar to state 20, whenever motion and proximity sensors are triggered immediately after state 27, it means someone entered the home through the open door, so IVM is activated and user identity is verified.

State 31 occurs in the system when the user closes an open door and comes back into the home triggering the motion and proximity sensors. The system is not sure about the identity of the person re-entering the home so the IVM is activated to verify user identity. In state 32, the user closes an open door and then opens it and comes back in to the home triggering the motion and proximity sensors. Similar to state 31, after triggering state 32 user identity has to be verified.

After states 11 and 13, if user re-enters the home triggering the sensors after the state change timer has expired without changing the state of the door (door left open) the algorithm will activate the IVM to verify the identity of the user. The IVM is placed inside the home so whenever the user identity is successfully verified the home becomes occupied and all the previous states of the door are cleared and the current state is set to state 0.

If there are multiple primary access points in a home, in order to determine user actions at various access points, motion and proximity sensors has to be deployed at each of the access points. In a single person occupied home the user can only use one primary access point at a time to enter or exit from a home. The algorithm can be implemented at each of the primary access points independently. When more than one primary access point is open and user stepped out of the home leaving the door open, state change timer is started, the algorithm observes motion and proximity sensor readings at each of the open primary access points because the user can enter the home through any of the open primary access points. If the proximity and motion sensors are triggered at any of the open access points state change timer is reset. If none of the proximity and motion sensors are triggered then upon expiry of the state change timer the algorithm changes the home state to empty. When user re-enters the home without changing the state of the door, special case 34 from Table II is triggered for that particular access point and IVM is activated and user is asked to confirm his identity. When primary access point states are changed by the re-entering user, then corresponding state from Table I is triggered for that particular access point.

Status of other primary access points are only checked when the user closes a primary access point and the home state is changed to empty. If the user closes a primary access point and the home state is changed it could mean the user is leaving the vicinity of the home, so the algorithm checks the status of other primary and secondary access points and warns the user to secure them if they are open. If the user is leaving the primary access point open and stepped out through it and the home state is changed to empty after the expiration of the state change timer, it means user is likely close to the home, so other primary access point status is not checked to warn the user. When he re-enters the empty home, IVM is triggered and user identity is verified regardless of the state of any other primary access points.

When multiple access points are open and the user stepped out of the home through one access point leaving it open and enters the home through another open access point without changing its state before the state change timer expires triggering special state 33 from Table II for that particular access point. If user changed the state of the primary access point through which he entered corresponding states from Table I are triggered for that access point. The time difference between the motion and proximity sensor triggers from each of the access points used by the user to exit and enter the home is considered along with the distance between access points to distinguish between normal user behaviour and sneaky attack behaviour. The algorithm takes into account the minimum time required by the user to move between the access points to identify attack behaviours.

3.3.2 Secondary Access Points

The balcony door and windows form the secondary access points in a home. In a typical home, the balcony door is not used as the main access point to and from a home. Usually balcony door opens into a relatively secure and private area, sometimes even a few floors up. So, these balcony doors can remain open for long periods of time when the house is occupied. When the home becomes empty an observant, resourceful and proficient intruder can use this door to gain access to the home, in order to avoid that, balcony doors must be closed when the home becomes empty. Moreover, when the home is empty the balcony door should not be opened under any circumstances. The algorithm keeps monitoring the state of the balcony door, so in an empty home when the balcony door is opened the system triggers intrusion defence mechanisms without waiting for any identity verifications.

In a typical home windows are opened from the inside under normal circumstances. So, by placing motion and proximity sensors near the window inside the home we can identify if windows are opened from inside or outside. The proximity and motion sensors should be strategically deployed so that the window cannot be opened from inside without triggering them. Similar to balcony doors, windows in a home should not be opened when the home is empty, so when the home is empty and the window is opened the system triggers the intrusion defence mechanisms without waiting for identity confirmation. In addition, when the home is occupied and the window is opened from the outside without triggering the

motion and proximity sensors placed near the window, the system triggers a warning and asks the user to confirm his identity because under normal circumstances windows are rarely opened from the outside.

The proposed system also observes the bed in which a user sleeps in to determine if the user is in bed or not. The algorithm observes reading from the force sensors placed underneath the mattress to determine if a user is occupying the bed. Various force sensors are placed underneath the mattress to identify force at different areas of the bed. The algorithm considers these sensor readings to distinguish between users occupying a bed and foreign objects placed on the bed. Highly accurate measurement of force underneath the bed is not necessary to identify when a bed is occupied and distinguish between user and foreign objects on the bed. When the bed is empty the force on the bed is significantly less compared to when it is occupied by the user.

In a single person occupied home if any of the access point states are changed when the user is in bed it indicates an intrusion to the home. The chance of intrusion significantly increases if this happens during night as most intruders prefer the cover of darkness to infiltrate their targets. So, if the user is in bed and any of the primary or secondary access point states are changed during night between 10:00 p.m. and 6:00 a.m. the algorithm activates intrusion defence mechanisms without waiting for identity verification. When this happens during the day between 6:01 a.m. and 9:59 p.m. the algorithm triggers a warning indicating the change of state of the access point and the user is asked to confirm his identity. The time of the day for alarm trigger can be varied depending upon user preference, location of the home, outside accessibility to secondary access points etc. In some cases an open balcony door or window may be closed without user interference even when the user is in bed due to wind, so in the proposed system this scenario is also considered before identifying intrusion.

Even when there are more than two secondary access points, the algorithm can identify intrusion attempts in real time by observing each secondary access points individually and implementing the logical sensing algorithm for windows and doors and considering user position in bed when necessary.

3.3.3 Fire Alarm

The work of B. Fouladi [7] discussed the weakness in the existing smart home architecture and demonstrated how an attacker will compromise various networked elements in a home. The easiest way to get the inhabitants out of a home is to trigger an emergency alarm like the fire alarm. When a fire alarm is triggered all the automatic locks of a home are disabled. During home fire the carbon monoxide and the ambient temperature levels in the area of the fire will go up and inversely the humidity in and around the area will go down. If there is no change in humidity, temperature or carbon monoxide levels, the algorithm warns the user about a possible attack attempt which the user can verify.

Each twelve second average of the temperature, humidity and carbon monoxide sensor readings are compared to detect fire. If there is more than 2o difference between the twelve second average temperatures and more than 3% difference in twelve second average humidity then the triggered fire alert is validated. It takes around 24 hours for the carbon monoxide sensor to stabilize, so carbon monoxide readings are only considered once the system had been activated for at least 24 hours. If the average Rs/Ro resistance value of the MQ 9 gas sensor employed is less than 8.9 over the twelve second period, the fire alarm triggered is confirmed. Whenever a fire alarm is triggered the proposed algorithm checks the carbon monoxide, temperature and humidity levels in the area of the fire to make sure the alarm is triggered by fire and not by a manipulative attacker. The proposed system implements MQ 9 gas sensors, temperature and humidity sensors to detect fire in the home.

Warnings are triggered as a means to get user attention when something out of the ordinary happens. In the proposed system warnings are generated when: (a) user leaves the door open to a home and steps outside, also the state change timer is about to expire; warning is triggered when there is 20 seconds remaining in the state change timer as a means to let the user know, the home is about to change its state to empty soon. Upon receiving the warning user can either, step back into the home triggering the motion and proximity sensors or choose to ignore the warning and let the home become empty when the state change timer runs out. (b) The home becomes empty and at least one of the secondary access points are

unsecure; the algorithm warns the user about the open secondary access point. Upon receiving the warning the user comes back into the home and secures the secondary access point. The warning will not stop until all the secondary access points are secured. (c) One or more of the secondary access point windows are opened from the outside when the home is occupied. Identity verification timer is activated and the user is asked to confirm his identity because under normal operating conditions windows are rarely opened from the outside. (d) Any of the secondary access points changed states between 6:01 a.m. and 9:59 p.m. when the user is in bed. User is asked to confirm his identity and identity verification timer is started because during the day when the user is in bed it is unlikely that secondary access points changes states when the user is in bed. (e) Fire alarm is triggered without sufficient change in humidity, ambient temperature or carbon monoxide levels. The warning informs the user about a possible fire alarm manipulation which user can verify on sight.

An alarm is triggered when the algorithm is sure there is an intrusion in the home. Alarms are usually triggered when the user fails to confirm their identity within the identity verification time or failed thrice consecutively with their identity verification attempts. When the primary access point is opened from the inside when the home is empty the algorithm doesn't wait for any identity verification to sound the alarm. Similarly, when any of the secondary access points are opened when the home is empty alarm is triggered without waiting to confirm user identity. When the user is in bed and any of the closed secondary access points changed states between 10:00 p.m. and 6:00 a.m. alarm is activated without waiting for user identity confirmation. The algorithm uses alarm as the primary intrusion alert mechanism.

When there are multiple occupants in a home the states of the main door remains the same. When a situation arises that changes the home state to empty, the user is asked to confirm the empty state change. When the home is non-occupied the user confirms the state as empty and the algorithm changes the home state to empty. When the home is empty irrespective of the number of inhabitants the defensive mechanisms proposed in Section III are applicable. When there is an additional inhabitant in the apartment the secondary access point behaviour during day and night changes. If the inhabitants are using the same bed to sleep in, then the

corresponding force sensor values when they both occupy the bed together and individually is analysed to determine the occupancy of inhabitants in the bed during night. If both of them are in bed and any of the closed secondary access points are opened during the night (10:00 p.m. and 6:00 a.m.) alarm is triggered and if it happens during the day (6:01 a.m. and 9:59 p.m.) a warning is triggered and the algorithm asks the user to confirm their identity. If inhabitants are using different beds, force sensor readings from different beds are considered to determining user occupancy in bed.

3.4 BEHAVIOUR PREDICTION METHODOLOGY

In the previous section, we developed logical sensing techniques to predict all the possible user actions at primary and secondary access points in order to identify intrusion attempts and react to it, to keep the home secure. When someone enters the empty home the existing home security system activates the user identity verification timer. The user has a fixed amount of time (identity verification time) to confirm his identity. The algorithm has a 90 second pre-set identity verification time. It can be argued that, 90 seconds is plenty of time for a targeted robber to get into the home and escape before the defensive measures are triggered. Identity verification time can be significantly reduced or varied depending on user behaviour.

This section of the thesis proposes a behaviour prediction algorithm to further improve the security of the home. The thesis defines behaviour as any recognizable pattern in a sequence of observations. When analysing behaviour in smart homes the algorithm tries to identify recognizable patters in user behaviour during the training phase of the algorithm, this information is later used to predict normal user behaviours in a timely fashion.

This thesis identifies behaviour prediction parameters and develops behavioural prediction algorithm to successfully distinguish legitimate user behaviour from attack behaviour. Often an intruder only requires a few minutes to complete their task, so any security measure implemented at home has to identify intrusions swiftly to prevent the intrusion attempt being successful. The work utilized various contact sensors, light sensors, piezo-electric sensors, microcontrollers and Raspberry Pi to identify and detect user behaviours. Fig. 3.8 and Fig.

3.9 represents parameters and sensors considered for machine learning and behaviour prediction.

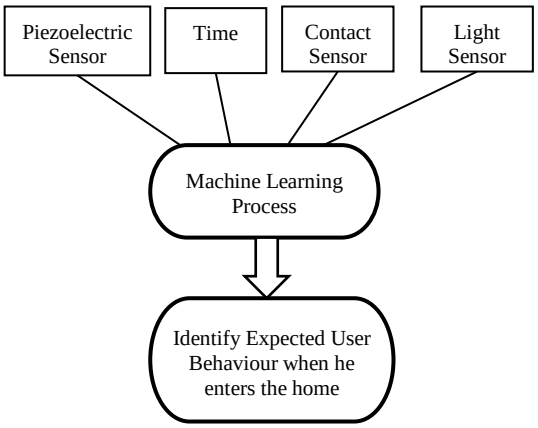


Figure 3.8. Represents factors and sensors considered for machine learning.

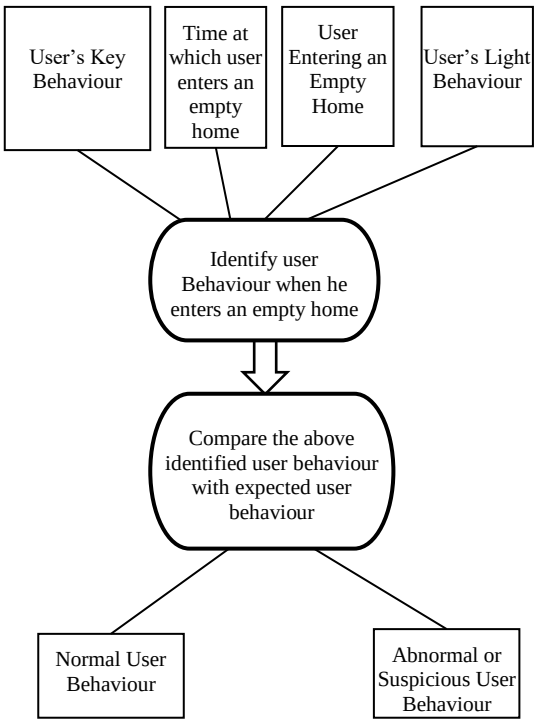


Figure 3.9. Represents the parameters considered for timely identification of user behaviour when he enters an empty home.

3.4.1 Time Parameters

In order to successfully identify user behaviour a twenty four hour day is divided into two parts, namely day and night. User behaviour between 6:30 am and 5:30 pm is classified as day behaviour and behaviour between 5:31 pm and 6:29 am is classified as user's night behaviour. User behaviour during day and night is analysed separately because there is a drastic difference between user's day and night behaviours. Before the algorithm is capable of identifying legitimate user behaviour it needs to be trained to look for familiar behaviour.

In time parameter, the algorithm identifies the time at which an empty home becomes occupied or in other words the algorithm identifies the time of the day or night in which a legitimate user comes home to an empty home. Time in which someone enters an empty home is of particular importance because almost all intruders tries to break into the home when it is empty. Moreover, if an intruder tries to break into an occupied home there is a very high chance that user can react to the break-in. If the intrusion happens when the user is in bed the security system identifies user's position in bed and reacts to it by activating defensive measures as discussed in section 3.3.

Once there is enough data to predict the normal time at which a user comes home the algorithm uses the information to identify normal user actions and differentiate them from suspicious actions. User's entry time to the home during day and night are analysed separately. The algorithm observes the primary access point when home status is empty, the time at which a user comes home to an empty home can be identified when the primary access point is opened and the user walks into an empty home. Contact and motion sensors were used to identify this.

3.4.2 Light Behaviour

When a legitimate user enters the home and it is dark he switches the lights on within a very short period of time (few seconds) because he already knows the position of the light switches inside his home. On the contrary, if an intruder or outsider enters the home he does not know where the light switches are, so, it takes longer for an intruder to switch the lights on. Moreover, some intruders prefer to work in the dark as turning on the lights might give

away their presence in the home. Light behaviour is only considered as part of the behaviour prediction algorithm when the intensity of the light inside the apartment is low and visibility is limited because the user may choose not to turn the lights on as there is enough natural light inside the apartment. This mostly limits the algorithm's light behaviour prediction to night time.

During training, the algorithm calculates the time taken by a legitimate user to turn the lights on when he enters a dark unoccupied home. When an empty home becomes occupied and the light intensity is low in the apartment, the time taken by the person entering the apartment to turn the lights on is identified and compared with normal user behaviour. If there is a significant change in the observed behaviour and expected user behaviour the observation is classified as abnormal user behaviour. The abnormal behaviour contributes negatively to the legitimacy of the person entering the apartment. The flowchart given in Figure 3.10 shows the light behaviour prediction in the algorithm.

3.4.3 Other User Behaviour

Any repetitive user behaviour which can be predicted and identified within a short time after the user had entered an empty home can be integrated into the algorithm for legitimate user identification. Integrating other predictable user behaviour into the algorithm enhances the user identification capability of the approach. These parameters can be user specific like keeping the house key in a bowl, keeping the footwear in a specific place, opening of a specific window upon entering the home. Important factors when considering other behavioural parameters is the ease with which a user behaviour can be identified and the time taken to detect the specific user behaviour. Time taken to identify user behaviour becomes a significant factor because entry of a non-legitimate user to an empty home had to be identified in a timely manner to activate the defensive measures.

The thesis uses piezo-electric sensors to identify when the apartment key is placed inside the bowl. The user always places the keys to his home in the bowl when he enters the home, irrespective of the time of the day (day or night). So this behaviour prediction can be used during day and night to identify user legitimacy. During training the algorithm learns the

average time taken by a legitimate user to place the key in the bowl once the door to an empty home is opened. During its operation the algorithm calculates the time taken by the user to place the keys in the bowl after he first enters an empty home, this behaviour is compared with the pre-established user behaviour to identify the legitimacy of the user entering the home. If the identified and expected user behaviour differ it contributes negatively when identifying the identity of the person entering the home.

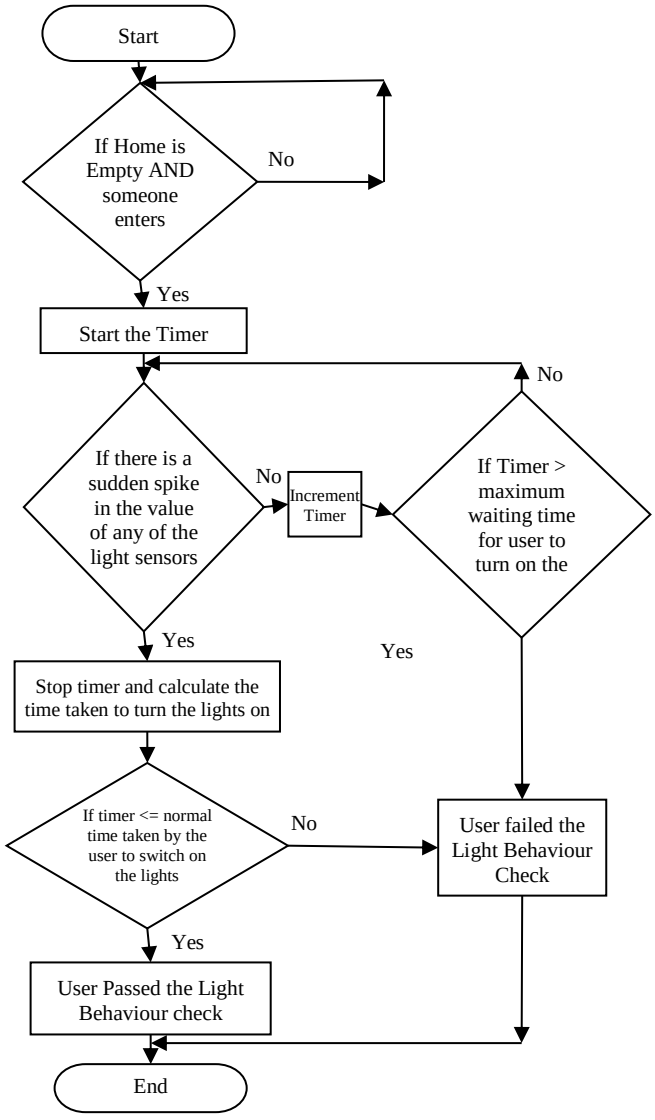


Figure 3.10 Flowchart demonstrating User's Light Behaviour Prediction.

After identifying and learning the behaviour prediction parameters for a particular home the algorithm uses naive Bayes classifier to predict normal user behaviour. The Bayes classifier will give the probability of a user entering the home being a legitimate user depending on their behaviour inside the home. The behaviour parameters for the algorithm are modelled to distinguish normal and abnormal user behaviour in a timely manner so that intrusion defensive mechanisms can be activated to effectively prevent intrusion attempts.

When an anomaly in user behaviour is identified by the behaviour prediction algorithm it could mean that there is an intrusion attempt or the legitimate user is behaving in an unexpected manner, either way from a security stand point it is a situation which requires further analysis. As a defensive mechanism to the anomalous user behaviour, the algorithm can warn the user about the abnormal behaviour and further reduce the identity verification time from 90 seconds to 45 seconds. The warning about the anomalous user behaviour and potential intrusion to the home can be given on all the electronic devices carried by the user like smart phones, tablets, smart watches, laptop etc.

CHAPTER 4 EXPERIMENT SETUP

4.1 CHAPTER OBJECTIVES

This chapter explains the experiment setup and hardware needed to conduct the experiment and test the developed algorithms. The chapter is divided into three sections. Section 4.1 narrates the chapter objectives while section 4.2 explains the experiment setup and specifications for testing the proposed device fingerprinting algorithm. Section 4.3 details the hardware design and specification of other devices like micro controllers, sensors and raspberry Pi necessary to test the developed logical sensing and behavioural prediction algorithms.

4.2 EXPERIMENT SETUP FOR DEVICE FINGERPRINTING

The experiment discussed in this thesis tried to identify a physical device accessing the test website www.fingerprintmydevice.com uniquely using the Device Fingerprinting algorithm developed and mentioned in section 3.2.

Java is used to provide the programming language platform for our work, Java Servlet was utilized for proper server implementation. The Servlet was implemented using Apache Tomcat version 7.0. The work was hosted on a shared server running Linux Centos operating system version 6.8. The server has 18GB RAM and 1TB Hard disk with Intel(R) Xeon(R) CPU E5-2620 0 processor operating at 2.00GHz. Our work was implemented using a database consisting of four tables. The database implementation at the server was done using MySQL 5.1.73.

For simplicity our device identification attempt was restricted to PCs and laptops. Our algorithm was developed to work with and identify devices that uses the 4 most popular web browsers, namely, Mozilla Firefox, Google Chrome, Microsoft Internet Explorer or Edge and Apple Safari. Since 2008, over 95% of internet users use either one of these four web browsers [140]. Access to the website was also restricted to trusted people in order to maintain the accuracy of the collected data. The data was collected between May 2016 and July 2016.

4.3 EXPERIMENT SETUP AND HARDWARE DESIGN FOR LOGICAL SENSING AND BEHAVIOUR PREDICTION

The proposed access monitoring and control mechanism at home is implemented using Raspberry Pi 3 which has 4× ARM Cortex-A53 processor operating at 1.2GHz, Broadcom VideoCore IV graphics processor, 1GB LPDDR2 (900 MHz) built in RAM, one 10/100 Mbps Ethernet port, 2.4GHz 802.11n built in wireless adapter and a 32GB class 10 micro Secure Digital (SD) Card as the hard disk storage. The Pi works on a Raspbian Operating System (OS) optimized for Raspberry Pi. The OS is burned on to the SD card from a laptop which is then inserted into the Pi. The algorithms are implemented using Java as the programming platform and MySQL as the database. Java 7 JDK (Java Development Kit) and MySQL are installed in the Raspberry Pi from Debian repositories using the APT (Advanced Packaging Tool) commands with root user permissions.

At the access point Arduino Uno microcontroller with ATmega328P IC is used to gather data. Arduino Uno module has fourteen digital input/output pins (6 of which can be used as Pulse Width Modulation (PWM) outputs), six analog inputs, a USB connector port, a 16 MHz ceramic resonator, a power jack, an In-Circuit Serial Programming (ICSP) header, and a reset button. Arduino is flexible and offers a variety of digital and analog pins, it can be connected to a PC using USB, and it can run in standalone mode or as an interface connected to a PC. Arduino is cost effective and is an open-source project backed up by a strong online community. Each microcontroller in the experiment is connected to a PC using USB and programmed using the Arduino Interactive Development Environment (IDE).

A Micro Contact/Limit Switch is used at the doors and windows to sense the state of doors and windows. Adjustable Passive Infrared (PIR) Motion Sensors and HC-SR04 ultrasonic range sensors capable of noncontact measurement from 2 cm to 400 cm are used to identify user activities inside the home near an access point. Every living thing with temperature above absolute zero emits heat energy in the form of radiation, this may be invisible to the naked eye but can be detected by PIR sensors. The PIR sensors implemented here has a field of view less than 180^{o}.

A Light Dependent Resistor (LDR) is used to measure the intensity of light at various parts in the apartment. LDR uses photo-resistor with variable resistance which changes with the intensity of light. The resistance decreases with the increase in intensity of light on the sensor surface. Piezoelectric sensors are used to identify when the apartment keys are placed inside the bowl. Piezoelectric sensor uses the piezoelectric effect which converts between electrical and mechanical forms. In Piezoelectric sensor, the acoustical pressure is converted into electrical voltage which can be measured using the Arduino Uno module. Acoustic pressure is generated when the user places his keys into the bowl. Figure 4.1 (a) and (b) shows the deployment of piezoelectric sensor in the key bowl.

(a) (b)

Figure 4.1

(a) Shows Piezoelectric sensor deployed in the key bowl. (b) Shows keys placed inside the key bowl.

The communication between the microcontroller and the Pi is wireless. Wireless communication technology is easy to install and reduces system cost. We considered various wireless technologies like Wi-Fi and Bluetooth. Wi-Fi was discarded because of its high power consumption and high cost, while Bluetooth was discarded because of high power consumption, limited range and security issues [20]. The proposed system is implemented using ZigBee technology based on the IEEE 802.15.4 standard with a communication range varying from 10 – 100m. ZigBee allows large-scale network configurations and utilizes low power radio with a data-rate capability of 250 kb/s. These features makes ZigBee the ideal communication technology in smart home networks. Moreover, many secure communication techniques with ZigBee [141] – [143] were suggested and successfully

implemented. This makes ZigBee a comparatively secure wireless communication technology.

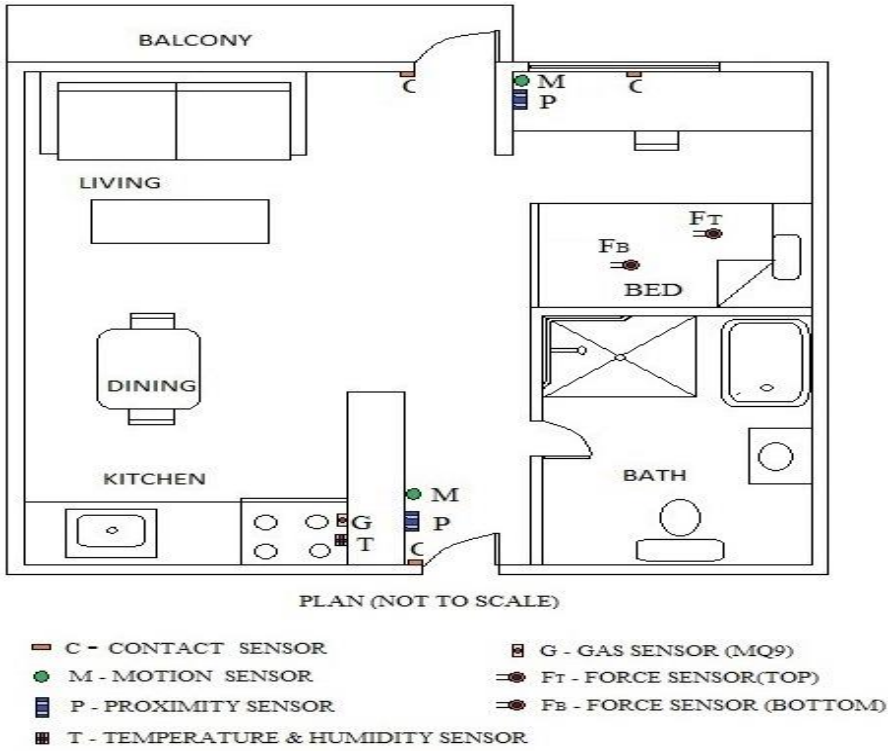


Figure 4.2. Shows the plan and location of the sensor deployments in the apartment during the logical sensing.

The experiment for testing logical sensing algorithms were conducted in a studio apartment for over one month time period, while the experiment for testing behaviour prediction algorithms were conducted for a period nine weeks in a studio apartment. The apartment is situated on the second floor of a six story flat complex. The balcony door opens to a 15 square feet small balcony on the second floor. Physical access to the balcony is only possible through the home or by scaling the building. The front door opens to a common area which all the inhabitants of the flat complex has access to. The window of the apartment is on the second floor so under normal circumstances it will not be opened from the outside. An open balcony door or window has a chance of being closed by wind but under no circumstances a closed window or door will be opened without human interference. The main door which is the primary access point cannot be closed or opened without user interference. Four major

points in the home were considered to obtain logical sensing parameters, namely; primary access point (the main door to and from the apartment), secondary access points (balcony door and window), the bed in which user sleeps on, the kitchen where the fire detection parameters are collected. Figure 4.2 shows the plan and location of the sensor deployments in the apartment during the logical sensing experiment.

Four major areas in the home were considered to obtain behaviour prediction parameters, namely; primary access point (the main door to and from the apartment) to identify entry to the home, light intensity near the front door inside the apartment, light intensity inside the apartment near the balcony door, placement of apartment key inside the bowl. Figure 4.3 shows the plan and location of the sensor deployments in the apartment during the behaviour prediction experiment.

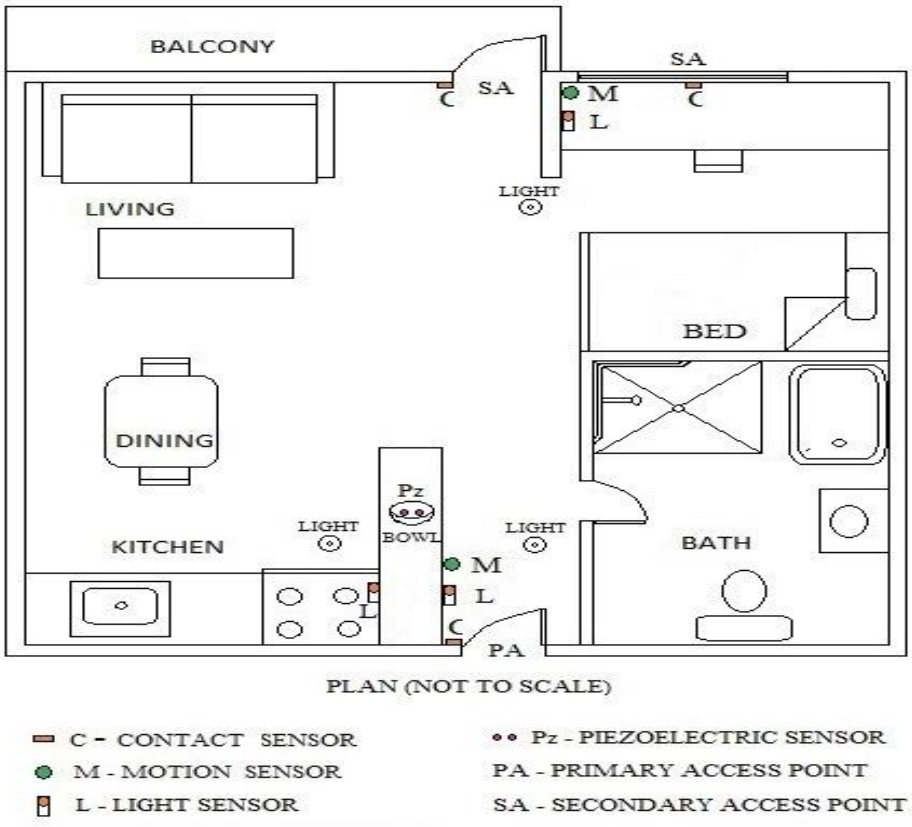


Figure 4.3. Shows the plan and location of the sensor deployments in the apartment during the logical sensing.

Four boards are designed to obtain logical sensing parameters. Board I is deployed near the primary access point. It consist of wires to proximity, motion and contact sensors to obtain various logical sensing parameters. The board is connected to the Arduino Uno module which supplies the power to the sensors. The Arduino Uno microcontroller is then connected to a ZigBee communication module. Figure 4.4 shows the board I and microcontroller deployment at Primary Access Point and Figure 4.5 shows board one installation at the primary access point. Board II is deployed near the secondary access point. It is connected to two contact sensors, a motion and proximity sensors. Each of the contact sensors are connected to window and balcony door. The motion and proximity sensors are deployed near the window to identify user movements before the window is opened. Board II also has provisions to be connected to Board IV which is connected to the force sensors in the bed. Similar to board I, board II is also connected to Arduino Uno microcontroller which powers the board and communicates the sensor values to the Pi using ZigBee module. Figure 4.6 shows board two installation at secondary access points and Figure 4.7 shows board II and microcontroller deployments at secondary access point.

Board III is deployed in the kitchen, it is connected to MQ 9 carbon monoxide sensor, DHT 11 temperature and humidity sensor. Board III is designed to measure the carbon monoxide, temperature and humidity levels in the area to detect fire. Similar to board I and II board III relies on Arduino Uno module for power and ZigBee module for communication. Figure 4.8 board three installation in the kitchen. Board IV is deployed near the bed it is connected to two circular 0.5 inch diameter force sensing resistors. Both force sensors are deployed underneath the mattress. One in a region where the user places his shoulder and another close to the middle where the user's abdomen and pelvic region usually rests. Both force sensors has a full scale measurement accuracy of ±5%. Moreover, force sensors used in the experiment are readily available, cost effective, flexible and provides un-obstructive force measurements. Light sensors were integrated into boards I, II and III to measure the intensity of light near the primary access point and inside the apartment.

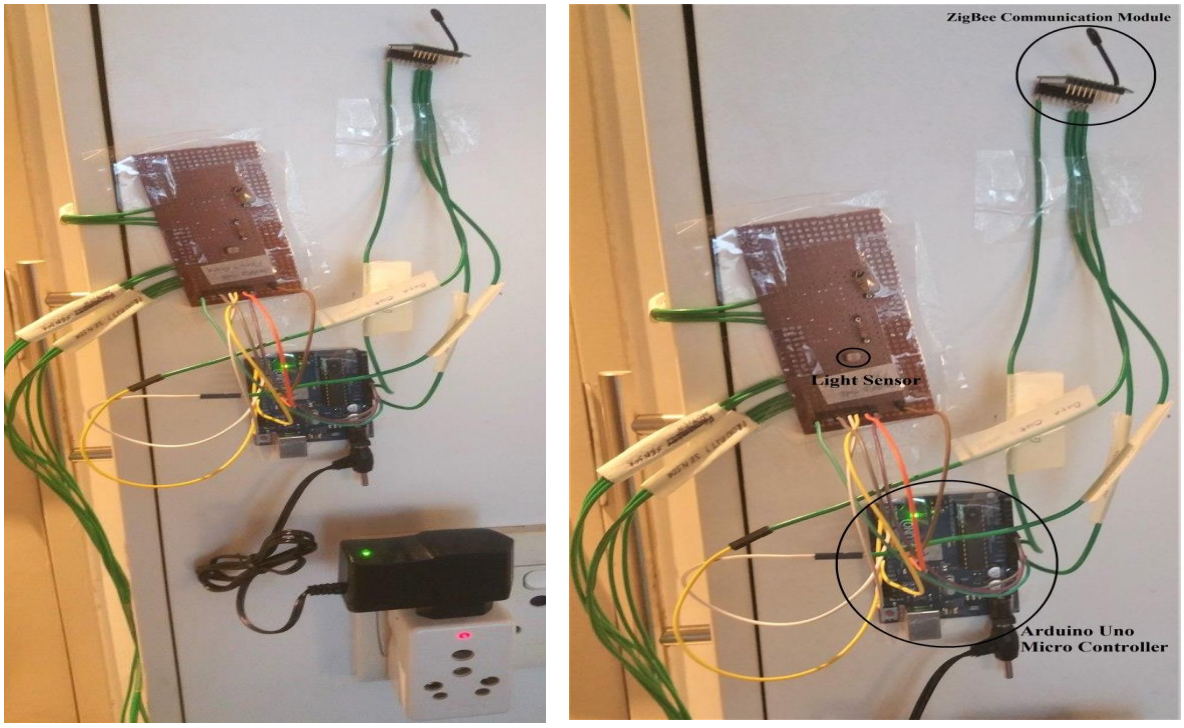
Board IV is connected to board II through wires, like board II it also draws power from the Arduino module connected to board II and uses the same ZigBee module to communicate

values to the Pi. In short, the experiment setup consist of one Raspberry Pi 3 which acts as the central server connected to a ZigBee module to receive sensor data from various boards and a buzzer alarm to signal intrusion; four boards connected to various sensors to collect logical sensing parameters; three Arduino Uno boards with their own power supply connected to ZigBee communication modules to send sensor data to the Pi.

Figure 4.4. Shows Sensors, board and microcontroller deployment at Primary Access Point.

The power to each of the Arduino Uno boards is routed through a power bank. The power banks needs to provide a 9V–1000mA DC output to the Arduino boards. Any reliable power bank which has 5000mAh or higher capacity with 5V–1000mA USB power output would be enough to provide the power backup. The 5V output power from the power bank is

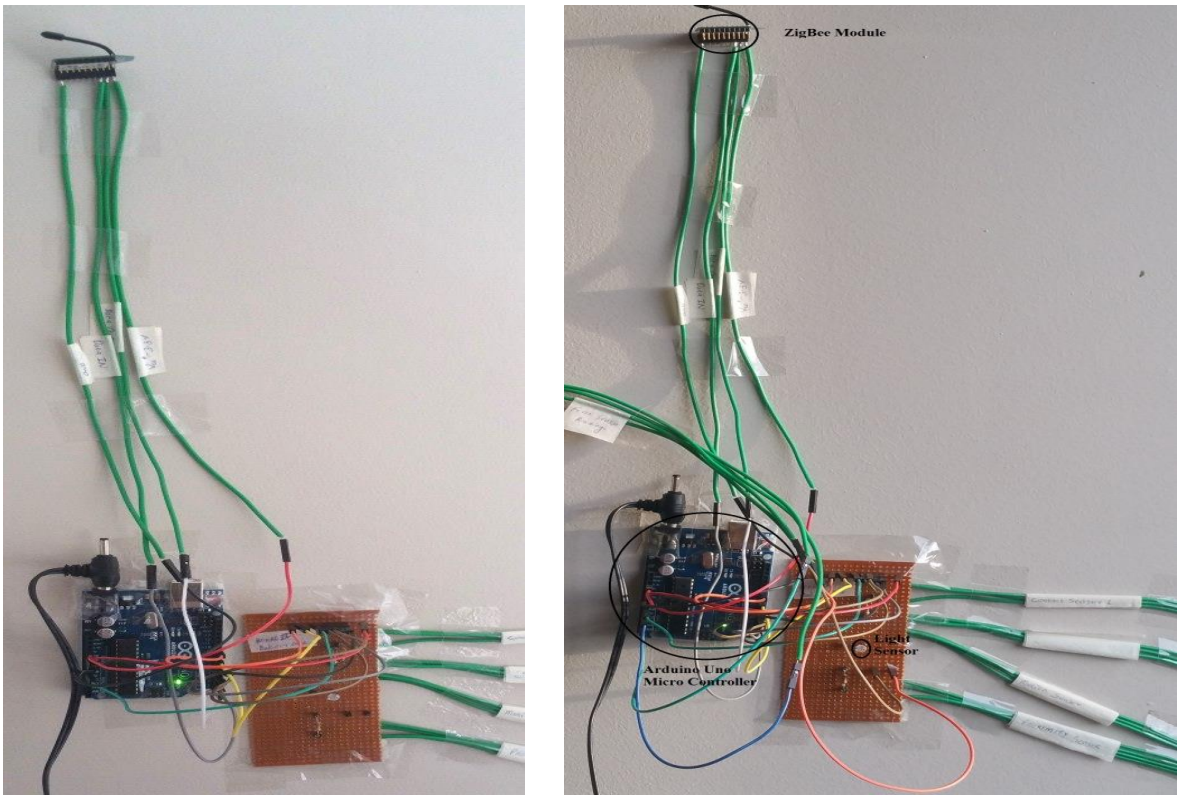
boosted to 9V when connected to the Arduino board through a USB to 2.1mm DC 9V Booster Cable. The power bank is plugged into the apartment's 230V–50Hz AC power supply through a 5V–1000mA AC to DC adapter. Even when the power to the apartment is cutoff the Arduino Uno boards and the sensors are still active and will be able to identify intrusion attempts by drawing power from the power bank. Figure 4.8 shows the modified power source. Compared to AA or AAA batteries power banks offers reliable and durable power supply over time without replacement. Moreover, power banks are relatively cheap, readily available and can supply power to the Arduino boards and sensors for a week without recharging.



(a) (b)

Figure 4.5. (a) Board one installation at Primary Access Point. (b) Board one deployed at primary access point with highlighted light sensor utilized by behaviour prediction algorithm.

All the ZigBee communication is implemented using ZigBee Series 1 module. ZigBee module at the Pi is configured as the ZigBee Coordinator (ZC) while the modules attached to the microcontrollers is configured as the ZigBee End Device (ZED). The data rate of all the ZigBee modules are set at 9600 bits per second (bps).



(a) (b)

Figure 4.6. (a) Board Two installation at Secondary Access Points in the experiment setup. (b) Board Two installation at Secondary Access Point with highlighted light sensor utilized by behaviour prediction algorithm.

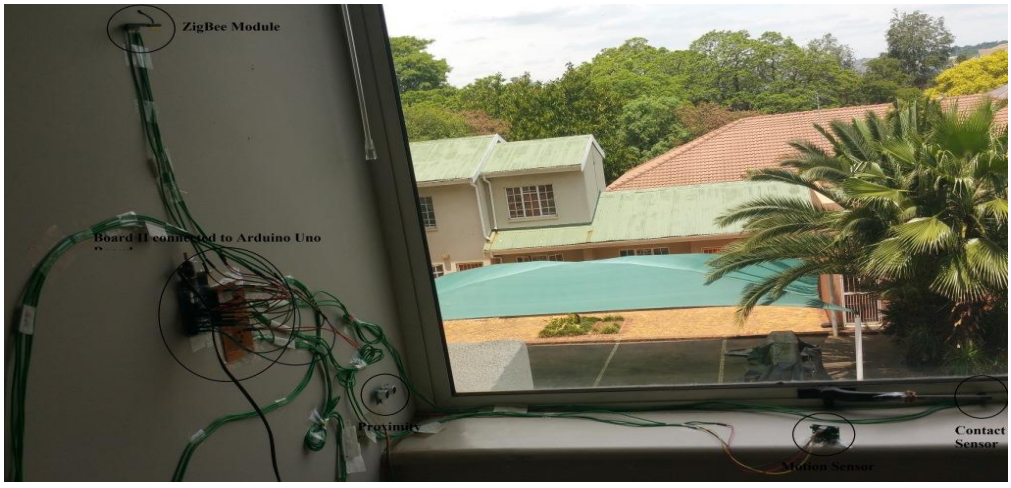


Figure 4.7. Shows Sensors, board and microcontroller deployments at Secondary Access Point.

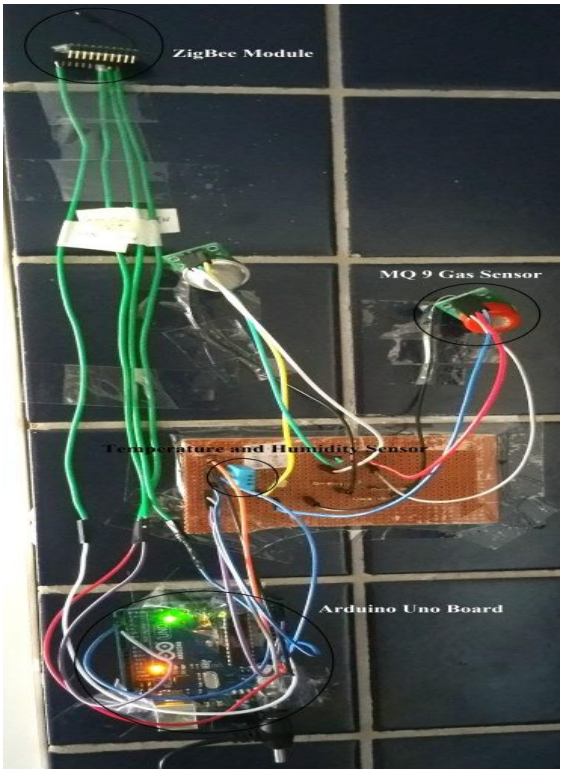


Figure 4.7. Board Three installation in the Kitchen; in the experiment setup.

Figure 4.8. Shows the modified power source, power to the microcontroller is routed through a power bank.

All the ZigBee modules implemented uses AES encryption, to enhance security, the coordinator is configured not to allow unsecured joins to the network, so under no circumstances the encryption key is sent as plain text over the air. Each ZigBee module is programmed using a free XCTU software utility which allows communication with Digi RF modules.

The lights in the apartment can be broadly classified into two sets namely, lights near the front door (front door lights) and lights inside the apartment (inside lights). LDR sensors deployed on board I and board III identifies actions of front door lights while LDR deployed on board II will identify the intensity of inside lights. User's light behaviour is only considered when the intensity of light inside the apartment is low. The contact sensor uses a 5V supply voltage and is connected to the board using a 10kΩ resistance. Similarly, LDR sensor also uses a 5V supply voltage and 10kΩ load resistance when connecting to the board. Measuring the voltage from LDR can give 1024 distinct light intensity values.

MQ 9 sensor uses a supply voltage of 5V and a load resistance of 10 kΩ, it can measure carbon monoxide concentration from 200 ppm to 1000 ppm along with LPG and methane gases. MQ 9 sensor can work under temperatures from -20oC to 50oC, relative humidity of 95% and oxygen concentration ranging from 2% to 21%. As the concentration of carbon monoxide gas increases the measured voltage also goes up.

The ratio of air resistance Rs to Ro gives the concentration of measured gasses. Air resistance Rs, can be calculated using the equation:

$$Rs = \frac{(Vcc - V)}{V} \tag{4.1}$$

Where Vcc is the supply voltage and V is the voltage measures across the sensor.

From the MQ 9 sensor data sheet it is clear that the ratio of Rs to Ro in clean air is 9.9, so the value of Ro is obtained from the Rs value calculated by putting the sensor in clean air, using the equation:

$$Ro = \frac{Rs}{9.9} \tag{4.2}$$

The sensor was left in clean air for 24 hours to be stabilized before Ro value is calculated; the calculated Ro value is 2.05. The sensor board is then moved and installed to its working area. The calculated Ro value is used to calculate the Rs to Ro ratio during its operation.

The force sensor is connected to the board using a 10 kΩ resistance and uses a 5V supply voltage. Force sensor is made of Polymer Thick Film (PTF) which decrease in resistance when pressure is applied to the surface of the sensor. In the experiment, bed occupancy is determined by measuring the voltage across the resistance. Force Sensor Resistance (FSR) is calculated using the equation:

$$FSR = \frac{(Vcc - V) * R}{Vcc} \tag{4.3}$$

Where Vcc is the supply voltage, V is the voltage measures across the sensor and R is the connected resistance. Using FSR, conductance and the applied force is calculated.

The motion sensor is deployed 2.5 ft. from the front door and proximity sensor at 3 ft. from the front door so they will only sense activities inside the home. The verification of the user is done using a laptop connected to the Pi using a wireless modem. The Pi is accessed from the laptop by means of Secure Shell (SSH) via a username and password. The user enters an eight character password to verify his identity. Once the alarm is triggered it can be killed using a 12 character password. The door observation time for the main access door is set as 15 seconds, identity verification timer is set as 90 seconds and the time for the user to get

back into the home after stepping out leaving the main door open before changing home state (home state change timer) is set as 120 seconds.

CHAPTER 5 RESULTS

5.1 CHAPTER OBJECTIVES

This chapter discusses the result of the experiments conducted in chapter 4. The chapter is divided into four sections. Section 5.1 outlines the chapter objectives while section 5.2 discusses the result and accuracy of the device fingerprinting algorithms discussed in this thesis along with the mathematical modelling to calculate the entropy of the developed device fingerprint. Section 5.3 illustrates the efficiency of the logical sensing algorithms proposed in the thesis; section 5.4 discusses the efficiency and accuracy of the proposed behavioural prediction algorithms.

5.2 DEVICE FINGERPRINTING EXPERIMENT RESULT AND MATHAMATICAL MODELLING

5.2.1 Result of the proposed device fingerprint algorithm test

The test website www.fingerprintmydevice.com received 283 hits between May 2016 and July 2016, out of them 97 were unique devices. Devices that visited our test website belong to different countries like South Africa, India, Canada, United States etc. The test website saw visits from devices with a wide range of operating systems like, Windows XP, Windows 7, Windows 8, Windows 10, Mac OS and Linux in our test dataset. The machines that visited our test website used either Internet explorer, Firefox, Chrome or Safari.

Out of the 97 unique devices visited, the algorithm was able to uniquely identify 95 devices, which accounted for 97.93% of the total devices visited. Our algorithm failed to uniquely identify 2 machines. The 2 machines that our algorithm failed to identify had both flash and geolocation enabled and were identical in almost every aspect. They were machines from the University library with cloned hard disks so their OS, browsers, plugins, mime and all other browser specific and device specific information were identical. Moreover, their screen sizes, pixel depth and resolution were identical as well. Since both these machines were located very close in the University library their geolocation parameters were also indistinguishable. The graph given in Figure 5.1, shows the result.

Out of the 95 uniquely identified machines, 58 of them had their Geolocation enabled and our script was able to precisely determine their latitude and longitude. 71 out of the 95 machines had their flash enabled and we successfully collected the 40 flash parameters including system fonts from these devices. 34 machines had both Flash and Geolocation enabled. Figure 5.2, shows the number of devices that has enabled flash, geolocation and both.

The algorithm was able to successfully identify 97.93% of the machines that visited our test website using our device fingerprinting algorithm. A total of 14 browser specific parameters, 14 device specific parameters and 40 flash parameters were used to develop a device's fingerprint in an ideal case when all the parameters are available.

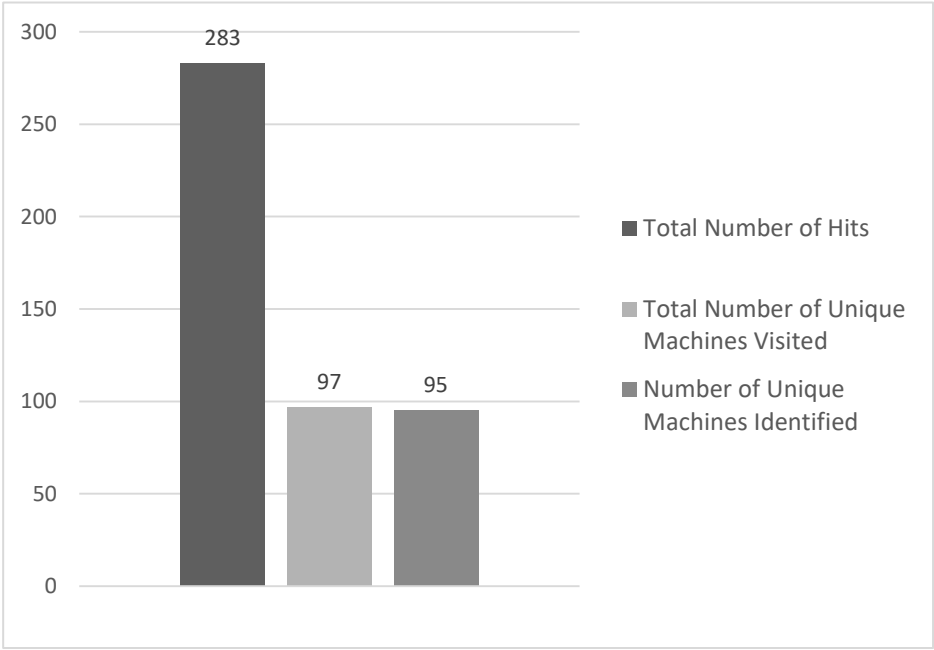


Figure 5.1. Shows the number of total hits, number of unique device's visited our website along with number of devices uniquely fingerprinted and identified.

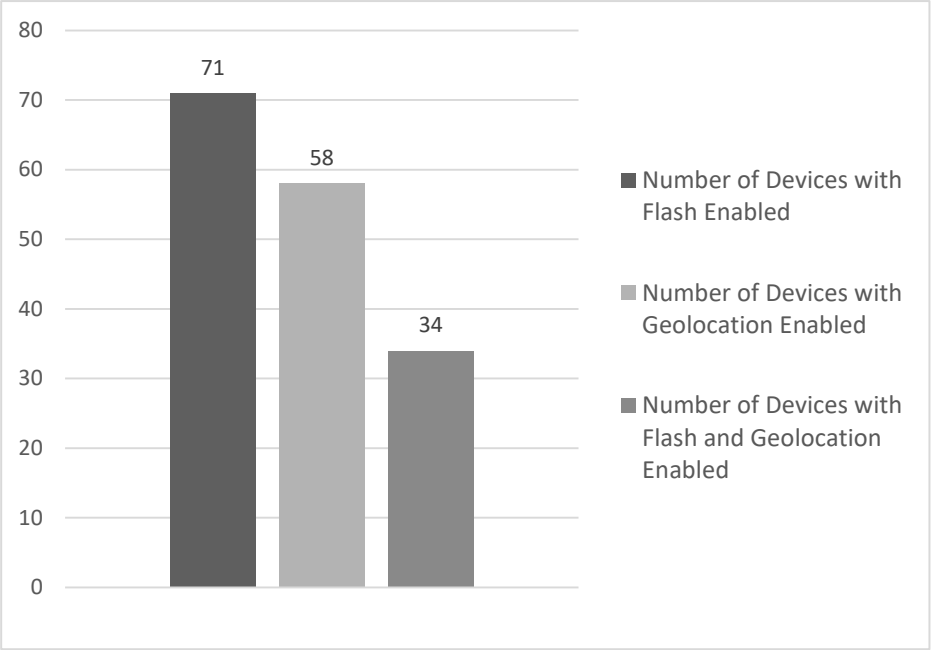


Figure 5.2. Shows the number of devices that has enabled Flash, Geolocation and both.

5.2.2 Result of the device fingerprint algorithm test

The information contained in a device fingerprint can be calculated using Entropy; Shannon entropy $H(X)$ of a discrete variable X with possible values $\{x_1, x_2 \ldots\ldots x_n\}$ is given by the equation:

$$H(X) = \sum_{i=1}^{n} \mathcal{P}(x_i) I(x_i) = -\sum_{i=0}^{n} \mathcal{P}(x_i)\, log_b\, \mathcal{P}(x_i) \qquad (5.1)$$

where $\mathcal{P}(x_i)$ is the probability of each value and $I(x_i)$ is the information content and 'b' is the base of the logarithm; Shannon Entropy is calculated with logarithm to the base 2.

When all parameters are available the fingerprint algorithm mentioned in this thesis gave an entropy of around 22.57 bits; it drops down to around 22.47 bits when the geographical location parameters are unavailable. It further drops to 21.25 bits when screen parameters are inaccessible and to 21.25 bits when user agent parameters are unavailable. Figure 5.3

shows the Shannon Entropy of our proposed fingerprinting algorithm when all fingerprinting parameters are available and various parameters are unavailable.

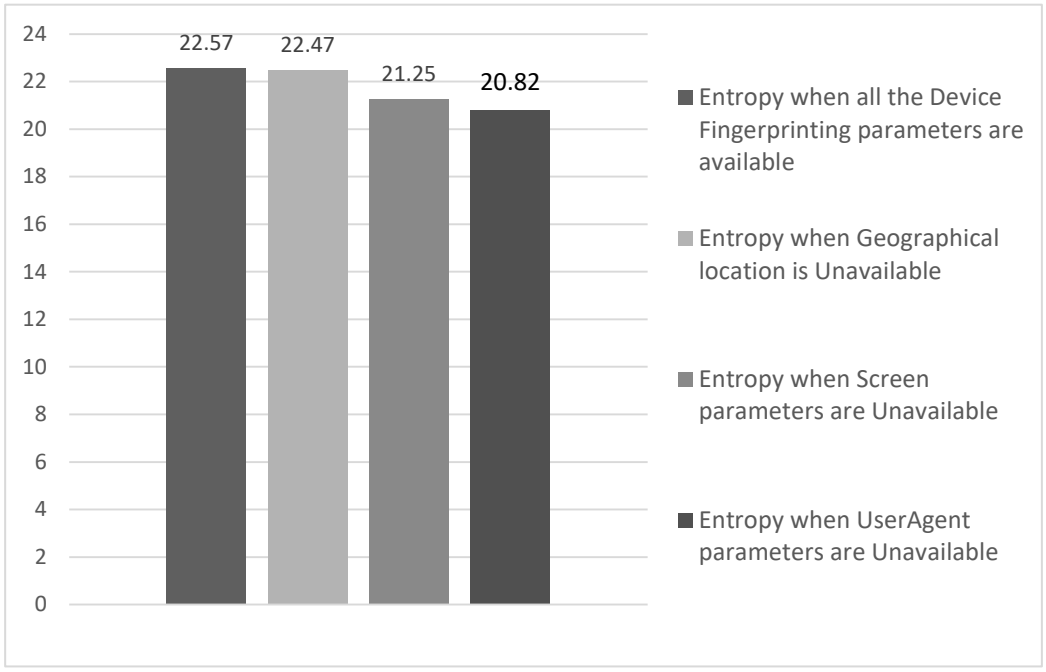


Figure 5.3. Shows the Entropy of the fingerprinting algorithm when all fingerprinting parameters are available and various parameters are unavailable.

5.3 LOGICAL SENSING EXPERIMENT RESULT

During the one month period the main access point changed state 305 times. The algorithm was able to detect all these and reduce them to 190 state changes by identifying and eliminating the intermediate state changes mentioned in Table 3.4. The most common state triggered was state 4 in Table 3.4, it was triggered 46 times. While state 17 was triggered 33 times making it the second most popular. State 13 was triggered 20 times making it the third most triggered state. State 6 and state 1 were triggered 12 and 11 times respectively. State 31 happened 6 times, State 19 was triggered 13 times, States 9 and 16 were triggered five times; states 7, 14, and 15 were triggered four times; states 5, 11, 20, 22 and 26 were triggered thrice; states 2, 8 and 12 were triggered twice; states 3, 10, 18, 21, 27 and 32 were triggered

only once during the one month time period. States 23, 24, 25, 28, 29 and 30 were not triggered during the one month time period.

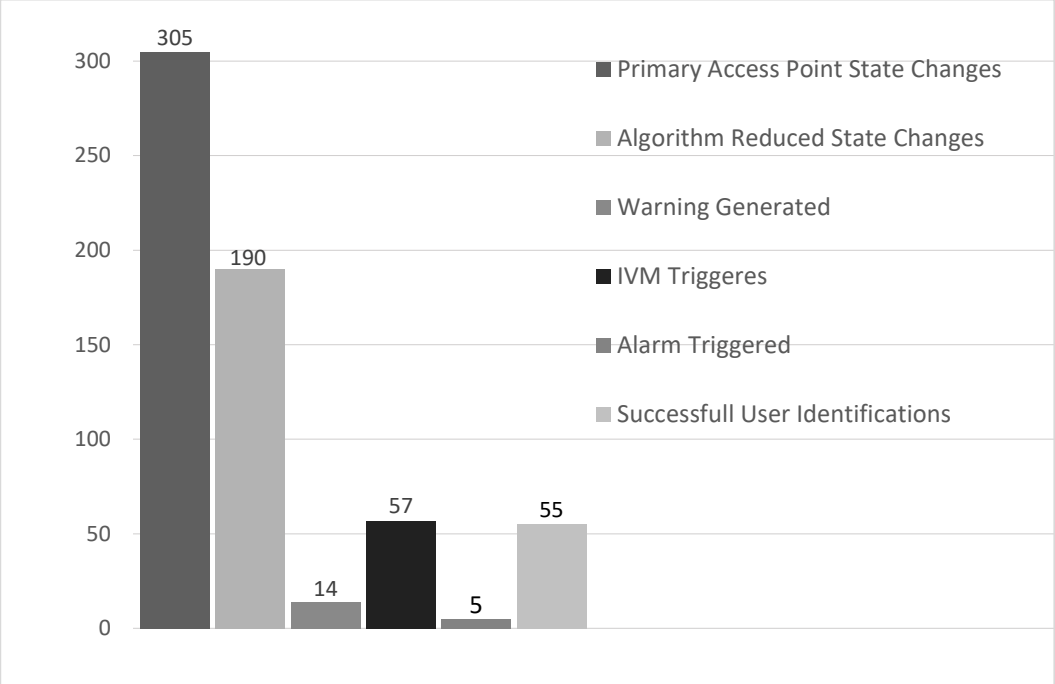


Figure 5.4. Graph showing the number of state changes for primary access point, algorithm reduced state changes, total number of warnings generated, IVM triggers, number of user identity verifications and number of alarms generated.

The algorithm generated 14 warnings, 8 regarding the open primary access point, 3 for not securing the secondary access point before leaving the home and another 3 warnings for opening the balcony door during day when the bed was occupied. Intruder alarm was triggered 5 times during the experiment, 4 were related to primary access points while one was related to secondary access points. IVM was triggered 59 times and user successfully verified his identity 55 times. Alarm was killed using the 12 character password five times. The state change timer was activated 23 times while the user re-entered the home before the state change timer expired 15 times, so the home state changed due to state change timer expiry 8 times. The graph in Figure 5.4 shows the number of state changes for primary access point, total number of warnings generated, IVM triggers, number of user identity

verifications and number of alarms generated. Figure 5.5 shows the number of state triggers for frequently triggered states namely State 1, 4, 6, 9, 13, 17, 19 and 31.

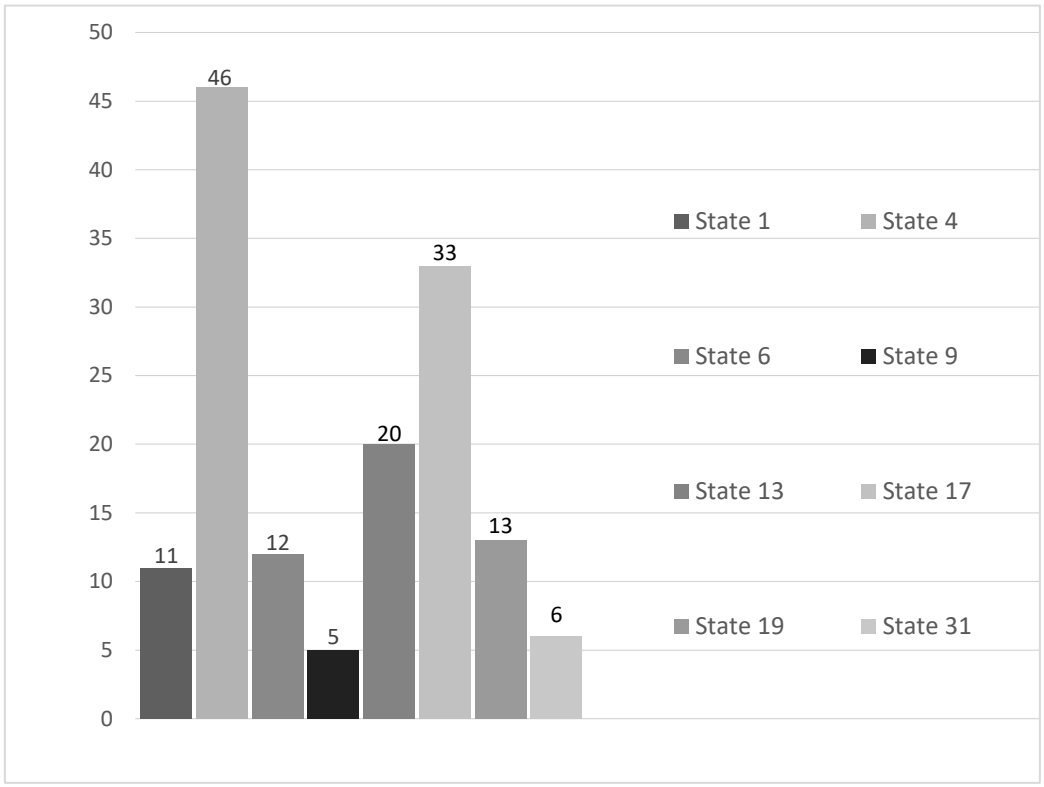


Figure 5.5. Graph showing number of state triggers for State 1, State 4, State 6, State 9, State 13, State 17, State 19 and State 31 mentioned in Table 3.4 during the experiment.

Secondary access points changed states 56 times; balcony door changed state 27 times and window state was changed 29 times. All of the 27 times balcony door changed state the user was at home but 3 of them happened when the user was in bed, so these 3 times warnings were generated and IVM was activated along with the identity verification timer, which was reset when the user conformed their identity. Once the open balcony door was closed due to wind when the user was in bed, so no identity verification mechanism was initiated. The home was occupied all 29 times when the state of the balcony window was changed. Once the window was opened from the outside when the user was in bed during night; the intrusion defence mechanisms (audible alarm) were triggered without waiting for any user identity verifications. Figure 5.6 shows the total number of state changes for secondary access points,

balcony door and window state changes, secondary access point triggered IVM, warnings and alarms generated due to secondary access points.

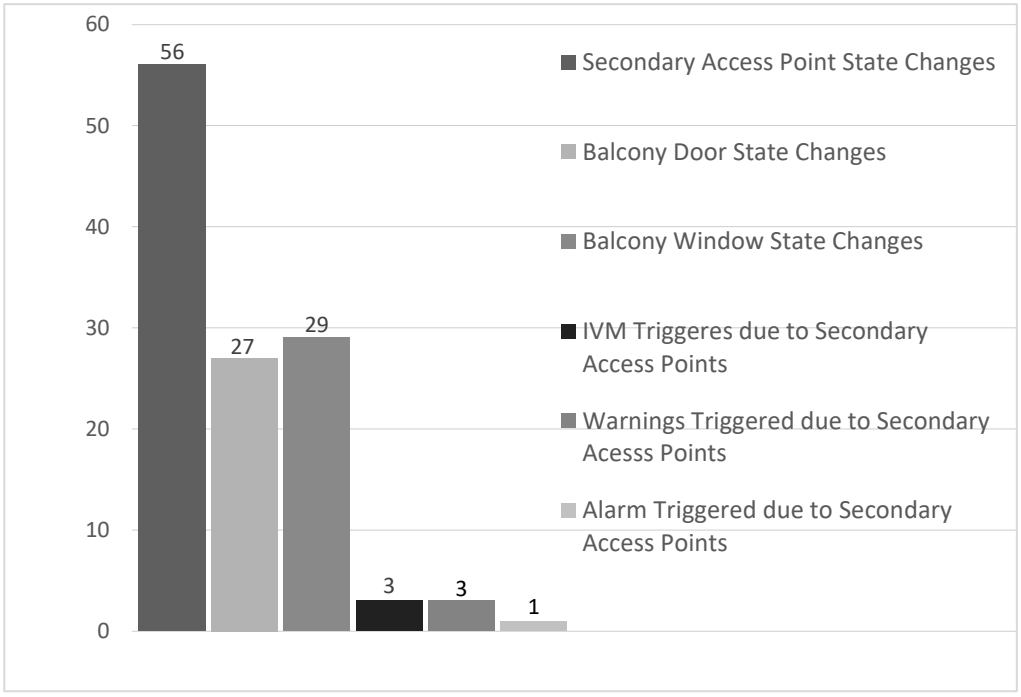


Figure 5.6. Graph showing the number of state changes for secondary access points, balcony door state changes, balcony window state changes, IVM triggers due to secondary access points, warnings generated due to secondary access points and number of alarms generated due to secondary access points.

MQ 9 sensor values are only considered after the sensor is allowed to be stabilized for 24 hours. Voltage measured when there is no carbon monoxide detected varied from 0.23V to 0.21V, calculated Rs value ranged from 20.73 to 22.27 and Rs to Ro ratio ranged from 10.11 to 10.86. Multiple matches are lit in an enclosed space near the sensor to determine sensor values during the presence of fire. Measured voltage when there is fire and carbon monoxide detected varied from 0.26V to 0.28V, calculated Rs value ranged from 16.85 to 18.24 and Rs to Ro ratio ranged from 8.22 to 8.9.

During fire the twelve second average value of temperature and humidity sensors were also noted. Before the fire the twelve second average temperature inside the apartment was 24^{o}C,

the average temperature increased by 2.5ºC to 26.5ºC within 12 seconds and further increased to 30ºC within 36 seconds. 12 second average humidity in the apartment before fire was 32% it decreased by 3% to 29% within 12 seconds of the fire and after 36 seconds it further declined to 25.5%. Figure 5.7 shows the measured voltage, Rs value, Rs to Ro ratio of the of MQ 9 sensor along with temperature and humidity sensor values under normal operational conditions and during fire.

When the bed was unoccupied, the force sensor at the bed top gave average readings between 0 and 2 Newton, the force sensor at the bed bottom gave average readings between 0 and 3 Newton. When objects like stack of books or some heavy boxes are placed on various areas of the bed, the average force at the shoulder region was between 1 and 3 Newton and average force at the abdominal region was between 2 and 4 Newton. When the user occupied the bed, the average force sensor values deployed at the shoulder region varied between 5 and 8 Newton, while force sensor values deployed at the abdominal region shifted between 7 and 10 Newton.

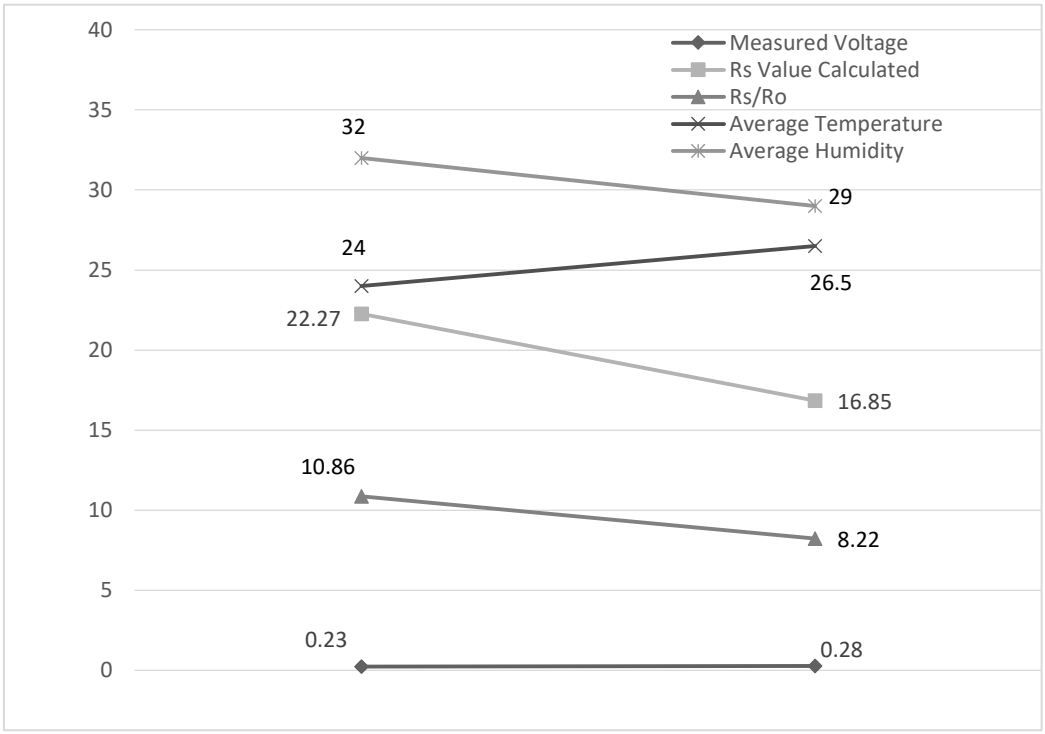


Figure 5.7. Graph showing Measured Voltage, Rs Value, Rs to Ro Ratio of the of MQ 9 sensor along with temperature and humidity values under normal operational conditions and during fire.

5.4 MACHINE LEARNING FOR BEHAVIOUR PREDICTION, MATHAMATICAL MODELLING AND EXPERIMENT RESULT

The first seven weeks of the behaviour prediction algorithm testing experiment were used for training or learning normal user behaviour. During the seven weeks the home was also subjected to intrusion attempts by a trusted friend, who was asked to break into the home unnoticed. He is asked to use the primary or secondary access points to break-in at any time during day or night. The gathered data is then added to the security algorithm and implemented in the apartment for 2 weeks. In order to avoid a serious security situation like neighbours calling the police during a simulated break-in attempt, the friend who is asked to do the break-in is given keys to the home, as any skilled robber would pick the lock with ease. The person simulating the break-in is also unaware of the parameters considered for behaviour prediction.

5.4.1 Machine Learning for Behaviour Prediction Algorithm

During machine learning the algorithm identifies the time at which a user enters the home during day and night. User behaviour during day and night are looked at separately, so user behaviour from 06:30 am until 5:30 pm is classified as user's day behaviour while user behaviour between 5:31pm and 6:29 am is considered as user's night behaviour. Over the seven week training period, during day the empty home became occupied 54 times. Out of the 54 times when someone walked into an empty home, 47 times it was legitimate user while other 7 times was intrusion attempts. In the 47 times a legitimate user came home, 39 of the 47 times the user came home between 2:30 pm and 4:30 pm. So the time between 2:30 pm and 4:30 pm is referred as the 'day prime time'. Figure 5.8 shows various time of the day when someone entered the occupied home. Rest of the 8 times, user came home some other time during day. From the figure, it is clear that there was considerably more number of entries during day prime time compared to other times of the day.

Out of the seven intrusion attempts happened during the day, only one of them happened during day prime time and the rest of the six attempts happened during other times of the day. Six times an intruder tried to break into the home when it was occupied. Twice he

opened the primary access point from the outside when the home was occupied, once he opened the primary access point and the secondary access point when the user was in bed, in all these case the intrusion defensive mechanism was activated and intruder alarm was triggered based on the security algorithm proposed for logical sensing. Twice the secondary access point was opened from the outside during the day, during which time a warning was triggered.

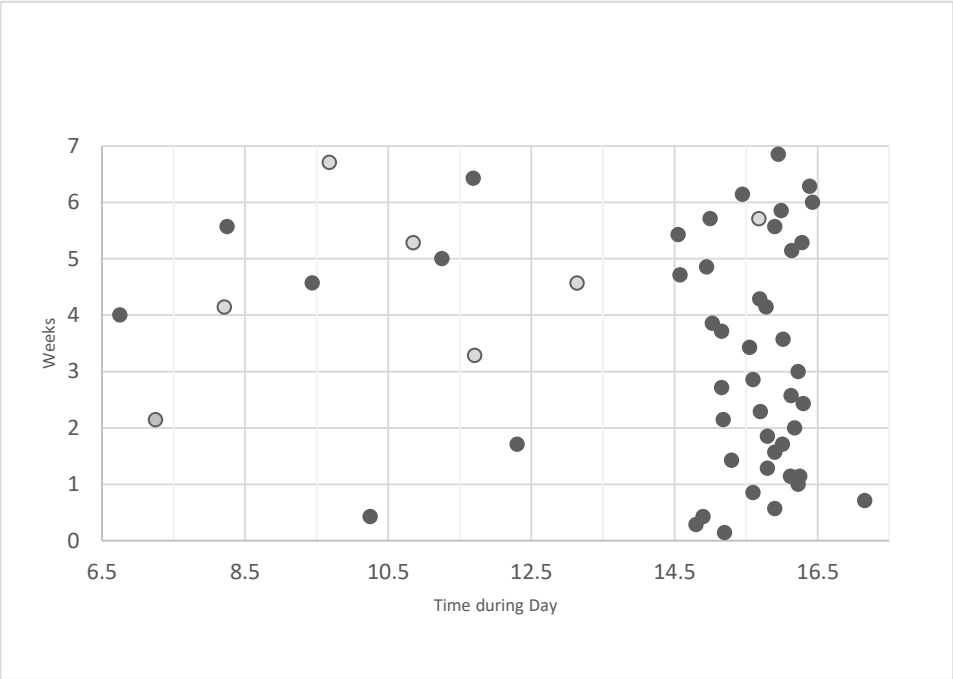


Figure 5.8. Represents various times at which user access their empty home (shown using dark spots) along with intruder accessing the home (shown using light spots).

During day the algorithm identifies the number of times when a key is placed in the key-bowl soon after the user had entered the home. Out of the 47 times the legitimate user entered the empty home 45 times he placed his keys in the key-bowl. Thrice he was preoccupied with other things and did not place the apartment keys in the key-bowl. Out of the three times, twice the user forgot to place the key in the bowl during day prime time and once during other time. During the 7 times the intruder entered the home not once he placed the keys inside the bowl. Moreover, a real intruder may not have the apartment key with him, he usually picks the lock or finds some other way into the home. On average, it took the user

about 15 seconds to place the apartment keys in bowl. So the algorithm takes 15 seconds to identify user's key placement behaviour.

During day user's light behaviour dramatically varied. The light was switched on only 11 out of the 47 times a legitimate user entered the home during day time. Seven times the lights were switched on when the intensity of lights in the apartment is low all light sensor values indicated light intensity less than 350. Rest of the four times, the light intensity in the apartment was high (greater than 350) but the user still switched on the lights. So day time light behaviour is disregarded for behaviour prediction.

While learning user behaviour the algorithm predicts the time at which a legitimate user comes home more frequently during night called the 'night prime time'. During night, between 5:31pm and 6:29 am the empty home was occupied 59 times. Out of the 59 entries, 32 of them were legitimate user accessing the home and 27 were intrusions. 18 of the 32 legitimate user entries to the home occurred between 10:00 pm and 12:00 pm, so the two hour window between 10:00 pm and 12:00 is identified as the 'night prime time'. Rest of the 14 legitimate user entries occurred some other time during the night. 3 of the 27 intrusion attempts also occurred during night prime time while 24 of the intrusion attempts happened during other times of the night. Out of the 24 intrusion attempts occurred during other rimes of the night twice the intruder was able to switch on the lights near the front door.

In addition to the 27 intrusion attempts, twice during night the primary access point was opened from the outside when the home was occupied and user was not in bed and once when the user was in bed, once secondary access point was opened from the outside during the night when the user was in bed and the home was occupied and once when user was not in bed. In all the above 5 cases intrusion defensive mechanism was activated and alarm was sounded based on the logical security algorithm proposed.

During night, out of the 32 times when the legitimate user entered the home, 27 times he switched on the light near the front door within six seconds, while 23 times light near the balcony door or inside lights were switched on within 12 seconds. 17 times both lights near

the front door and inside the apartment were switched on. Twice when the legitimate user entered the home none of the lights inside the apartments were switched on within the time expected time frame i.e. lights near the front door were not switched on within 6 seconds and lights near the balcony door or inside lights were not switched on within 12 seconds (Once each during night prime time and other time of the night). During the 27 intrusion attempts twice the intruder switched on the lights near the front door within 6 seconds upon opening the front door. These two times did not happen during night prime time intrusions.

11 of the 18 times when legitimate user entered the home during night prime time, lights near the front door were switched on within 6 seconds and lights inside the apartment were switched on within 12 seconds. 15 of the 18 times when legitimate user entered the home during night prime, time lights near the front door were switched on within 6 seconds. 14 of the 18 times when legitimate user entered the home during night prime time lights inside the apartment was switched on within 12 seconds. 6 out of the 14 times when legitimate user entered the home at night during other times, lights near the front door were switched on within 6 seconds and lights inside the apartment were switched on within 12 seconds. 12 out of 14 times when legitimate user entered the home at night during other times, lights near the front door were switched on within 6 seconds. 9 out of the 14 times when legitimate user entered the home at night during other times lights inside the apartment was switched on within 12 seconds.

The key was placed in the key-bowl 27 out of the 32 times when a legitimate user entered the home during night. 5 times during the night the user took more than 15 seconds to place the key in the bowl or did not place the key in the bowl at all. Once the user forgot to place the key in the bowl during night prime time and once during other time during the night when the lights near the front door and lights inside the apartment were switched on. Similarly, once user forgot to place the key in the key-bowl during night prime time and once during other time when lights near the front door were switched on. The user also forgot to place the key in the bowl once during night prime time and once during other time during night when lights inside the apartment was switched on.

Once the normal user behaviour was identified a naïve Bayesian network is constructed and integrated into the behaviour prediction algorithm. The behaviour prediction algorithm observes user actions when he enters an empty home. A user's light behaviour is identified within 12 seconds and his key behaviour is recognized within 15 seconds. So the proposed algorithm takes 15 seconds to identify and distinguish between normal and abnormal user behaviours.

5.4.2 Mathematical Modelling

Bayesian probabilities for user behaviour during night can be calculated by the equation:

$$\begin{aligned}\mathcal{P}(N, L_{1\&3}, L_2, K, T) \\ = \mathcal{P}(N|L_{1\&3}\,)\,\mathcal{P}(L_{1\&3}\,) \times \mathcal{P}(N|L_2)\mathcal{P}(L_2) \times \mathcal{P}(N|K)\mathcal{P}(K) \\ \times \mathcal{P}(N|T)\mathcal{P}(T)\end{aligned} \tag{5.2}$$

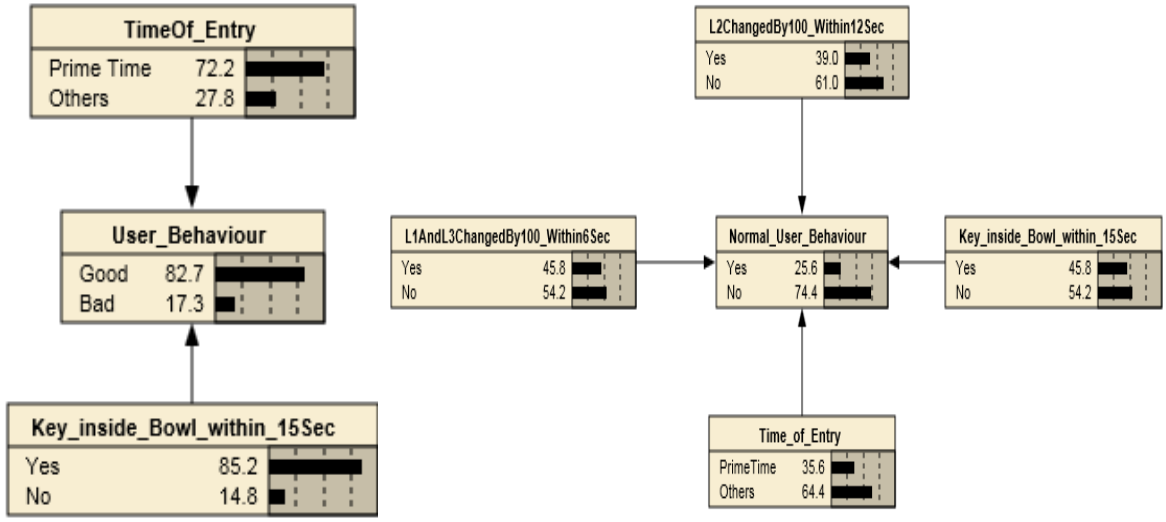
Where 'N' is the normal user behaviour '$L_{1\&3}$ ' is the status of lights near the front door, 'L_2' is the status of the light inside the apartment, 'K' is placement of the key in the key-bowl during night , 'T' is the time at which user comes home during night.

Bayesian probabilities for user behaviour during day can be calculated by the equation:

$$\mathcal{P}(N, K, T) = \mathcal{P}(N|K)\mathcal{P}(K) \times \mathcal{P}(N|T)\mathcal{P}(T) \tag{5.3}$$

Where 'N' is the normal user behaviour 'K' is placement of the key in the key-bowl during day, 'T' is the time at which user comes home during day. All the Bayesian variables discussed here are binary whose value are obtained from the sensors deployed at various parts of the home.

Netica is an open source software package to work with Bayesian belief network. The work utilizes Netica to represent the Bayesian belief networks in this thesis. Figure 5.9 (a) shows the structure of the Bayesian network during the day and Figure 5.9 (b) shows the structure of the Bayesian network during the night.



(a) (b)

Figure 5.9. (a) Shows the structure of the Bayesian network during the day in Netica (b) shows the structure of the Bayesian network in Netica during the night.

The table, Table 5.1 gives all the possible combinations of parameters considered for behaviour prediction during day and the corresponding figure showing the resulting probability of Good and Bad User Behaviour represented using Netica; Table 5.2 gives all the possible combinations of parameters considered for behaviour prediction during night and the corresponding figure showing the probability of resulting Good and Bad User Behaviour represented using Netica.

Table 5.1. Shows the possible combinations of parameters considered for behaviour prediction during day and the corresponding figure showing the resulting probability of good and bad user behaviours.

No	Time of Entry	Keys Inside the Bowl within 15 Seconds	User Behaviour
1	Prime Time	Yes	Figure 5.10 (a)
2	Prime Time	No	Figure 5.10 (b)
3	Other Time	Yes	Figure 5.10 (c)
4	Other Time	No	Figure 5.10 (d)

The probability of good and bad user behaviour during the day given in Figure 5.10 is calculated using the Bayesian probability equation (5.3).

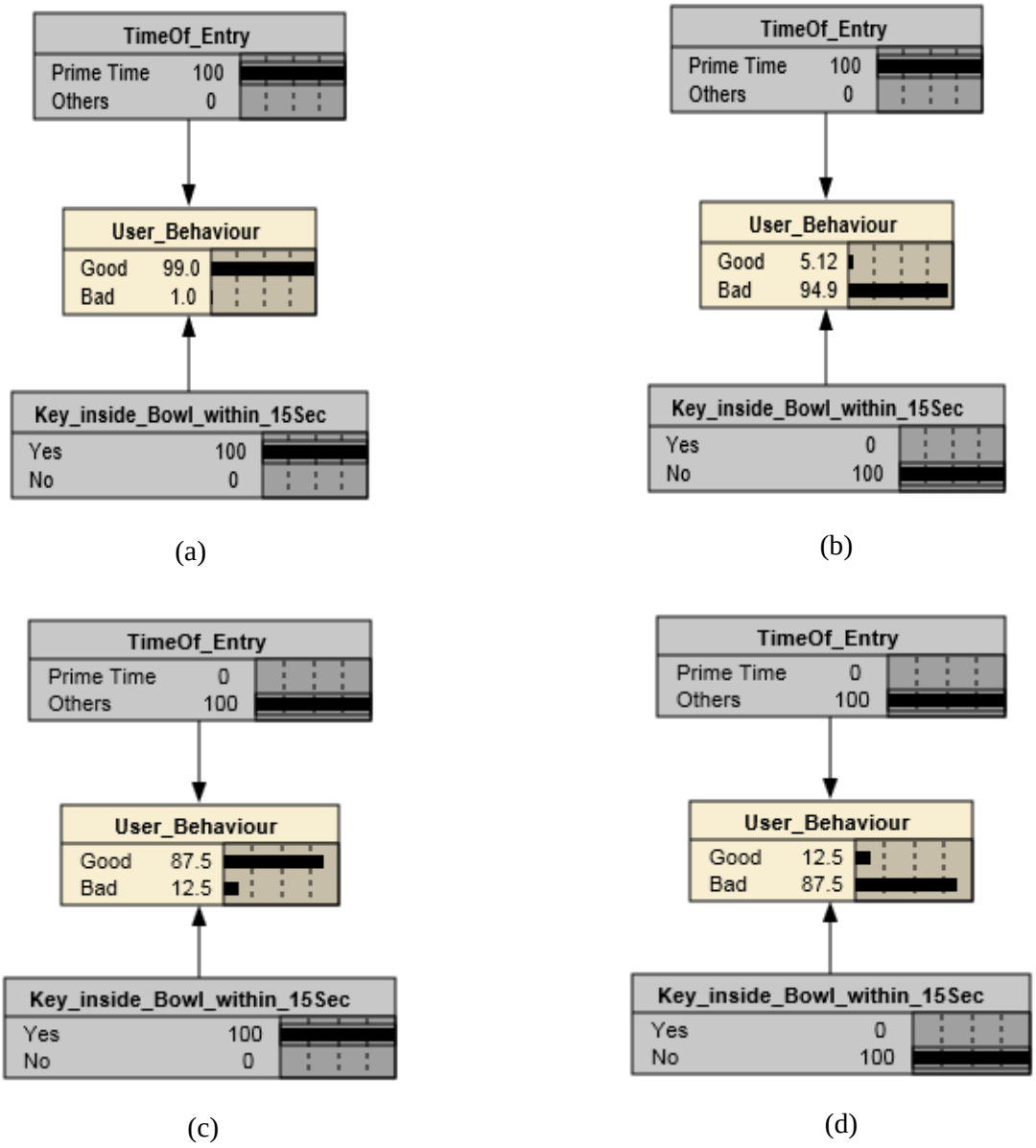


Figure 5.10. (a) shows the structure of the Bayesian network during the day in Netica when user access the home during primetime and keys are placed inside the bowl within 15 seconds (b) shows the structure of the Bayesian network in Netica during the day in Netica when user access the home during primetime and keys are not placed inside the bowl within 15 seconds (c) shows the structure of the Bayesian network during the day in Netica when user access the home during other times and keys are placed inside the bowl within 15 seconds (d) shows the structure of the Bayesian network during the day in Netica when user access the home during other times and keys are not placed inside the bowl within 15 seconds.

Table 5.2. Shows the possible combinations of parameters considered for behaviour prediction during night and the corresponding figures showing the resulting probability of good and bad user behaviours.

No	L1andL3 changed by 100 within 6 sec.	L2 changed by 100 within 12 sec.	Time of Entry	Keys inside the bowl within 15 sec.	User Behaviour
1	Yes	Yes	Prime Time	Yes	Figure 5.11
2	No	Yes	Prime Time	Yes	Figure 5.12
3	Yes	No	Prime Time	Yes	Figure 5.13
4	No	No	Prime Time	Yes	Figure 5.14
5	Yes	Yes	Other Time	Yes	Figure 5.15
6	No	Yes	Other Time	Yes	Figure 5.16
7	Yes	No	Other Time	Yes	Figure 5.17
8	No	No	Other Time	Yes	Figure 5.18
9	Yes	Yes	Prime Time	No	Figure 5.19
10	No	Yes	Prime Time	No	Figure 5.20
11	Yes	No	Prime Time	No	Figure 5.21
12	No	No	Prime Time	No	Figure 5.22
13	Yes	Yes	Other Time	No	Figure 5.23
14	No	Yes	Other Time	No	Figure 5.24
15	Yes	No	Other Time	No	Figure 5.25
16	No	No	Other Time	No	Figure 5.26

The probability of good and bad user behaviour during the day given in Figure 5.11to Figure 5. 26 is calculated using the Bayesian probability equation (5.2).

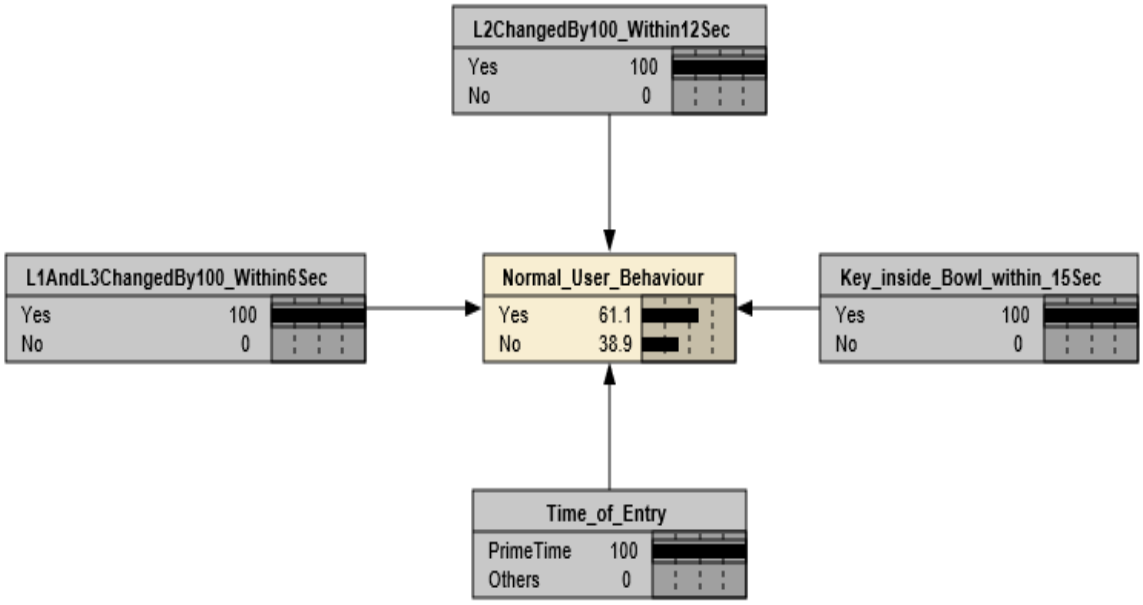


Figure 5.11. Shows the structure of the Bayesian network during night in Netica when user access the home during primetime and keys are placed inside the bowl within 15 seconds and lights near the front door and lights inside the home are switched on within 6 and 12 seconds respectively.

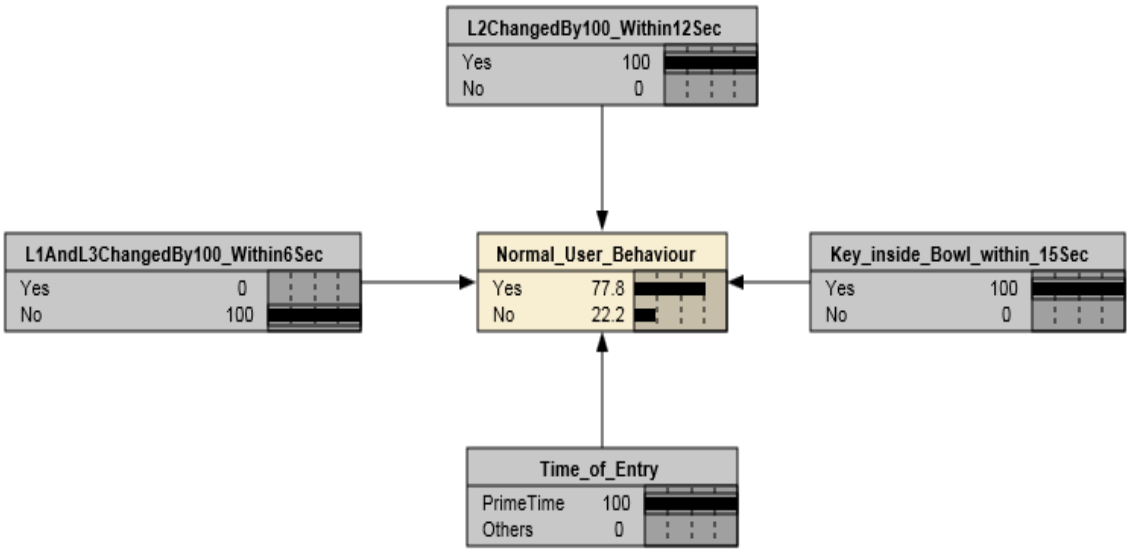


Figure 5.12. Shows the structure of the Bayesian network during night in Netica when user access the home during primetime and keys are placed inside the bowl within 15 seconds and lights near the front door is not switched on within 6 seconds and lights inside the home are switched on within 12 seconds.

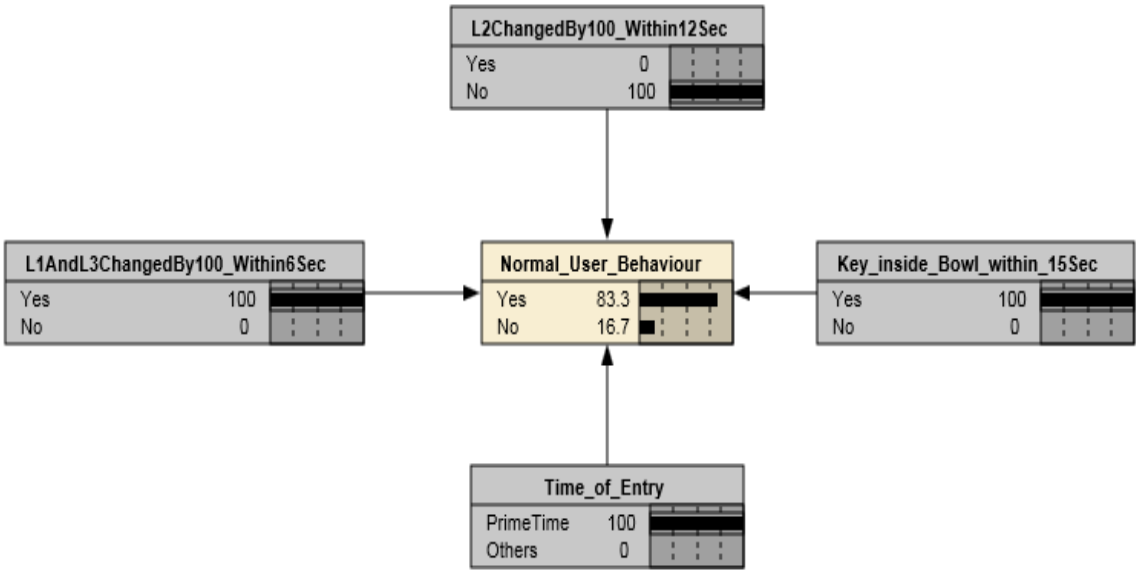


Figure 5.13. Shows the structure of the Bayesian network during night in Netica when user access the home during primetime and keys are placed inside the bowl within 15 seconds and lights near the front door is switched on within 6 seconds and lights inside the home are not switched on within 12 seconds.

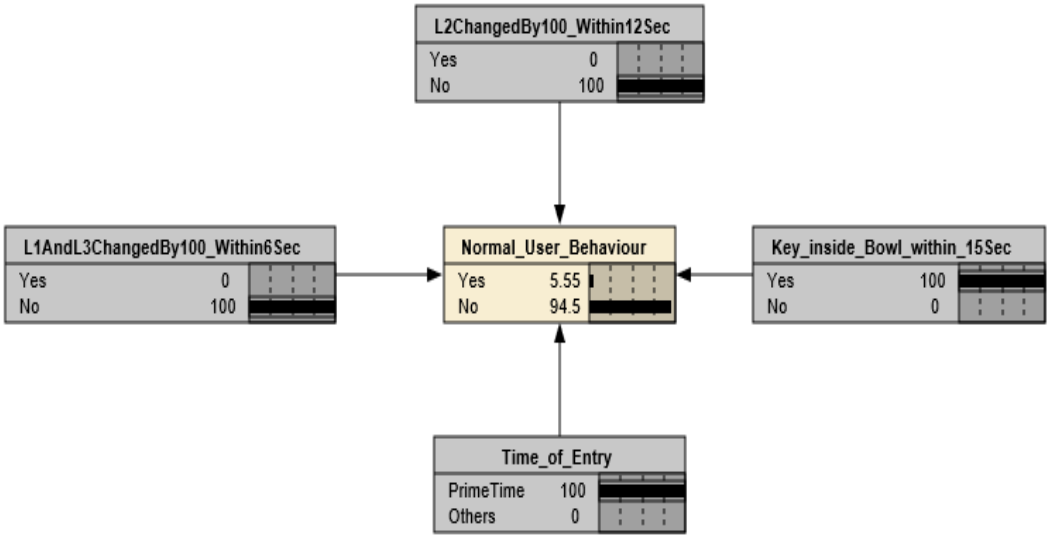


Figure 5.14. Shows the structure of the Bayesian network during night in Netica when user access the home during primetime and keys are placed inside the bowl within 15 seconds lights near the front door and lights inside the home are not switched on within 6 and 12 seconds respectively.

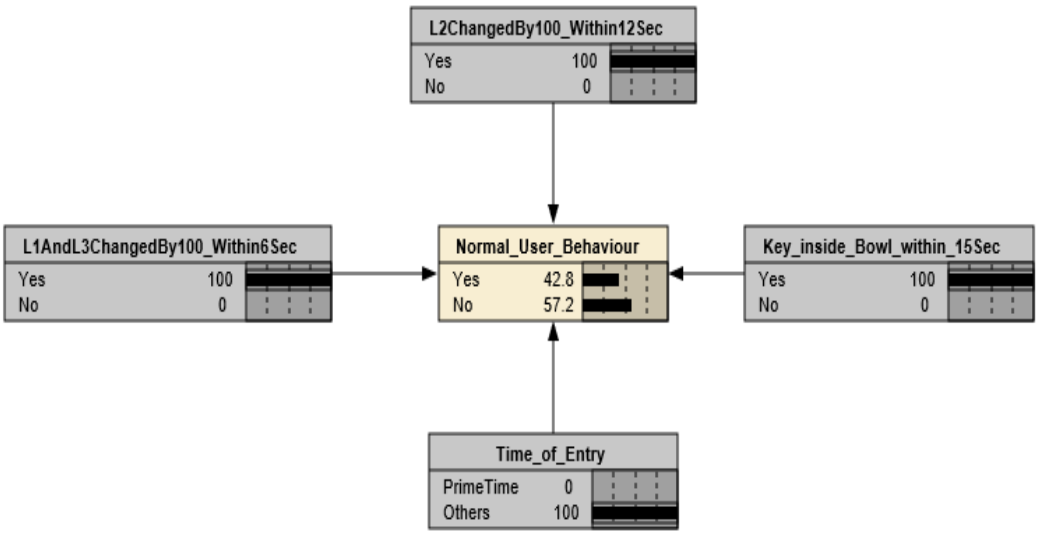


Figure 5.15. Shows the structure of the Bayesian network during night in Netica when user access the home during other times and keys are placed inside the bowl within 15 seconds, lights near the front door and lights inside the home are switched on within 6 and 12 seconds respectively.

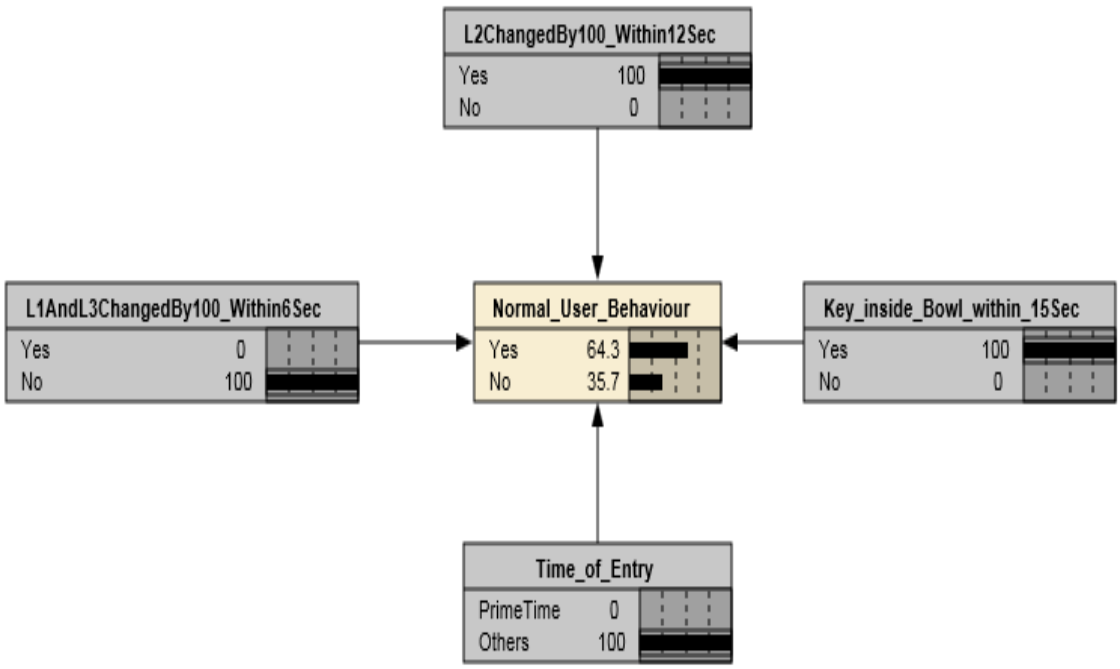


Figure 5.16. Shows the structure of the Bayesian network during night in Netica when user access the home during other time and keys are placed inside the bowl within 15 seconds and lights near the front door is not switched on within 6 seconds and lights inside the home are switched on within 12 seconds.

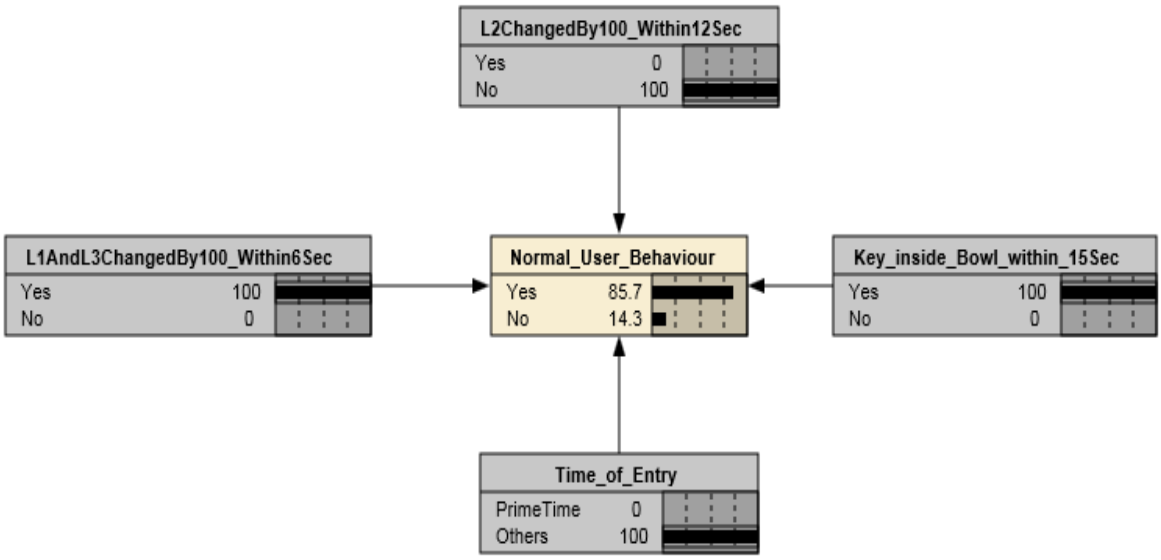


Figure 5.17. Shows the structure of the Bayesian network during night in Netica when user access the home during other time and keys are placed inside the bowl within 15 seconds and lights near the front door is switched on within 6 seconds and lights inside the home are not switched on within 12 seconds.

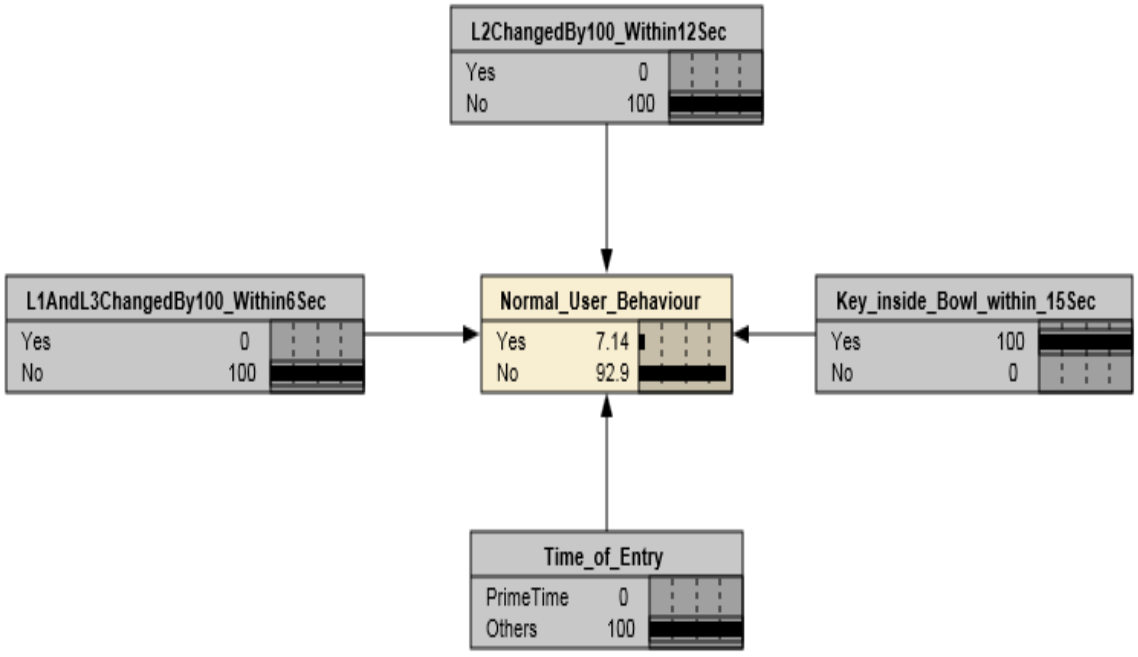


Figure 5.18. Shows the structure of the Bayesian network during night in Netica when user access the home during other time and keys are placed inside the bowl within 15 seconds, lights near the front door and lights inside the home are not switched on within 6 and 12 seconds respectively.

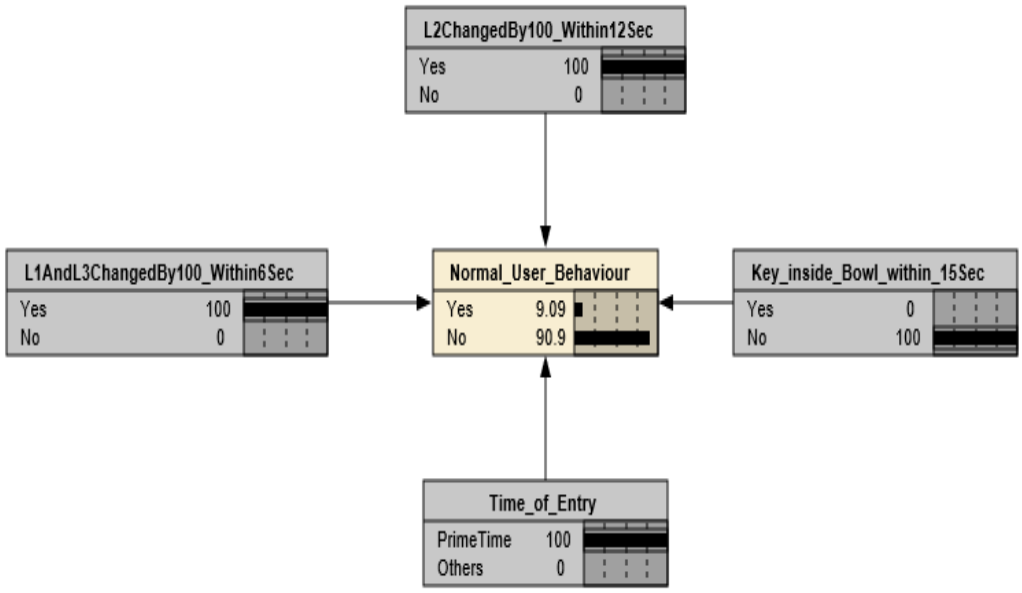


Figure 5.19. Shows the structure of the Bayesian network during night in Netica when user access the home during primetime and keys are not placed inside the bowl within 15 seconds and lights near the front door and lights inside the home are switched on within 6 and 12 seconds respectively.

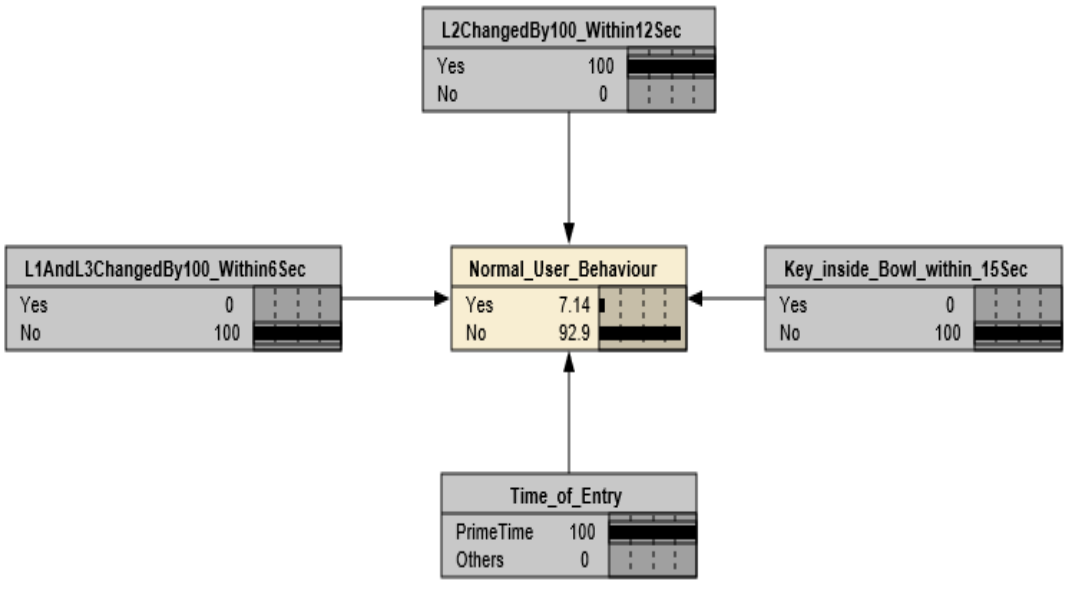


Figure 5.20. Shows the structure of the Bayesian network during night in Netica when user access the home during primetime and keys are not placed inside the bowl within 15 seconds and lights near the front door is not switched on within 6 seconds and lights inside the home are switched on within 12 seconds.

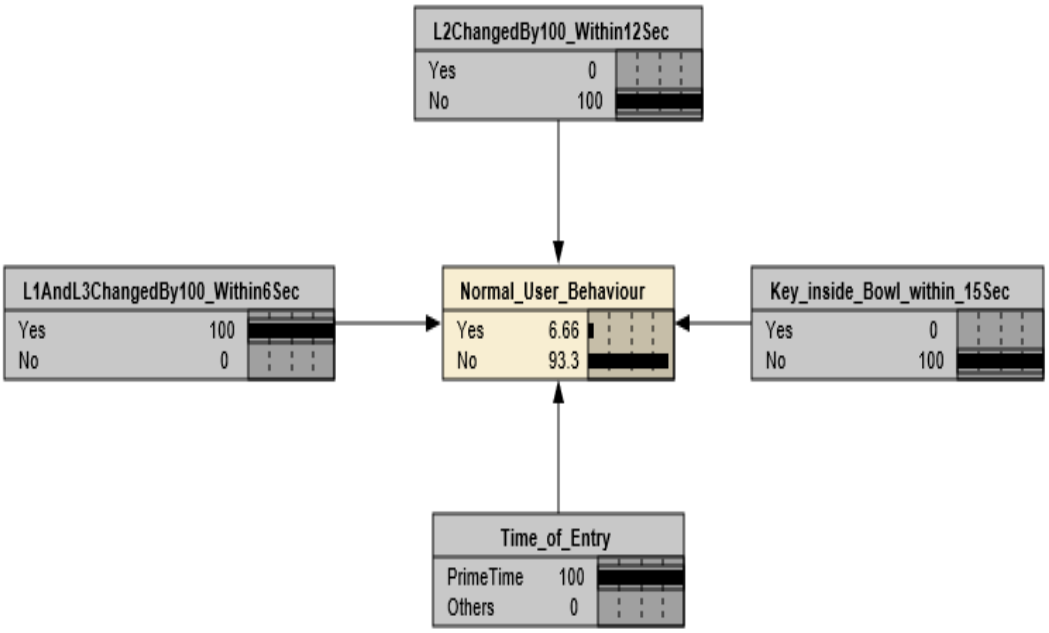


Figure 5.21. Shows the structure of the Bayesian network during night in Netica when user access the home during primetime and keys are not placed inside the bowl within 15 seconds and lights near the front door is switched on within 6 seconds and lights inside the home are not switched on within 12 seconds.

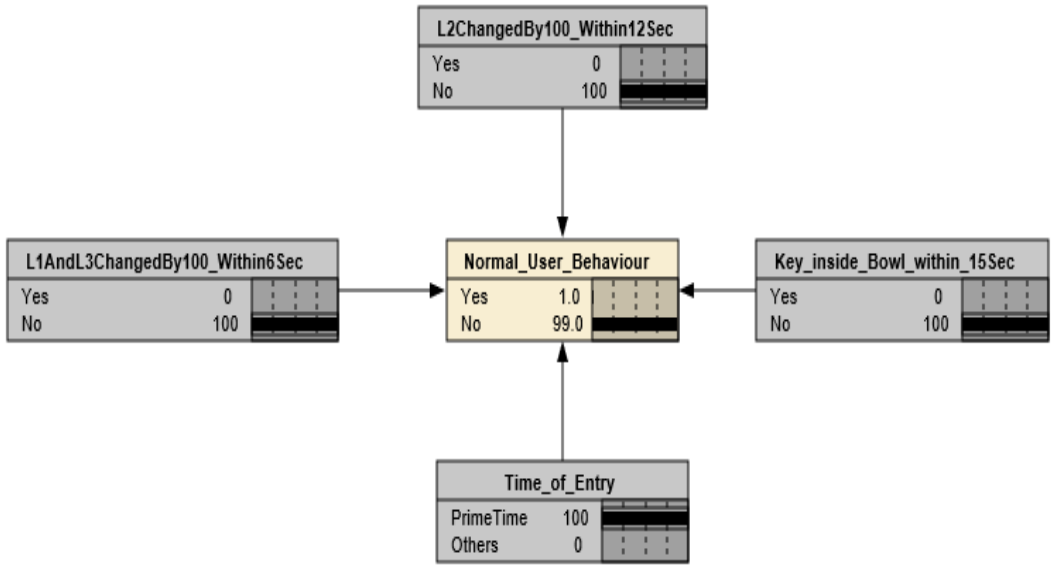


Figure 5.22. Shows the structure of the Bayesian network during night in Netica when user access the home during primetime and keys are not placed inside the bowl within 15 seconds lights near the front door and lights inside the home are not switched on within 6 and 12 seconds respectively.

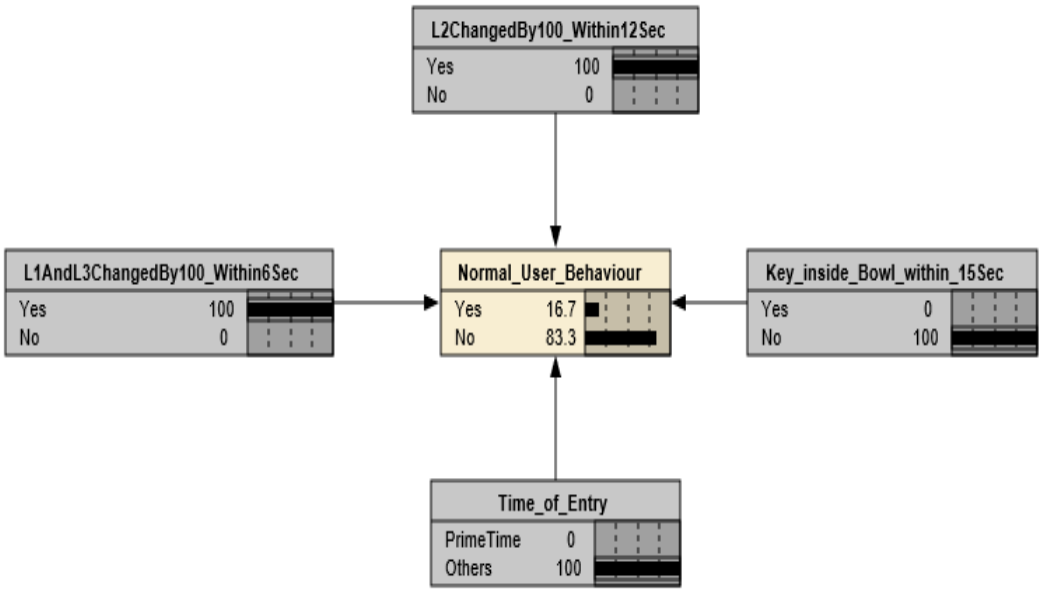


Figure 5.23. Shows the structure of the Bayesian network during night in Netica when user access the home during other time and keys are not placed inside the bowl within 15 seconds and lights near the front door and lights inside the home are switched on within 6 and 12 seconds respectively.

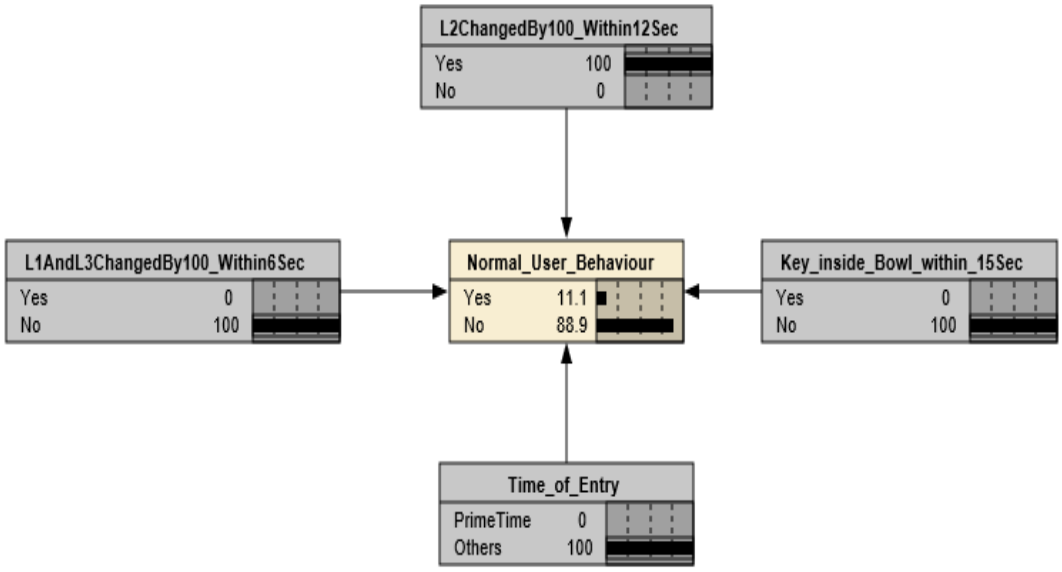


Figure 5.24. Shows the structure of the Bayesian network during night in Netica when user access the home during other time and keys are not placed inside the bowl within 15 seconds and lights near the front door is not switched on within 6 seconds and lights inside the home are switched on within 12 seconds.

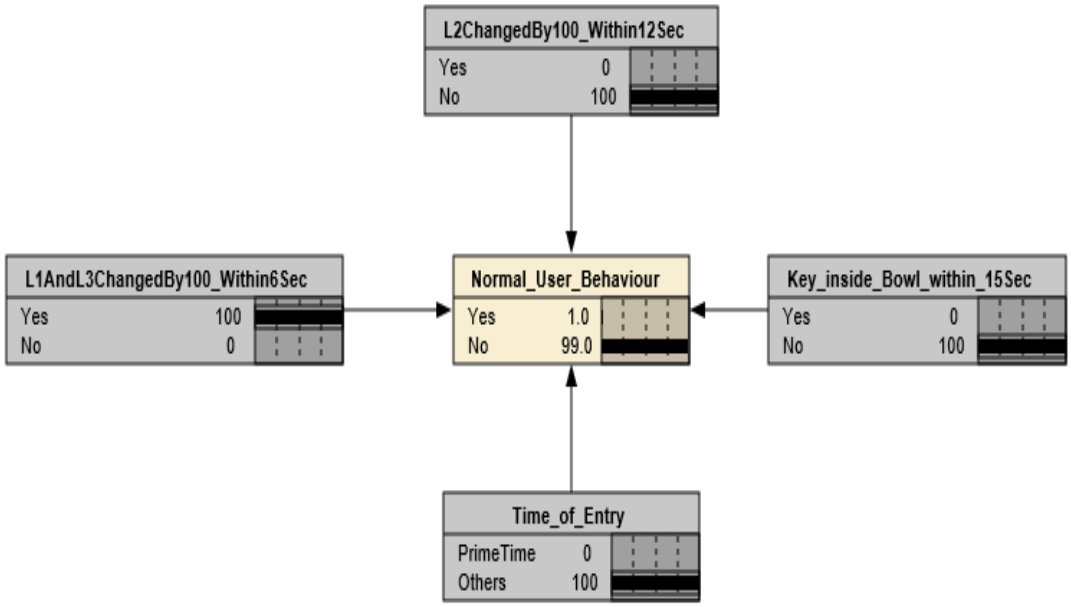


Figure 5.25. Shows the structure of the Bayesian network during night in Netica when user access the home during other time and keys are not placed inside the bowl within 15 seconds lights near the front door and lights inside the home are not switched on within 6 and 12 seconds respectively.

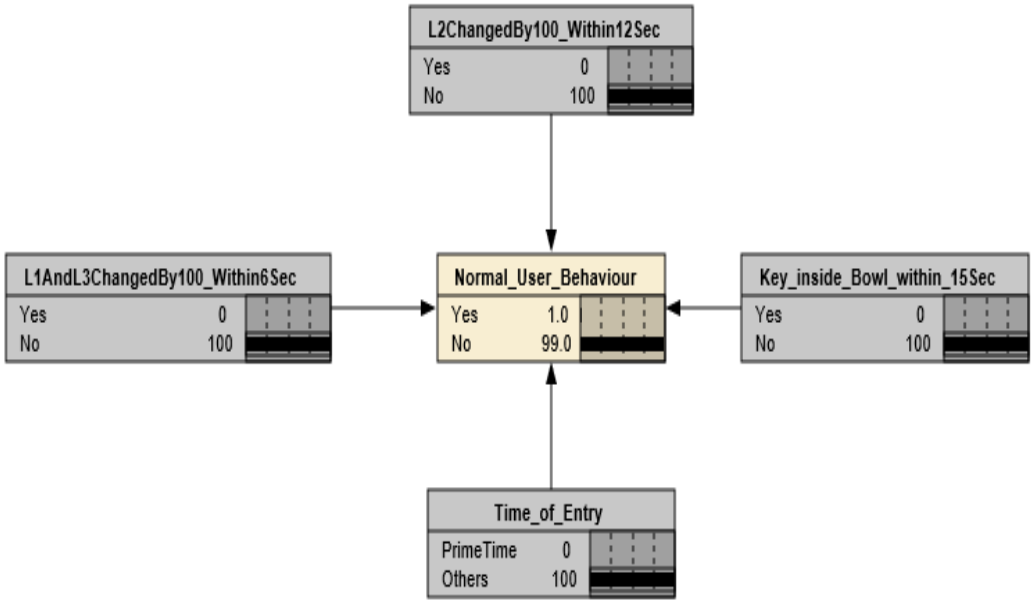


Figure 5.26. Shows the structure of the Bayesian network during night in Netica when user access the home during other time and keys are not placed inside the bowl within 15 seconds lights near the front door and lights inside the home are not switched on within 6 and 12 seconds respectively.

5.4.3 Behaviour Prediction Experiment Results

The behaviour prediction parameters analysed and identified during training was implemented as a behaviour prediction algorithm in the apartment for 2 weeks, during which the user entered the empty home 24 times in total, 13 times during the day (between 06:30 am and 5:30 pm) and 11 times during night (between 5:31pm and 6:29 am) . Figure 5.27 shows the total number of user entries to the home, number of day and night entries, number of warnings generated along with number of alarms triggered. During the 13 day time entry to the home, 10 of them happened during day prime time while rest of the 3 entries happened during the other times of the day. During the 13 times user entered the empty home during the day 12 times he placed the apartment keys inside the key-bowl within 15 seconds of entry to the home. During those 12 times the algorithm gave user 90 seconds to verify his identity. Once the user took more than 15 seconds to place the keys in the bowl, so the algorithm triggered the warning and informed the user about the abnormal user behaviour and the identity verification time was reduced to 45 seconds.

During the night the user entered the home 11 times out of those 8 times were during night prime time and 3 times were during other time of the night. The user switched on the lights near the front door 8 times within 6 seconds while lights inside the apartment was switched on within 12 seconds twice. Once lights near the front door and lights inside the apartment was switched on within 6 and 12 seconds respectively. One time the user did not switch on the lights near the front door but went into the home and switched on the lights inside the apartment within 12 seconds of entering the home. The key was placed in the key-bowl all the 11 times the user entered during the night. Once the user took more than 12 seconds to switch on any of the lights in the apartment after entering, so a warning about a possible intrusion was generated and the identity verification time was reduced to 45 seconds. Figure 5.28 shows the number of prime time entries during the night, number of other time entries to the home during night, number of times apartment keys are placed into the bowl, number of times lights near the front door and lights inside the apartment are switched on.

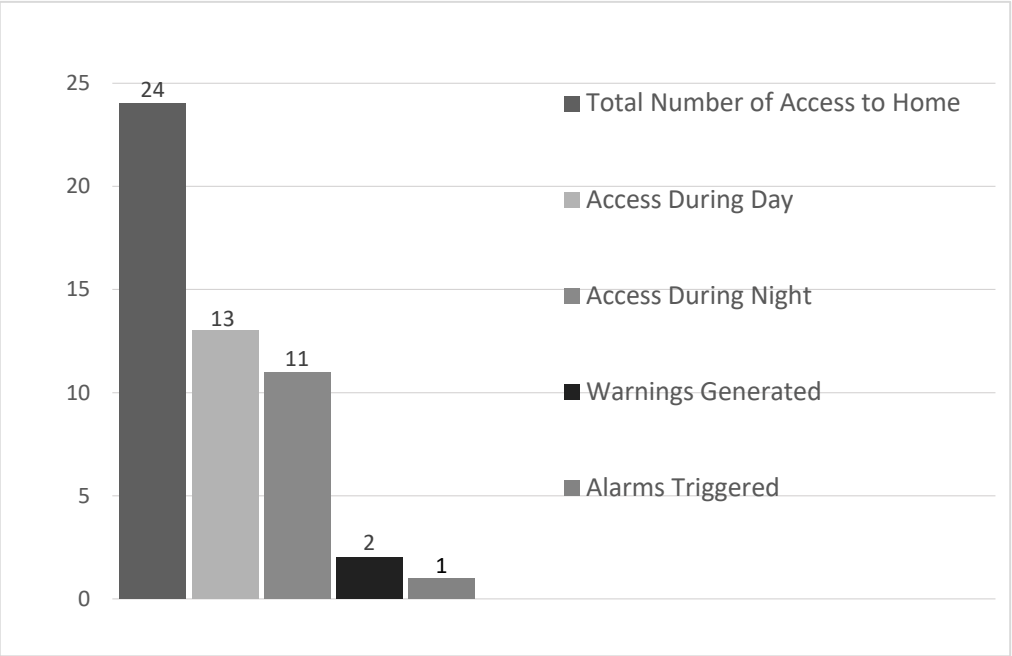


Figure 5.27. Represents total number of user entries to the home, number of day and night entries, and number of warnings generated along with number of alarms triggered.

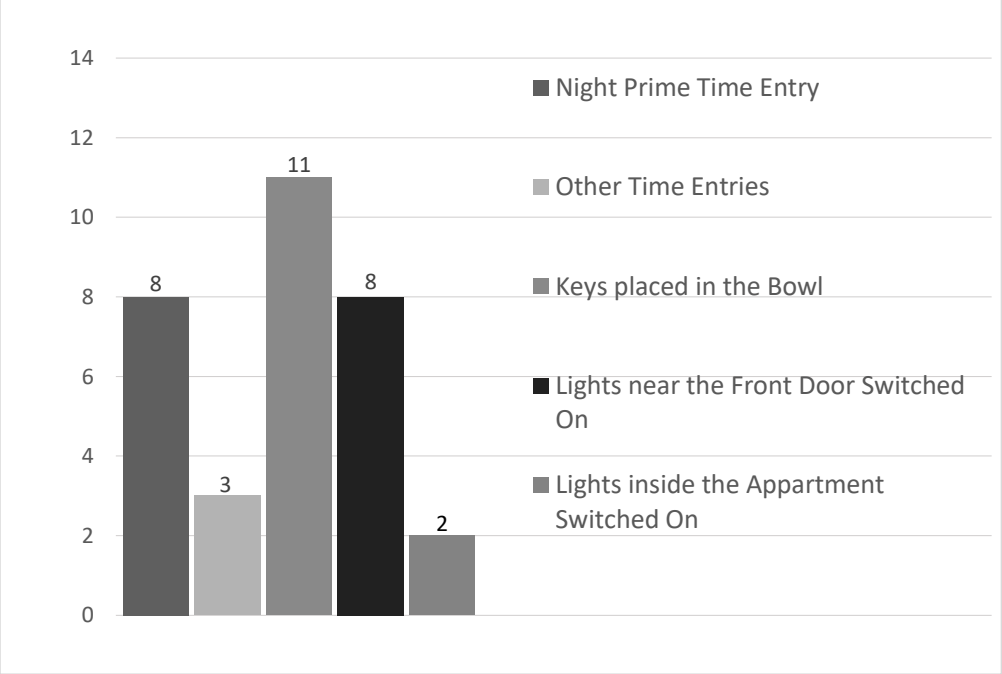


Figure 5.28. Shows the number of prime time entries during the night, number of other time entries to the home during night, number of times apartment keys are placed into the bowl, number of times lights near the front door and lights inside the apartment are switched on.

CHAPTER 6 DISCUSSION

6.1 CHAPTER OBJECTIVES

Chapter 6 discusses the result of the experiments which are stated in chapter 5. The chapter is divided into mainly four subsection. Section 6.1 discusses the chapter objectives while section 6.2 discusses and explains the results, features and comparison of the device fingerprinting experiment stated in chapter 5 subsection 5.2. Section 6.3 explain in detail the results and features of the logical sensing experiment mentioned in chapter 5 subsection 5.3 and how these results are achieved. Section 6.4 discusses the results, features and comparison of the behaviour prediction experiment illustrated in chapter 5 and subsection 5.4.

6.2 DEVICE FINGERPRINTING RESULT DISCUSSION, COMPARISON AND FEATURES

6.2.1 Device Fingerprinting Discussion

When a device accesses our test website www.fingerprintmydevice.com, the JavaScript embedded in the webpage collects the information about the device accessing the webpage. The “navigator.userAgent” accesses the User Agent string of the web browser and sends it back to the server, where it is analysed. Device fingerprint parameters such as browser name, browser version, OS name, OS version, OS Bits are isolated and identified by analysing the user agent string.

The JavaScript also checks the return values for “navigator.javascriptEnabled”, “navigator.flashEnabled”. When the “navigator.userAgent” returns the user agent string from the browser, it means JavaScript is enabled in the browser, so “navigator.javascriptEnabled” must return ‘true’, if it does not return ‘true’ it means the browser is intentionally giving misinformation. Irrespective of the return value from “navigator.javascriptEnabled” if the browser is returning device fingerprinting parameters it means JavaScript is enabled in the web-browser. Similarly, when flash device fingerprinting

parameters are available and “navigator.flashEnabled” return ‘false’ it means the browser is intentionally giving misinformation and it is ignored.

Unlike previous approaches to device fingerprinting, the algorithm checks if cookies are enabled in the browser by actually setting a cookie, accessing the set cookie and deleting the set cookie. Previous approaches to device fingerprinting utilized “navigator.cookieEnabled” to identify if cookies are enabled in the web browser. Similarly, whether local storage is enabled or disabled in web browser is identified by storing, retrieving and deleting values from the local storage.

Total number of installed plugins in the browser can be identified from “navigator.plugin.length”. The “navigator.plugin” object returns an array of plugin objects, with each plugin object corresponding to each unique plugin. Each object from the plugin array is analysed separately to gather more information in order to improve the accuracy of the developed fingerprint. “[EachPluginObject].name” gave the name of the plugin associated with that particular unique plugin object. By analysing the string returned from “[EachPluginObject].description” the algorithm was able to identify the version number of each of the plugin. “[EachPluginObject].length” gave the number of MIME type associated with each of the plugin.

When a plugin gets updated its version number changes. With an update the version number of the plugin always go up, which means, when a particular device visits the test website again while comparing the version number of the plugin, the extracted plugin version number is checked if it is greater than or equal to the existing plugin version number in the database. Moreover, the order in which plugins are installed in the browser corresponds to the date and time in which they are installed in the bowser, which means plugins installed at an earlier date will appear first in the retrieved plugin array compared to those plugins installed at a later date. So, while comparing the device fingerprints to identify a revisiting device the order of the plugins were also considered for comparison.

Total number of MIME types associated with the browser can be obtained from “navigator.mimeTypes.length”. “navigator.mimeTypes” returns an array of objects; each object is corresponds to each of the MIME type installed on the browser. Each MIME type is identified by “[EachMimeObject].type”. Suffixes associated with each of the MIME type is identified from “[EachMimeObject].suffixes”. Number of plugins associated with each MIME type is obtained from “[EachMimeObject].enabledPlugin.length”. Device fingerprinting parameters such as “navigator.language””, “navigator.product”, “navigator.appVersion”, “navigator.appName” were ignored after careful consideration as they were unreliable and gave inconsistent or false values across different browsers. Parameters such as “screen.updateInterval”, “screen.buffer” were eliminated from developing a device’s fingerprint because they are highly unreliable.

Various screen parameters are considered to develop a device’s fingerprint. The maximum available width and height of the screen is obtained from “navigator.screenmaxWidth” and “navigator.screenmaxHeight” respectively. The current width and height of the screen can be identified “navigator.screenavailWidth” and “navigator.screenavailHeight” respectively. Taskbar position and size on the screen is identified from screen’s maximum width, maximum height and current width and current height using simple arithmetic. Screen’s colour depth and pixel depth are identified from “navigator.screencolorDepth” and “navigator.pixelDepth” respectively.

The time zone and county in which the fingerprinted device is located can be identified from “navigator.date” object. Current system time is also obtained from “navigator.date” object. The latitude and longitude indicating the physical location of the device can be obtained from the “navigator.geoLocation.getCurrentPosition ()”. The gathered location information (latitude and longitude) is used as arguments for the Google Application Program Interface (Google API) to obtain the actual country name. The country name obtained from Google API is compared with the country name obtained from the “navigator.date” object. If the country names are a mismatch, then it indicates “navigator.date” object is intentionally giving misinformation. So, all the information from “navigator.date” object was ignored while developing the device’s fingerprint.

All the flash parameters are obtained from “flash.system.Capabilities” class. There are about forty parameters obtained using flash, these parameters are mentioned in table 3.3 under section 3.2.1. The number of fonts installed on the system is unique as well as the order in which these fonts are installed. So when comparing a device’s fingerprint system fonts as well as the order in which they are installed are compared.

When an device access the home the algorithm first checks its login credentials, if they match the fingerprint of the device is generated and compared with fingerprints in the ‘blacklist’ and if no match is found the fingerprint is then compared with those in the ‘whitelist’. All the device fingerprinting parameters are classified into nine as discussed in section 3.2.3. The significance of these nine parameters vary depending on the finger-printable information they have regarding the client device. Depending on the varying degree of similarity with the device fingerprints in the database each of these parameters are assigned scores. The computed device fingerprint score determines the level of similarity between fingerprints of the device accessing the web page and those in the database.

Our algorithm allows a 12^{o} degree of variation in both latitude and longitude, so it was able to successfully identify a machine’s fingerprint even when their geographical location was changed. The proposed algorithm is employed for device identification in smart homes, so a significant change in physical location or a change in time zone of a user device is unlikely. When a legitimate user is going abroad, it will result in a significant change in his geographical location and time zone, he has to make the necessary adjustments in his device identification system to still have access to his home via internet. Other modifications in the algorithm can be made specifically for legitimate users who frequently travels abroad.

Even though the version number of a particular plugin or the browser changed with a software upgrade, our algorithm was able to account for those changes and match the information with the corresponding fingerprint. The system fonts also changed over time due to the installation or uninstallation of a plugin or software but a dramatic change in the number or names of the fonts were not seen in the data.

The algorithm identified 95 out of the 97 devices visited the test web page. 2 of the devices the algorithm failed to identify were from the University of Pretoria Library even though they had both Flash and geo-location enabled. Their hard disks were cloned, so their OS, browsers, plugins, mime and all other browser specific and device specific information were identical along with their geographical location, as they are physically located in the same library.

6.2.2 Comparison of the Device Fingerprinting Algorithm

The 97.93% device identification accuracy when combined with legitimate login credentials provide substantial security to modern smart homes when they are accessed over the internet. Mayer's [52] study provided an accuracy of 96% while our proposed device fingerprinting algorithm provided an accuracy of 97.93%. Mayer utilized screen parameters, mime parameters, plugin parameters and other browser specific parameters for his identification. In our work, in addition to these parameters we considered flash and geolocation parameters for device identification and received a better device identification accuracy.

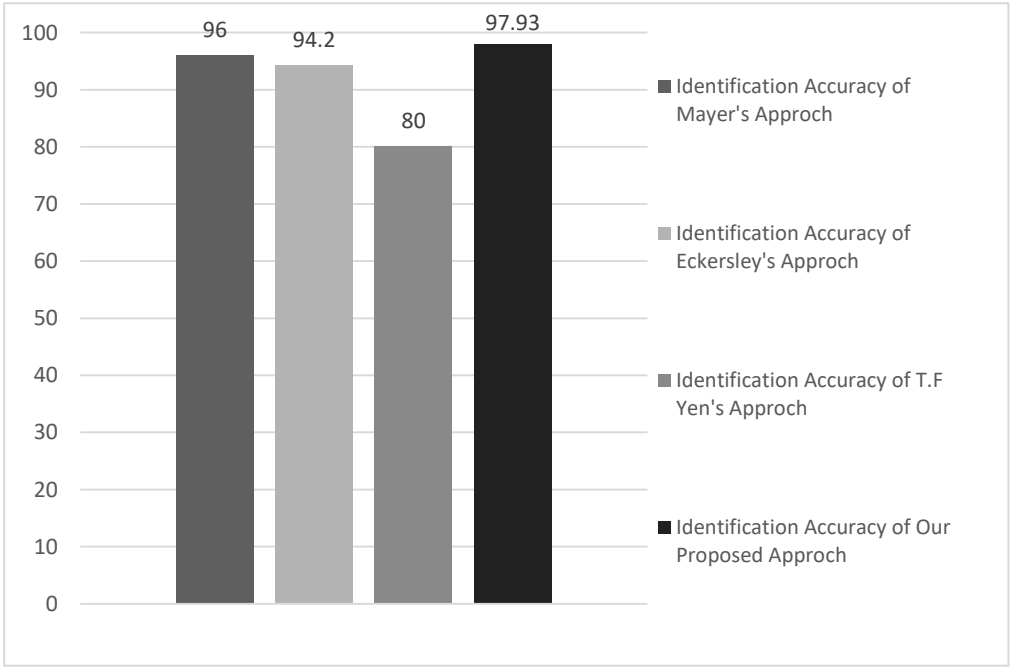


Figure 6.1. Shows the identification accuracy of the fingerprinting techniques proposed by Mayer [52], Eckersley [53], Yen et al. [54] and our proposed work.

Compared to our algorithm Eckersley's [53] study provided a lower device identification accuracy of 94.2%. He utilized java script and Flash for device identification in his study. The proposed work in this thesis uses geolocation fingerprinting in addition to flash and java script.

The approach of Yen et al. [54] provided a device identification accuracy of 80% at best when they combine user agent parameter with the IP prefix information, which is less than the identification capability of our proposed work. The author only utilized user agent parameters along with IP prefix information to develop their fingerprint, while our proposed work used 68 device fingerprinting parameters (including user agent) to develop our device fingerprint. The thesis did not consider any IP prefix information as they are not reliable and can be easily spoofed. Figure 6.1 compares the identification accuracy of the fingerprinting techniques proposed by Mayer [52], Eckersley [53], Yen et al. [54] and our proposed work.

The work of P. Eckersley [53] had 18.1 bits of entropy while our proposed work has a better entropy of 22.57 bits. The work of Yen et al. [54] had an entropy of 20.29 bits at best when IP addresses were combined with UA information which is lower in comparison with our obtained entropy of 22.57 bits. Figure 6.2 compares the entropy values of P. Eckersley [51] and Yen et al. [54] with our work.

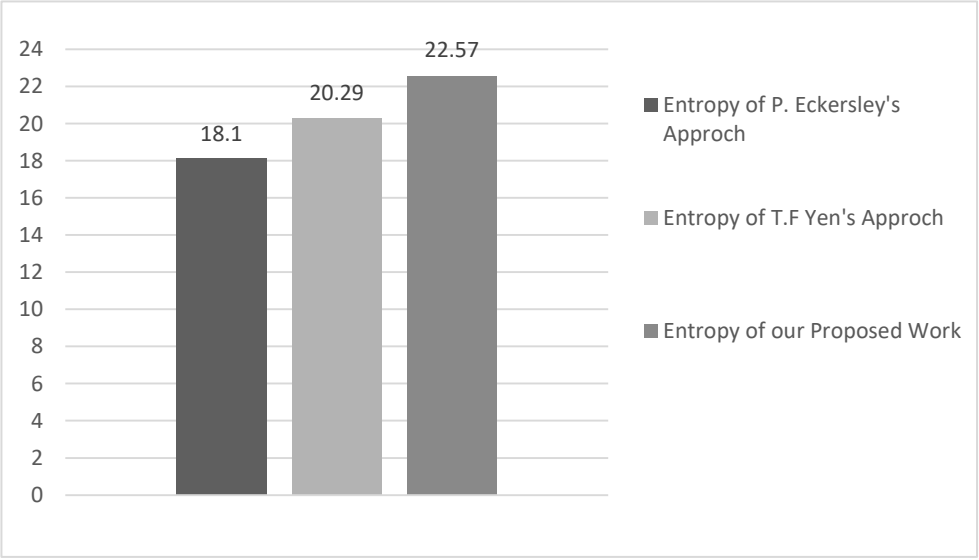


Figure 6.2. Shows the entropy values of P. Eckersley [51] and Yen et al. [54] and our work.

Moreover, none of the previous device fingerprinting approaches include any form of device fingerprinting parameter validation to defend against parameter spoofing. In the proposed work, user agent, screen parameters, date object and flash parameters are validated, which increases the legitimacy of the generated fingerprint.

6.2.3 Features of the proposed Device Fingerprinting Algorithm

The device fingerprinting technique discussed in this thesis was designed for home automation systems with security as the primary objective. Our work attempts to identify the person operating the device as well as the device used to access the home; this is achieved through a two stage verification process. Comparing and verifying different parameters like OS name in UA string, and screen maximum width and height in Screen parameter, with those from flash, helps us to establish the legitimacy of UA and Screen parameters. Moreover, getting the country name from Google API by utilizing the latitude and longitude obtained from Geo-Location and comparing the country name with the country name in the date object helps us to determine the validity of the date object and time zone. These validations safeguard against parameter spoofing and enhances security.

Hash function can be used to encrypt the device fingerprinting parameters (with a few exception where version number has to be checked) to protect against eavesdropping attack or man in the middle attack attempts. The authenticity of the JavaScript used for fingerprinting can be verified by java script self-evaluation using Message-Digest algorithm 5 (MD5) checksum.

The proposed system gives clients accessing the home a reason to enable Flash, JavaScript and Geo-Location and encourage device fingerprinting, as it improves the security of their home over the internet. Using blacklist and whitelist the proposed home security system could identify and distinguish between legitimate and attacker device fingerprints, which significantly reduce the need to contact the user every time a fingerprint mismatch occurs, this improves user convenience.

6.3 LOGICAL SENSING RESULT DISCUSSION, COMPARISON AND FEATURES

6.3.1 Logical Sensing Discussion

States 1 to 16 occurred when the home was occupied. The most common state triggered was state 4, which happened when the home was occupied and user opened a closed primary access point from the inside triggering the motion and proximity sensors and stepped out of the home and closed the door behind him. State 4 is usually triggered when the user leaves the home. After the door was closed the algorithm waited for 15 seconds for any intermediate state changes and since door remained closed, it changed the state of the home to empty. State 1 is triggered when the user opened the home from the inside triggering the motion and proximity sensors and came back into the home leaving the door open again triggering the sensors. State 7 happened when the user closed the open door from the inside and came back into the home, motion and proximity sensors are triggered before the initial state and after the final state.

State 8 occurred when the user walked from inside the home and closed an open door and stepped out, the motion and proximity sensors are triggered before the door is closed when the user walked towards the door; sensors are not triggered after the door is closed. After state 8 the algorithm waited until the door observation timer expired and changed the home state to empty. State 6 is triggered when the occupant came in from inside the home and opens a closed door and before the door observation timer expires closed it again and goes back into the home, making the open state of the door an intermediate state. Motion and proximity sensors are triggered before the door was opened and after it was closed. State 12 occurred when an open door is closed and opened by a user coming from inside the home and after opening the user went back inside. The motion and proximity sensors are triggered before the door is closed and after the door was opened.

State 11 happened when the user came from inside the home triggering the motion and proximity sensors and closed an open door and then opened it again before the door observation timer expired and stepped outside the home without triggering the sensors after

the final state of the door. After state 11 the door is left open and the home is vacant, so the home state change timer is started which upon close to expiry warned the user about the impending home state change and when expired changed the home state to empty. State 13 occurred when the user comes in from the home and opens a closed door triggering the motion and proximity sensors and leaves the door open and exits the home. Similar to state 11, after state 13 is triggered the door is left open and the home is empty so the home state change timer is started which warns the user and changes home state to empty when expired. During the experiment State 11 and state 13 were triggered a combined 23 times. Whenever states 11 or 13 is triggered state change timer is started, so home state change timer was started 23 times.

State 9 occurred when the open door is closed from the outside and user stays outside, so motion and proximity sensors are not triggered before or after closing the door. When state 9 is triggered the running state change timer is not reset but it continues as the user did not re-enter the home, so when it expires home state is changed to empty. State 10 is triggered when an open door is closed and opened from the outside without triggering any motion or proximity sensors before closing the door or after opening the door. Similar to state 9, after triggering state 10 state change timer is not reset but it continues and upon expiry home state is changed to empty. Both states 9 and 10 are only triggered when previous states are either 3, 10, 11 or 13. State 3 happened when a closed door is opened from the outside without triggering the motion or proximity sensors before the initial state or after the final state. State 3 happened when the state change timer was running and the previous state was either 5 or 9. Since the user stayed outside the home in state 3, the state changer timer was not reset and allowed to continue.

State 14 happened when an open door is closed and the user came back into the home from the outside. Motion and proximity sensors were not triggered before the door is closed but they were triggered as the user walked into the home after closing the door. After triggering state 14 as the user re-entered the home state change timer was reset without changing the state of the home to empty. State 2 was triggered when the user opened a closed door and came back into the home. State 2 happened when the state change timer was still running

and user re-enters the home (previous states are either 5 or 9), so the state change timer was reset without changing the home state.

State 15 was triggered when a closed door was opened and closed while the state change timer was still running and previous states were either 5 or 9. Motion and proximity sensors were not triggered before opening the door but as the user re-enters the home the sensors were triggered after the door was closed. Like in states 14 and 2 state change timer was reset without changing the home state. State 16 occurred when the open door was closed and then opened with the home state change timer still running. As user entered the home from the outside, motion and proximity sensors were not triggered before closing the door but they were triggered when the user came in after opening the door. Like in states 14, 2 and 15 the state change timer was reset without changing the home state. Whenever states 14, 2, 15 or 16 were triggered state changer timer was reset without changing the home state to empty. During the experiment, states 14, 2, 15 and 16 occurred a total of 15 times. So state change timer was reset 15 times. Home state changed from occupied to empty due to expiration of the state change timer 8 times, this accounted for 8 of the 14 warnings generated by the algorithm. When user re-entered the home these 8 times IVM was activated and he was asked to confirm his identity. Twice the user forgot to confirm his identity upon re-entry, which accounted for 2 of the 5 intrusion alarms raised.

States 17 to 32 are triggered when the home is empty and the main access point state is changed. State 17 was triggered when the home is empty and someone opens the main door from the outside and enters the home after closing the door behind them triggering the motion and proximity sensors. When state 17 occurred someone entered the empty home, so IVM was triggered. If user verification is not done within identity verification time period (90 seconds) or if verification fails alarm is triggered. If the algorithm confirms user identity, then the state of the home is changed from ‘empty’ to ‘occupied’.

State 19 occurred when a closed door was opened from the outside and the user left the door open and came into the home triggering the motion and proximity sensors. User identity has to be verified as the user entered an empty home, so IVM was activated. States 17 and 19

were triggered 46 times, which accounted for 46 of the 57 IVM activations. Out of all the 46 IVM activations after states 17 and 19, the user identity was successfully verified 46 times without triggering any alarm. State 22 happened when the closed main door was opened and closed from the outside without the user entering the home. Motion and proximity sensors were not triggered before opening the door or after closing the door. IVM was not activated because no one entered the home, door remains closed so algorithm waited for another state change of the door. After triggering state 22, states 17, 19, 20, 22 could be triggered.

State 26 is triggered when the open main door is closed from outside without triggering the motion and proximity sensors before or after closing the door. After state 26 was triggered, no one entered the home, so no user identification was necessary. State 27 was triggered when the open main access point was closed then opened from the outside without triggering the motion and proximity sensors before closing the door or after opening the door. After state 27 was triggered the primary access point remained open, so the algorithm kept observing the motion and proximity sensors until another state change was triggered to determine user entry into home. States 26 and 27 were only triggered when the previous states were either 11, 13, 27 or 20.

State 20 was triggered when a closed door was opened from the outside and left open. Motion and proximity sensors were not triggered before or after opening the door, so nobody entered the home when state 20 was triggered hence IVM was not activated. Like in case 27, after triggering state 20 the algorithm kept observing the motion and proximity sensors until another state change was triggered to determine user entry into home. State 20 was triggered thrice during the experiment; out of the 3 times once user re-entered the home without changing the door state triggering special case 34 in Table II. The algorithm identified this action by motion and proximity sensor triggers, so IVM was immediately activated and user identity was confirmed. Twice state 31 was triggered after state 20 when user closed the door and entered the empty home.

State 31 was triggered when the user closed an open door and entered an empty home triggering the motion and proximity sensors after closing the door. Identity of the user was

verified as the user entered an empty home. State 32 occurred when an open main door was closed and opened by the user as he entered the home. Motion and proximity sensors were triggered after the door was opened as the user walked into the empty home, so user identity was verified. State 31 and 32 occurred 7 times which accounted for seven of the IVM activations.

Each of the states 21 and 18 were triggered once. When these states were triggered motion and proximity sensors were triggered before the door was opened. This will not happen in an empty home. So intrusion alarm was activated without waiting for user identity confirmation. This accounted for 2 of the 5 intrusion alarms generated. Thrice balcony door was opened when the user was in bed during day time, this happened due to the presence of a second person in the apartment; 3 of the IVM and warnings were triggered as a result of this. User confirmed his identity before user verification timer was expired. Once after state 13 and after expiration of the state change timer (home state changed to empty) when user re-entered the home by triggering state 31 he forgot to confirm his identity so the intrusion alarm was activated after identity verification timer was expired. Similarly, once after state 13 and expiration of the state change timer user re-entered the home triggering state 27 and forgot to confirm his identity so intrusion alarm was triggered. Once the balcony window was opened from the outside without triggering the motion or proximity sensors deployed inside the home near the window during night when the user was in bed. A window will never be opened from the outside when the user is in bed, so without waiting for identity confirmation the intrusion alarm was triggered.

The measured voltage across the MQ 9 sensor went up when there was fire, the Rs to Ro ratio varied from 10.86 when there was no fire to 8.22 when there was fire. 12 second average temperature showed a spike of 2.5oC and average humidity showed a 3% decline within 12 seconds when there was fire. When the bed was unoccupied the force sensors gave a reading because they are triggered by the weight of the bed mattress, as they are placed underneath the mattress. When objects like stack of books and heavy boxes are placed on various parts of the bed the sensor values increased due to their weight but value of bed top and bottom sensor did not increase simultaneously as the placed objects did not have the weight

distribution to trigger both sensors. When objects are placed close to the top of the bed, force sensor values at the shoulder region increases but force value at the abdominal region of the bed remained constant with minimal variation. Similarly, when objects are placed close to the bottom of the bed force sensor values at the abdominal region varies while force detected at the shoulder region of the bed remained constant.

6.3.2 Features and Comparison of the proposed Logical Sensing Algorithm

The proposed work observes primary and secondary access points to identify logical sensing parameters and detect intrusion and does not cause inconvenience to the user with wearable tags or laser grids. It offers implementation ease and flexibility compared to the security system proposed by B. Schilit et al. [13]. The system requires minimal user input to identify when the home becomes empty or occupied, it was able to observe various access points in the home and deduce the change of state of the home. The algorithm was able to successfully predict home state changes and activate identity verification mechanisms when necessary.

In their research, B. Fouladi [7] gained access and manipulate the system by eavesdropping on the ZigBee communications in the home network and was able to capture the encryption key in plain text. All the ZigBee wireless communication used in the work is encrypted using 128 bit AES encryption and the encryption key is never exchanged in clear text over the air, so an eavesdropping attacker will not be able to gain access to the system and manipulate it. O. Yurur et al. [80] utilized context aware computing to improve home security, user context is identified by saving, analyzing and sharing data regarding user behaviour and context which raised security and privacy concerns. In the proposed work, access point data is stored in a data base on Raspberry Pi which is kept inside the home and secured using physical locks with access limited to authorized personals. Moreover, the stored data is never shared and it can be encrypted to further improve security.

Morsalin et al. [87] utilized NFC tags to predict user location in a home and the user had to verify his fingerprint and enter the password each time when the user wants to access their home. The security algorithm proposed in the thesis, did not utilize NFC communication tags which reduces the complexity of the system and improves user convenience. Moreover,

the IVM is only triggered when someone enters an empty home. The algorithm was also able to identify secondary access point actions initiated by the user and was able to distinguish them from intruder actions.

6.4 BEHAVIOUR PREDICTION RESULT DISCUSSION, COMPARISON AND FEATURES

6.4.1 Behaviour Prediction Discussion

During machine learning the 24 hours of the day was divided into two based on the intensity of light in the apartment. The user cannot be expected to switch on the lights in the apartment when there is enough natural light in the apartment. Even if he does switch on the lights a few times during day when he enters the home the user has no reason to switch the lights on during day when there is enough light in the apartment. So the algorithm cannot logically consider user's light behaviour during day. Moreover, dividing the day into two allows two prime time identifications, day prime time and night prime time which adds to the algorithm's user behaviour prediction accuracy.

In order to understand intruder behaviour a friend is asked to do the intrusions to the home by considering and maintaining all the parameters an intruder might use. He identifies when the home will more likely be empty and how many access points he could use to enter the home. The intruder is provided with the keys to the home so that he should not have to pick the lock in public areas and risk other inhabitants of the flat complex seeing him and alerting the authority. We assume a real intruder would pick the lock and gain access to the home. The intruder tried his luck trying to break in when the home was occupied during the day six times. All those attempts failed and four times alarm was triggered since the user was in bed or the door was opened from the outside and twice the user was warned about the intrusion attempt.

During the initial seven weeks of the experiment, user's behaviour inside a home after he enters an empty home is observed and a Bayesian network was constructed based on the knowledge. There was a significant disparity between user behaviour during the day and

night, so they were considered and analysed separately. During day since there was ample light in the apartment, user rarely switched the lights on when he entered an empty home. So while considering user's behaviour during the day light behaviour was not considered. On the contrary, during night user switched on at least one of the lights on within 12 seconds when he entered the empty home. So his light behaviour is considered for behaviour prediction during the night.

Irrespective of day or night the time at which a user enters the home and his key placement behaviour are considered for behaviour prediction. Upon analysis it is observed that, most often a user knows where the light switches in a home are and where to keep the keys. So he can turn the lights on quickly always keeps the apartment keys in the same place. Further observation implied that, more often user comes home during the day between 2:30 pm and 4:30 pm while during night between 10:00 pm and 12:00 pm. Most often user placed the keys inside the key-bowl within 15 seconds of entering an empty home during day or night.

A Bayesian belief network was constructed based on the learned user behaviour during day and night. When someone enters the home between 06:30 am and 5:30 pm his behaviour is compared against user's learned day behaviour and his behaviour between 5:31pm and 6:29 am is compared against user's learned night behaviour. Whenever someone enters an empty home his behaviour is analysed by the Bayesian network to identify and distinguish between normal behaviour from suspicious behaviour.

The Bayesian networks given in chapter 5 represented in Figures 5.10 to 5.26 shows the probabilities of good and bad user behaviour during day and night calculated using equations (5.2) and (5.3) respectively. The user behaviour calculated using the Bayesian networks are tailor made for an individual user which the algorithm learns during machine learning phase. Table 5.1 in chapter 5 shows the possible combination of user behaviour during day. Entry 1 in table 5.1 and Figure 5.10 (a) shows the probability of user's good and bad user behaviour when someone enters the home during prime time and keys are placed inside the bowl. Using equation 5.3, the good user behaviour probability was observed at 99% which is very high.

Entry 2 in table 5.1 and Figure 5.10 (b) shows good user behaviour probability when someone enters the empty home during prime time and when keys are not placed in the bowl. Based on prior knowledge of the Bayesian network in equation 5.3 probability of good user behaviour was observed at 5.12% which is very low. Entry 3 in table 5.1 and Figure 5.10 (c) shows good and bad user probabilities when someone enters an empty home during other time of the day and apartment keys are placed inside the bowl. The Bayesian network stated by the equation 5.3 calculates good user behaviour probability at 87.5% based on prior user behaviour. Similarly, entry number 4 in table 5.1 and Figure 5.10 (d) shows good user behaviour probability when someone enters an empty home during other time of the day and keys are not placed inside the bowl. Using the Bayesian belief network in equation 5.3 the good user behaviour probability was calculated at a low 12.5%. When cases 2 and 4 from table 5.1 is triggered user behaviour is identified as suspicious and Identity Verification Time (IVM) is reduced as defensive mechanism.

Table 5.2 and Bayesian network equation 5.2 shows various possible user behaviours during the night. Entry number 1 in table 5.2 and Figure 5.11 shows the probability of good user behaviour when someone enters an empty home during night at prime time. Lights near the front door and lights inside the apartment are switched on within 6 and 12 seconds respectively and keys are placed inside the bowl within 15 seconds. The Bayesian network defined by equation 5.2 calculates good user probability at 61.1%.

Entry number 2 in table 5.2 and Figure 5.12 shows the probability of good user behaviour when someone enters an empty home during the night at prime time. Lights near the front door are not switched on within 6 seconds and lights inside the apartment are switched on within 12 seconds and keys are placed inside the bowl within 15 seconds. The probability of good user behaviour was observed at 77.8% using the Bayesian network constructed using equation 5.2 based on prior knowledge. Unlike entry 2, entry 3 in table 5.2 shows user behaviour when lights near the front door are switched on within 6 seconds and lights inside the apartment was not switched on within 12 seconds. Figure 5.13 shows the probability of good user behaviour based on prior knowledge utilized in the Bayesian network; the good user behaviour probability was observed at 83.3% which is quiet high.

Entry number 4 in table 5.2 and Figure 5.14 shows the probability of good user behaviour when someone enters an empty home during the night at prime time. Lights near the front door and lights inside the apartment are not switched on within 6 and 12 seconds respectively and keys are placed inside the bowl within 15 seconds. The Bayesian network defined by equation 5.2 shows the probability of good user behaviour at 5.55% which is quiet low. So identity verification time is reduced.

Entry 5 in Table 5.2 and Figure 5.15 shows the structure of the Bayesian network during night in Netica when user access the home during other times and keys are placed inside the bowl within 15 seconds, lights near the front door and lights inside the home are switched on within 6 and 12 seconds respectively. The good user behaviour probability is 42.8% while the bad user behaviour is 57.2%, this is because the Bayesian network is customized to a particular user during machine learning. During machine learning out of the 14 times user entered the home during other time of the night and keys were placed inside the bowl both lights were switched on only 6 times. On the other hand, the good user behaviour was measured at 85.7% by the Bayesian network when user entered the home during other time of the night and only lights near the front door were switched on, because during machine learning 12 of the 14 times when user entered the home during other times of the night and keys were placed inside the bowl, within 12 seconds of entry, user only turned the lights near the front door on within the specified time.

Entry number 6 in table 5.2 and Figure 5.16 shows the probability of good user behaviour when someone enters an empty home during the night during other time of the night and keys are placed inside the key-bowl within 15 seconds of opening the door to the empty home. Lights near the front door are not switched on within 6 seconds and lights inside the apartment are switched on within 12 seconds. The Bayesian belief network defined by equation 5.2 gives good behaviour probability of 64.3%. Unlike entry 6, for entry 7 in table 5.2 lights near the front door are switched on within 6 seconds of entry to the home and lights inside the apartment was not switched on within 12 seconds of entry to the home. Figure

5.17 shows the probability of good user behaviour based on prior knowledge utilized in the Bayesian network; the good user behaviour probability was observed at 85.7% which is high.

Entry number 8 in table 5.2 and Figure 5.18 shows the probability of good user behaviour when someone enters an empty home during other time of the night and keys are placed inside the key-bowl within 15 seconds of opening the door to the empty home. Lights near the front door and lights inside the apartment are not switched on within 6 and 12 seconds respectively. The Bayesian network defined by equation 5.2 shows the probability of good user behaviour at 7.14% which is quiet low. So identity verification time was reduced as a defensive mechanism.

During machine learning, when the user enters the home during the night and keys are not placed inside the bowl within 15 seconds of opening the door, the user behaviour is generally considered as bad, this is because the Bayesian network is customized to the particular user during machine learning. During machine learning, out of the 32 times user entered the home during the night 27 times keys are placed inside the key-bowl within 15 seconds of opening the front door. So good user behaviour probability is quiet low for entries from 9 to 16 in table 5.2.

Entry 9 in table 5.2 and Figure 5.19 shows the probability of good user behaviour when someone enters an empty home during the night at prime time and keys are not placed inside the key-bowl within 15 seconds of opening the door. Lights near the front door and lights inside the apartment are switched on within 6 and 12 seconds respectively. The Bayesian network defined by equation 5.2 observes good user probability at 9.09% which is quiet low and identity verification time was reduced.

Entry 10 in table 5.2 and Figure 5.20 shows the probability of good user behaviour when someone enters an empty home during the night at prime time and keys are not placed inside the key-bowl within 15 seconds of opening the door. Lights near the front door are not switched on within 6 seconds and lights inside the apartment are switched on within 12 seconds. The Bayesian network defined by equation 5.2 observes good user probability at

7.14% which is quiet low, so identity verification time was reduced. Entry number 11 in table 5.2 is similar to entry 10 except it shows the probability of good user behaviour when lights near the front door are switched on within 6 seconds and lights inside the apartment was not switched on within 12 seconds. Figure 5.21 shows the Netica representation of the Bayesian network given using equation 5.2 for entry 11 in table 5.2. The good user probability was observed at 6.66% which is quiet low, so identity verification time was reduced as a defensive mechanism.

Entry 12 in table 5.2 and Figure 5.22 shows the probability of good user behaviour when someone enters an empty home during the night at prime time and keys are not placed inside the key-bowl within 15 seconds of opening the door. Lights near the front door and lights inside the apartment are not switched on within 6 and 12 seconds respectively. The Bayesian belief network defined by equation 5.2 observes good user probability at 1.0% which is quiet low, so identity verification time was reduced as a defensive mechanism.

Entry 13 in table 5.2 and Figure 5.23 shows the probability of good user behaviour when someone enters an empty home during other time of the night and keys are not placed inside the key-bowl within 15 seconds of opening the door. Lights near the front door and lights inside the apartment are switched on within 6 and 12 seconds respectively. The Bayesian network defined by equation 5.2 observes good user probability at 16.7% which is quiet low and identity verification time was reduced.

Entry number 14 in table 5.2 and Figure 5.24 shows the probability of good user behaviour when someone enters an empty home during other time of the night and keys are not placed inside the key-bowl within 15 seconds of opening the door. Lights near the front door are not switched on within 6 seconds of opening the door and lights inside the apartment are switched on within 12 seconds of opening the door to an empty home. The Bayesian network constructed based on the behaviour learned during machine learning defined by equation 5.2 observes good user probability at 11.1% which is quiet low, so identity verification time was reduced. Entry number 15 in table 5.2 is similar to entry 14, except lights near the front door are switched on within 6 seconds of opening the door and lights inside the apartments are

not switched on within 12 seconds of opening the door. Figure 5.25 shows the Netica representation of the Bayesian network defined by equation 5.2 observes good user probability at 1.0% which is quiet low, so identity verification time was reduced.

Entry 16 in table 5.2 and Figure 5.26 shows the probability of good user behaviour when someone enters an empty home during other time of the night and keys are not placed inside the key-bowl within 15 seconds of opening the door. Lights near the front door and lights inside the apartment are not switched on within 6 and 12 seconds respectively. The Bayesian belief network defined by equation 5.2 observes good user probability at 1.0% based on prior user behaviour learned during machine learning.

Whenever good user behaviour probability is less than 60%, the identity verification time is reduced from 90 seconds to 45 seconds as a defensive mechanism. The algorithm concludes that, if good user behaviour probability is less than 60% the present user behaviour compared to the learned user behaviour shows significant disparity and the algorithm has serious doubts about the identity of the person entering the home, so identity verification time was reduced. If good user behaviour is more than 60%, the algorithm concludes that based on the present user behaviour there is a high probability that the person entering the empty home is a legitimate user. His identity is reconfirmed using IVM, but the time allowed for the user to confirm his identity is not altered.

During night twice the intruder was able to switch on the lights near the front door. This behaviour can be accredited to the intruder familiarizing with the home after several repeated break-ins. If this happens after the behaviour prediction algorithm was implemented, the algorithm still would be able to warn the user about the intrusion attempts, as there was no key placed in the bowl. The invader did not place the keys in the key-bowl during any of the intrusion attempts as he was unaware of the parameters considered for the behaviour prediction algorithm. Having failed in his day time break-in attempts the intruder tried to access the home during the night through primary and secondary access points when the home was occupied and when the user was in bed. All those times resulted in the algorithm raising alarms.

After the algorithm was implemented, once during the day the user forgot to put his keys in the bowl within 15 seconds of opening the door which immediately resulted in a warning, informing the user about the anomalous user behaviour. The user headed the warning and immediately confirmed his identity. During night the user forgot to switch on any lights when he entered the home within 12 seconds which also resulted in warning which the user did not respond to promptly and an intrusion alarm was triggered after 45 seconds.

6.4.2 Features and Comparison

The proposed work takes into account the unique user behaviour after entering a home and identifies the identity of the user based on that. The other identity verification mechanism used in the algorithm from logical sensing reconfirms the user identity and enhances security. When an anomaly in user behaviour is detected the time given to confirm a user identity using identity verification mechanism is significantly reduced before activating intrusion defence mechanisms, thus giving little time to an intruder inside the home. The proposed security system can be always active even when the home is occupied, the user does not have to activate it when he leaves the home thus partially eliminating human negligence from the equation. The logical sensing algorithm can identify when a home becomes empty. Unlike any previous approaches based on the learned user behaviour in the experiment, the algorithm identified suspicious behaviour within 15 seconds of someone entering an empty home.

Unlike the approach proposed by the researchers [108] – [114], the behaviour prediction algorithm discussed in this thesis predicts user behaviour after entering the home from a security perspective without utilizing any wearable sensors. Elimination of wearable sensors improves user convenience and eliminates the chance of a wearable device being stolen by an intruder and misusing it to gain access to the home. The user activity recognition method proposed by P. N. Dawadi et al. [115] and Paavilainen et al. [116] takes considerable time to identify anomalies in various actions, on the contrary the proposed algorithm takes a maximum of fifteen seconds to identify unexpected user behaviour.

The work of Chahuara et al. [117] utilizes microphones deployed throughout the home to identify user position inside the home and considered parameters like user's speech, water consumption etc. which raises some serious privacy issues. The proposed work did not utilizes cameras or micro phones which significantly improves user privacy. User's behaviour data stored in the Raspberry Pi is never shared and it can be encrypted to further improve data security. Moreover, Pi is kept within the home which is secured using physical locks.

CHAPTER 7 CONCLUSION

Introduction of the thesis focuses on the general context of the concept of smart home and home automation. Literature review section focuses on the security aspect of the existing Home Automation Systems and points out its flaws. It shows how the concept of security and meaning of the word ‘intruder’ has changed in modern homes. The section goes on to points out the shortcomings of the existing Home Automation Systems in identifying and preventing sophisticated intruders in a home environment. The literature review section also explains various literatures relating to device fingerprinting, logical sensing and behavioural prediction.

The device fingerprint along with username/password based security proposed in this thesis, enables the verification of user as well as the device used to access the home, which significantly improves home security when they are accessed over the internet. In this work, the device fingerprinting algorithm was able to uniquely identify 97.93% of devices accessing our test website with an entropy of over 22 bits. Unlike any previous approaches to device fingerprinting, we use geolocation data in our algorithm which improves the fingerprint accuracy. The user agent verification, screen parameter verification and client’s date object verification proposed in our work drastically improves the legitimacy of the fingerprints generated.

Logical sensing detects user actions at primary and secondary access points in a home using different sensors. These detected user actions and behaviours are compared with normal user behaviour at various access points to identify intrusions or intrusion attempts. In the experiment, our proposed algorithm was able to successfully identify all 305 state changes of the main access point and reduce them to 190 user behaviours while the secondary access point changed state 56 times. The alarm was triggered five times when the user failed to confirm his identity. Six of the fourteen warnings generated were regarding secondary access points while the other eight were relating to primary access point when the home became empty. In addition to identifying intrusions in home, the algorithm also warns user about imminent and live potential security vulnerabilities by identifying the status of various access points, user position and behaviours.

Behaviour prediction identifies parameters by which legitimate user behaviour can be identified in a timely fashion to successfully utilize it to improve home security. The proposed algorithm takes into account the time at which the user usually comes home, user's light behaviour and his key placements to predict legitimate user behaviour. These parameters were deduced and normal user behaviour was identified from 7 weeks of training data. The proposed security algorithm was implemented in the apartment for a period of 2 weeks in which the user came home 24 times during day and night generating 2 warnings and 1 alarm.

For future works in device fingerprinting, identification accuracy of our fingerprints can be improved when the website is visited using different web clients from the same machine; Adding more fingerprinting parameters such as clock skew and other hardware specific parameters for improving the accuracy of the device fingerprint is also an idea worth pursuing. Moreover, depending on unique user behaviour more behaviour and logical sensing parameters can be identified and developed.

REFERENCES

[1] C. Karlof, D. Wagner, “Secure routing in wireless sensor networks: attacks and countermeasures”, *Ad Hoc Networks*, vol. 1, pp. 293–315, 2003.

[2] Zhong, S. Ji, S. Chen, T. (2014). Wormhole Attack Detection Algorithms in Wireless Network Coding Systems, *IEEE Transactions on Mobile Computing*, Vol. 1(1). pp. 62 - 76.

[3] Raza S., Duquennoy S., Höglund J., Roedig U. and Voigt T. (2014), Secure communication for the Internet of Things—a comparison of link-layer security and IPsec for 6LoWPAN, *Security Comm. Networks*, 7, pages 2654–2668.

[4] X. Wang and H. Qian, "Constructing a 6LoWPAN Wireless Sensor Network Based on a Cluster Tree," in *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 1398-1405, March 2012.

[5] Shahid Raza, Linus Wallgren, Thiemo Voigt, SVELTE: Real-time intrusion detection in the Internet of Things, *Ad Hoc Networks*, Volume 11, Issue 8, 2013, Pages 2661-2674.

[6] J. Wright, “Practical ZigBee Exploitation Framework”, *Toorcon*, Oct. 2011.

[7] B. Fouladi, S. Ghanoun, “Security Evaluation of the Z-Wave Wireless Protocol,” *Black hat USA*, Aug. 2013.

[8] T. Oluwafemi, S. Gupta, S. Patel, T. Kohno, "Experimental Security Analyses of Non-Networked Compact Fluorescent Lamps: A Case Study of Home Automation Security”, *Workshop on Learning from Authoritative Security Experiment Results, (LASER 2013)*, Arlington, Virginia, USA, October 16th - October 17th, 2013.

[9] A. Verrotti, D. Trotta, C. Salladini, G. Corcia, G. Latini, R. Cutarella, F. Chiarelli, “Photosensitivity and epilepsy: a follow-up study,” *Developmental Medicine & Child Neurology*, vol. 46, issue 5, pp. 347-351, May 2004.

[10] M.R Faghani, U.T Nguyen, "A Study of XSS Worm Propagation and Detection Mechanisms in Online Social Networks", *IEEE Transactions on Information Forensics and Security*, Vol. 8 (11). pp. 1815 - 1826, 2013.

[11] Tamara Denning, Tadayoshi Kohno, Henry M. Levy, "Computer security and the modern home", *Communications of the ACM*, Volume 56 Issue 1, January 2013, Pages 94-103.

[12] Anind K. Dey, "Understanding and Using Context," *Personal and Ubiquitous Computing*, vol. 5, issue. 1, pp. 4-7, February 2001.

[13] B. Schilit, N. Adams, R. Want, "Context-Aware Computing Applications," in *WMCSA '94 Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, pp. 85-90, 1994.

[14] Victoria Bellotti, Keith Edwards, "Intelligibility and Accountability: Human Considerations in Context Aware Systems," *Human-Computer Interaction*, vol. 16, issue 2, pp. 193-212, December 2001.

[15] Stephen S. Intille, "Designing a Home of the Future", *IEEE Pervasive Computing*, Vol. 1, issue 2, pp. 7682, April 2002.

[16] Sin-Min Tsai, Po-Ching Yang , Shyi-Shiou Wu , Shya-Shiow Sun, "A Service of Home Security System on Intelligent Network," *IEEE Transactions on Consumer Electronics*, vol. 44, issue. 4, pp. 1360-1366, November 1998.

[17] K. Atukorala, D. Wijekoon, M. Tharugasini, I. Perera, C. Silva, "SmartEye - Integrated solution to home automation, security and monitoring through mobile phones," *Third International Conference on Next Generation Mobile Applications, Services and Technologies, NGMAST '09*, pp. 64-69, September 2009.

[18] N. Sriskanthan, F. Tan, A. Karande, "Bluetooth based home automation system," Microprocessors and Microsystems, *Elsevier*, vol. 26, pp. 281-289, 2002.

[19] H. Kanma, N. Wakabayashi, R. Kanazawa, H. Ito, "Home Appliance Control System over Bluetooth with a Cellular Phone," *IEEE Transactions on Consumer Electronics,* vol. 49 , issue. 4, pp.1049-1053, November 2003.

[20] Mike Ryan, "Bluetooth: With Low Energy comes Low Security," *WOOT'13 Proceedings of the 7th USENIX conference on Offensive Technologies*, pp. 4-4, 2013.

[21] A. Alheraish, "Design and Implementation of Home Automation System," *IEEE Transactions on Consumer Electronics*, vol. 50, issue. 4, pp.1087-1092, November 2004.

[22] Malik Sikandar Hayat Khiyal, Aihab Khan, and Erum Shehzadi, "SMS Based Wireless Home Appliance Control System (HACS) for Automating Appliances and Security," *Issues in Informing Science & Information Technology*, vol. 6, January 2009.

[23] U. Saeed, S. Syed, S.Z. Qazi, N.Khan, A.Khan, M.Babar, "Multi-advantage and security based Home Automation system," *2010 Fourth UKSim European Symposium on Computer Modeling and Simulation (EMS)*, pp.7-11, November 2010.

[24] Armando Roy Delgado, Rich Picking and Vic Grout, "Remote-Controlled Home Automation Systems with Different Network Technologies," *Centre for Applied Internet Research (CAIR)*, University of Wales, 2009.

[25] Maurice Danaher, D. Nguyen, "Mobile Home Security with GPRS," in *Proceedings of the 8th International Symposium for Information Science,* October 2002.

[26] Bing-Fei Wu, Hsin-Yuan Peng, Chao-Jung Chen, "A Practical Home Security System via Mobile Phones," *TELE-INFO'06 Proceedings of the 5th WSEAS International Conference on Telecommunications and informatics*, pp. 299-304, 2006.

[27] Lili Yang, Shuang-Hua Yang, Fang Yao, "Safety and Security of Remote Monitoring and Control of intelligent Home Environments," *IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC '06*, vol.2, pp. 1149-1153, October 2006.

[28] U. Ali, S.J. Nawaz, N. Jawad, "A Real-time Control System for Home/Office appliances automation, from mobile device through GPRS network," *13th IEEE International Conference on Electronics, Circuits and Systems, ICECS '06*, pp. 854-857, 2006.

[29] Jian-she Jin, Jing Jin, Yong-hui Wang, Ke Zhao, Jia-jun Hu, "Development of Remote-Controlled Home Automation System with Wireless Sensor Network," *Fifth IEEE International Symposium on Embedded Computing, SEC '08*, pp. 169-173, October 2008.

[30] S.R. Das, S. Chita, N. Peterson, B. Shirazi, "Home Automation and Security for Mobile Devices," *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 141-146, 2011.

[31] Leo Kelion, "Breached webcam and baby monitor site flagged by watchdogs," http://www.bbc.com/news/technology-30121159, *BBC News*, 2014.

[32] L. Muhury, A.H.M.A. Habib, "Device Control by Using GSM Network," *15th International Conference on Computer and Information Technology (ICCIT)*, pp. 271-274, December 2012.

[33] Rahul Sasi, "How I DOS'ed My Bank," *Hack in the Box Security Conference (HITBSecConf2013)*, October 2013.

[34] A. Z. Alkar, U. Buhur, "An Internet Based Wireless Home Automation System for Multifunctional Devices," *IEEE Transactions on Consumer Electronics*, vol. 51, issue. 4, pp.1169-1174, November 2005.

[35] Ahmed ElShafee, Karim Alaa Hamed, "Design and Implementation of a WiFi Based Home Automation System," *World Academy of Science, Engineering and Technology*, vol. 6, 2012.

[36] J. Shah, B. Modi, R. Singh, "Wireless Home Appliances Controlling System," *International Conference on Electronics and Communication Systems (ICECS)*, pp. 1-6, February 2014.

[37] Matthias Gauger, Daniel Minder, Pedro José Marrón, Arno Wacker, and Andreas Lachenmann, “Prototyping Sensor-Actuator Networks for Home Automation,” in *Proc. of the 3rd Workshop on RealWorld Wireless Sensor Networks (REALWSN 2008)*, 2008.

[38] *The New York Times* - John Schwartz, “Giving the Web a Memory Cost Its Users Privacy”,[Online].Available: http://www.nytimes.com/2001/09/04/technology/04COOK.html

[39] B. Thomas, "Burnt offerings [Internet]", *IEEE Internet Computing*, Vol. 2, Issue. 6, pp. 84 - 86, Dec 1998.

[40] L.F. Cranor, "Can Users Control Online Behavioral Advertising Effectively?”, *IEEE Security & Privacy*, Vol. 10, Issue. 2, pp. 93 - 96, April 2012.

[41] B. Krishnamurthy, “Privacy leakage on the Internet”, *Proceedings of the Seventy-Seventh Internet Engineering Task Force*, March 2010.

[42] B. Krishnamurthy and C. E. Wills, “Generating a privacy footprint on the Internet”, *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, ser. IMC ’06, New York*, NY, USA, 2006, pp. 65–70.

[43] F. Roesner, T. Kohno, and D. Wetherall, “Detecting and defending against third-party tracking on the web”, *NSDI’12: Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, Berkeley, CA, USA: USENIX Association, 2012.

[44] J. R. Mayer, J. C. Mitchell, " Third-Party Web Tracking: Policy and Technology ", *Proceeding of 2012 IEEE Symposium on Security and Privacy*, San Francisco, CA, USA, May 2012, pp. 413 – 427.

[45] F.Z Borgesius, "Behavioral Targeting: A European Legal Perspective", *IEEE Security & Privacy*, Vol. 11, Issue. 1, pp. 82-85, Feb. 2013.

[46] J. Turow, J. King, C. J. Hoofnagle, A. Bleakley, and M. Hennessy, “Americans Reject Tailored Advertising and Three Activities that Enable It”, *SSRN Electronic Journal*, Sept. 2009.

[47] B. Ur, P. G. Leon, L. F. Cranor, R. Shay, and Y. Wang, "Smart, useful, scary, creepy: perceptions of online behavioral advertising," *Proceedings of the Eighth Symposium on Usable Privacy and Security, ser. SOUPS '12*, New York, NY, USA: ACM, 2012, pp. 4:1–4:15.

[48] S. Saha, "Consideration Points: Detecting Cross-Site Scripting", *(IJCSIS) International Journal of Computer Science and Information Security*, Vol. 4, No. 1 & 2, 2009.

[49] M. Ayenson, D. Wambach, A. Soltani, N. Good, and C. Hoofnagle, "Flash cookies and privacy ii: Now with html5 and etag respawning", *SSRN Electronic Journal*, 2011.

[50] S. Kamkar, "evercookie -- never forget", Sept. 2010. [Online]. Available: http://samy.pl/evercookie/

[51] comScore, "The Impact of Cookie Deletion on Site-Server and Ad-Server Metrics in Australia," January 2011.

[52] J. R. Mayer, "Any person... a pamphleteer: Internet Anonymity in the Age of Web 2.0", *Undergraduate Senior Thesis Submitted in partial fulfilment of the Requirements of Princeton University for the B.A Degree*. Princeton, New Jersey, USA: Princeton University, 2009.

[53] P. Eckersley, "How Unique Is Your Browser?" *Proceedings of 10th Privacy Enhancing Technologies Symposium (PETS)*, Berlin, Germany, July 2010.

[54] T.-F. Yen, Y. Xie, F. Yu, R. P. Yu, and M. Abadi, "Host Fingerprinting and Tracking on the Web: Privacy and Security Implications", *Proceedings of the 19th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, California, USA, 5th February – 8th February 2012.

[55] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, G. Vigna, "Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting", *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, 2013.

[56] K. Mowery, D. Bogenreif, S. Yilek, and H. Shacham, “Fingerprinting information in JavaScript implementations”, Proceedings of W2SP 2011, H. Wang, Ed. *IEEE Computer Society*, May 2011.

[57] J.-L. Gass´ee and F. Filloux, “Measuring Time Spent On A Web Page”, May, 2009, [Online]. Available: http://www.cbsnews.com/news/measuring-time-spent-on-a-web-page/

[58] K. Mowery and H. Shacham, “Pixel perfect: Fingerprinting canvas in HTML5”, *Proceedings of W2SP 2012*, M. Fredrikson, Ed. IEEE Computer Society, May 2012.

[59] T. Kohno, A. Broido, and K. Claffy, “Remote physical device fingerprinting”, *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, April-June 2005, pp. 93–108.

[60] S. Zander and S. J. Murdoch, “An improved clock-skew measurement technique for revealing hidden services”, *Proceedings of the 17th Conference on Security Symposium*, Berkeley, CA, USA: USENIX Association, 2008, pp. 211–225.

[61] D. J. Huang, K. T. Yang, C. C. Ni, W. C. Teng, T. R. Hsiang and Y. J. Lee, "Clock Skew Based Client Device Identification in Cloud Environments," *2012 IEEE 26th International Conference on Advanced Information Networking and Applications*, Fukuoka, 2012, pp. 526-533.

[62] G. Nakibly, G. Shelef, S. Yudilevich, "Hardware Fingerprinting Using HTML5", Cornell University Library (CoRR), abs/1503.01408, Mar. 2015. [Online]. Available: https://arxiv.org/abs/1503.01408

[63] M. Smart, G. R. Malan, and F. Jahanian, “Defeating TCP/IP Stack Fingerprinting”, *Proceeding of 9th Usenix Security Symposium (USENIX ‘00)*, Denver, USA, 2000.

[64] D. W. Richardson, S. D. Gribble, T. Kohno, "The Limits of Automatic OS Fingerprint Generation", *Proceedings of the 3rd ACM workshop on Artificial intelligence and security (AISec’10)*, Chicago, Illinois, USA, October 8, 2010.

[65] A. De Santis, A. G. Gaggia, U. Vaccaro, "Bounds on entropy in a guessing game", *IEEE Transactions on Information Theory,* Vol. 47, Issue. 1, pp. 468 - 473, Jan 2001.

[66] D.C. Feldmeier, P.R. Karn, "UNIX Password Security - Ten Years Later", *Lecture Notes in Computer Science*, Vol. 435, pp 44-63, 1990.

[67] E.I Tatli, "Cracking More Password Hashes With Patterns", *IEEE Transactions on Information Forensics and Security*, Vol. 10, Issue. 8, pp. 1656 - 1665, April 2015.

[68] K. Kato, V. Klyuev, "Strong passwords: Practical issues", *Proceeding of 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, Berlin, Germany, 12-14 Sept. 2013.

[69] D. Hart, "Attitudes and practices of students towards password security," *Journal of Computing Sciences in Colleges*, 23(5), pp. 169–174, 2008.

[70] J. Yan, A. Blackwell, R. Anderson, A. Grant, "Password memorability and security: empirical results", *IEEE Security & Privacy,* Vol. 2, Issue. 5, pp. 25 - 31, Oct. 2004.

[71] S. Farrell, "Password Policy Purgatory", *IEEE Internet Computing*, Vol. 12, Issue. 5, pp. 84-87, Oct. 2008.

[72] S. Rao, B. Jha, G. Kini, "Effect of grammar on security of long passwords," *CODASPY '13 Proceedings of the 3rd ACM Conference on Data and Application Security and Privacy*, pp. 317–324, 2013.

[73] G.J. Johnson, "A Distinctiveness Model of Serial Learning", *Psychological Review*, Vol. 98, No. 2, pp. 204–217, 1991.

[74] G.A. Miller, "The Magical Number Seven, Plus or Minus Two: Limits on Our Capacity for Processing Information", *Psychological Review,* Vol. 63, pp. 81–87, 1956.

[75] E. Rabinovitch, "Staying Protected from Social Engineering"', *IEEE Communications Magazine,* Vol. 45, Issue. 9, pp. 20-21, September 2007.

[76] M. A. Sasse, S. Brostoff, D. Weirich, "Transforming the 'Weakest Link' — a Human/Computer Interaction Approach to Usable and Effective Security", *BT Technology Journal*, Vol. 19, Issue. 3, pp. 122-131, July 2001.

[77] H. Thompson, "The Human Element of Information Security", *IEEE Security & Privacy*, Vol. 11, Issue. 1, pp. 32 - 35, December 2012.

[78] Jonghwa Choi, Dongkyoo Shin and Dongil Shin, "Research and implementation of the context-aware middleware for controlling home appliances," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 1, pp. 301-306, Feb. 2005.

[79] British Broadcasting Corporation (*BBC*) - Andrew Silke, "Webcams taken over by hackers, charity warns", [Online]. Available: http://www.bbc.com/news/uk-22967622

[80] O. Yurur, C. H. Liu and W. Moreno, "A survey of context-aware middleware designs for human activity recognition," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 24-31, June 2014.

[81] S. Saponara and T. Bacchillone, "Network architecture, security issues, and hardware implementation of a home area network for smart grid," *Journal of Computer Networks and Communication*, vol. 2012, pp. 534512-1–534512-19, 2012.

[82] D. M. Konidala, D.-Y. Kim, C.-Y. Yeun, and B.-C. Lee, "Security framework for RFID-based applications in smart home environment," *Journal of Information Processing Systems*, vol. 7, no. 1, pp. 111–120, 2011.

[83] S. Lee, K. N. Ha, and K. C. Lee, "A pyroelectric infrared sensor-based indoor location-aware system for the smart home," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 4, pp. 1311–1317, Nov. 2006.

[84] H. H. Kim, K. N. Ha, S. Lee and K. C. Lee, "Resident Location Recognition Algorithm Using a Bayesian Classifier in the PIR Sensor Based Indoor Location-Aware System," in *IEEE Transactions on Systems, Man, and Cybernetics*, Part C (Applications and Reviews), vol. 39, no. 2, pp. 240-245, March 2009.

[85] P. Kumar and P. Kumar, “Arduino based wireless intrusion detection using IR sensor and GSM,” *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 5, pp. 417–424, 2013.

[86] Y. Zhao and Z. Ye, “A low cost GSM/GPRS based wireless home security system,” *IEEE Transactions on Consumer Electronics*, vol. 54, no. 2, pp. 567–572, 2008.

[87] S. Morsalin, A. M. J. Islam, G. R. Rahat, S. R. H. Pidim, A. Rahman and M. A. B. Siddiqe, "Machine-to-machine communication based smart home security system by NFC, fingerprint, and PIR sensor with mobile android application," *3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, Dhaka, Bangladesh, 2016, pp. 1-6.

[88] P. H. Huang, J. Y. Su, Z. M. Lu, J. S. Pan, “A fire-alarming method based on video processing,” *Intelligent Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 359-364, Dec. 2006.

[89] F. Chen, X. Huang and J. Zhou, "Hierarchical Minutiae Matching for Fingerprint and Palmprint Identification," in *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4964-4971, Dec. 2013.

[90] S. M. Lajevardi, A. Arakala, S. A. Davis and K. J. Horadam, "Retina Verification System Based on Biometric Graph Matching," in *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3625-3635, Sept. 2013.

[91] J. Y. Jung, S. W. Kim, C. H. Yoo, W. J. Park and S. J. Ko, "LBP-fernsbased feature extraction for robust facial recognition," in *IEEE Transactions on Consumer Electronics*, vol. 62, no. 4, pp. 446-453, November 2016.

[92] D. Huang, Y. Tang, Y. Wang, L. Chen and Y. Wang, "Hand-Dorsa Vein Recognition by Matching Local Features of Multisource Keypoints," in *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 1823-1837, Sept. 2015.

[93] J. Tang, J. Luo, T. Tjahjadi and F. Guo, "Robust Arbitrary-View Gait Recognition Based on 3D Partial Similarity Matching," in *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 7-22, Jan. 2017.

[94] Z. H. Tan and B. Lindberg, "Low-Complexity Variable Frame Rate Analysis for Speech Recognition and Voice Activity Detection," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 5, pp. 798-807, Oct. 2010.

[95] J. Galbally, S. Marcel and J. Fierrez, "Biometric Antispoofing Methods: A Survey in Face Recognition," in *IEEE Access*, vol. 2, no. , pp. 15301552, 2014.

[96] H. Park, S. Hwang, M. Won and T. Park, "Activity-Aware Sensor Cycling for Human Activity Monitoring in Smart Homes," in *IEEE Communications Letters*, vol. 21, no. 4, pp. 757-760, April 2017.

[97] M. Skubic, G. Alexander, M. Popescu, M. Rantz, and J. Keller, "A smart home application to elder care: Current status and lessons learned," *Technol. Health Care*, vol. 17, no. 3, pp. 183–201, 2009.

[98] Y. Zigel, D. Litvak, and I. Gannot, "A method for automatic fall detection of elderly people using floor vibrations and sound - proof of concept on human mimicking doll falls," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 12, pp. 2858–2867, 2009.

[99] A. Fleury, M. Vacher, and N. Noury, "SVM-based multimodal classification of activities of daily living in health smart homes: Sensors, algorithms, and first experimental results," *IEEE Trans. Inform. Technol. Biomed.*, vol. 14, no. 2, pp. 274–283, 2010.

[100] K. Doughty and J. Costa, "Continuous automated telecare assessment of the elderly," *Journal of Telemedicine and Telecare*, vol. 3, no. 1, pp. 23–25, 1997.

[101] K. Z. Haigh, L. M. Kiff, and G. Ho, "The independent lifestyle assistant: Lessons learned," *Assist. Technol.*, vol. 18, no. 1, pp. 87–106, 2006.

[102] K. Cameron, K. Hughes, and K. Doughty, "Reducing fall incidence in community elders by telecare using predictive systems," in *Proc. 19th Annu. Int. Conf. IEEE Engineering in Medicine and Biology Society*, 1997, vol. 3, pp. 1036–1039.

[103] L. Atallah, B. Lo, R. Ali, R. King, and Y. Guang-Zhong, "Real-time activity classification using ambient and wearable sensors," *IEEE Trans. Inform. Technol. Biomed.*, vol. 13, no. 6, pp. 1031–1039, 2009.

[104] A. Fleury, M. Vacher, and N. Noury, "SVM-based multimodal classification of activities of daily living in health smart homes: Sensors, algorithms, and first experimental results," *IEEE Trans. Inform. Technol. Biomed.*, vol. 14, no. 2, pp. 274–283, 2010.

[105] T. D. Hunt, D. Rajendran, M. Nikora, S. Bennett, and A. Fendall, "A minimally intrusive monitoring system that utilizes electricity consumption as a proxy for wellbeing," *J. Appl. Comput. Inform. Technol.*, vol. 18, no. 2, 2014. [Online]. Available: https://doaj.org/article/30741addf579403387b4bb 9d3fb2bff3

[106] G. C. Franco, F. Gallay, M. Berenguer, C. Mourrain, and P. Couturier, "Noninvasive monitoring of the activities of daily living of elderly people at home—A pilot study of the usage of domestic appliances," *Journal of Telemedicine and Telecare*, vol. 14, no. 5, pp. 231–235, 2008.

[107] D. Fei and J.-Y. Xiong, "The investigation of the elder's monitoring system based on life supplying line," in *Proc. IEEE Int. Conf. Industrial Technology*, 2005, pp. 314–318.

[108] Y. Yang, J. S. Wang, and J. P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," *Pattern Recognition Letter*, vol. 29, no. 16, pp. 2213– 2220, 2008.

[109] M. J. Mathie, A. Coster, N. Lovell, G. Celler, S. Lord, and A. Tiedemann, "A pilot study of long-term monitoring of human movements in the home using accelerometry," *Journal of Telemedicine and Telecare*, vol. 10, no. 3, pp. 144–151, 2004.

[110] J. Wannenburg; R. Malekian, "Physical Activity Recognition From Smartphone Accelerometer Data for User Context Awareness Sensing," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems* , vol. PP, no.99, 2016, pp.1-8.

[111] J. Wannenburg and R. Malekian, "Body Sensor Network for Mobile Health Monitoring, a Diagnosis and Anticipating System," in *IEEE Sensors Journal*, vol. 15, no. 12, pp. 6839-6852, Dec. 2015.

[112] J. Merilahti, J. Pärkkä, K. Antila, P. Paavilainen, E. Mattila, E. J. Malm, A. Saarinen, and I. Korhonen, "Compliance and technical feasibility of long-term health monitoring with wearable and ambient technologies," *J. Telemed. Telecare*, vol. 15, no. 6, pp. 302–309, 2009.

[113] K. Van Laerhoven, D. Kilian, and B. Schiele, "Using rhythm awareness in long-term activity recognition," in *Proc. IEEE Int. Symp. Wearable Computers*, 2008, pp. 63–66.

[114] T. Maekawa, Y. Kishino, Y. Sakurai, and T. Suyama, "Activity recognition with hand-worn magnetic sensors," *Pers. Ubiquitous Comput.*, vol. 17, no. 6, pp. 1085–1094, 2013.

[115] P. N. Dawadi, D. J. Cook and M. Schmitter-Edgecombe, "Automated Cognitive Health Assessment From Smart Home-Based Behavior Data," in *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 4, pp. 1188-1194, July 2016.

[116] P. Paavilainen, I. Korhonen, J. L¨otj¨onen, L. Cluitmans, M. Jylh¨a, A. S¨arel¨a, and M. Partinen, "Circadian activity rhythm in demented and non-demented nursing-home residents measured by telemetric actigraphy," *J. Sleep Res.*, vol. 14, no. 1, pp. 61–68, Mar. 2005.

[117] Chahuara P., Fleury A., Portet F., Vacher M. "Using Markov Logic Network for On-Line Activity Recognition from Non-visual Home Automation Sensors." In: Paternò F., de Ruyter B., Markopoulos P., Santoro C., van Loenen E., Luyten K. (eds) Ambient Intelligence. *AmI 2012. Lecture Notes in Computer Science*, vol 7683. Springer, Berlin, Heidelberg.

[118] Tapia E.M., Intille S.S., Larson K. (2004) Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In: Ferscha A., Mattern F. (eds) Pervasive Computing. Pervasive 2004. *Lecture Notes in Computer Science*, vol. 3001. Springer, Berlin, Heidelberg.

[119] G. Virone, M. Alwan, S. Dalal, S. W. Kell, B. Turner, J. A. Stankovic, and R. Felder, "Behavioral patterns of older adults in assisted living," *IEEE Trans. Inform. Technol. Biomed.*, vol. 12, no. 3, pp. 387–398, 2008.

[120] M. J. Rantz, M. Skubic, S. J. Miller, C. Galambos, G. Alexander, J. Keller, and M. Popescu, "Sensor technology to support aging in place," *Journal of the American Medical Directors Association*, vol. 14, no. 6, pp. 386–391, 2013.

[121] J. Burke, "Interactive performance environments and the visualization of actor movement," *Digital Creativity*, vol. 13, no. 2, pp. 122–128, 2002.

[122] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E.Jansen, "The gator tech smart house: A programmable pervasive space computer," *Computer*, vol. 38, no. 3, pp. 50–60, 2005.

[123] S. Chernbumroong, S. Cang, A. Atkins, and H. Yu, "Elderly activities recognition and classification for applications in assisted living," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1662–1674, 2013.

[124] B. Johanson, A. Fox, and T. Winograd, "The interactive workspaces project: Experiences with ubiquitous computing rooms," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 67–74, Apr.–Jun. 2002.

[125] P. Rashidi and D. J. Cook, "Keeping the resident in the loop: Adapting the smart home to the user," *IEEE Trans. Syst. Man Cybern.* Part A Syst. Hum., vol. 39, no. 5, pp. 949–959, 2009.

[126] D. J. Cook and M. Youngblood, "Smart Homes," in *Encyclopedia of Human-Computer Interaction*. River Edge, NJ: Berkshire Publishing, 2004, pp. 623–627.

[127] A. Almudevar, A. Leibovici, and A. Tentler, “Home monitoring using wearable radio frequency transmitters,” *Artificial Intelligence in Medicine*, vol. 42, pp. 109–120, 2008.

[128] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz, and D. Hahnel, “Inferring activities from interactions with objects,” *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 50–57, 2004.

[129] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse, “An activity monitoring system for elderly care using generative and discriminative models,” *Personal and Ubiquitous Computing*, pp. 1–9, 2009.

[130] F. Doctor, H. A. Hagras, and V. Callaghan, “An intelligent fuzzy agent approach for realising ambient intelligence in intelligent inhabited environments,” *IEEE Trans. Syst., Man, Cybern.*, Part A: Syst. Humans, vol. 35, no. 1, pp. 55–65.

[131] F. Rivera-Illingworth, V. Callaghan, and H. A. Hagras, “A neural network agent based approach to activity detection in AmI environments,” IEEE Seminar on Intelligent Building Environments, *IEEE Seminar Digests*, vol. v2-9, 2005.

[132] A. Mihailidis, B. Carmichael, and J. Boger, “The use of computer vision in an intelligent environment to support aging-in-place, safety, and independence in the home,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 8, no. 3, pp. 238–247, 2004.

[133] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg, “A scalable approach to activity recognition based on object use,” in *Proc. IEEE 11th Int. Conf. Comput. Vision*, Rio de Janeiro, Brazil, Oct. 14– 20, 2007, pp. 1–8.

[134] A. Aztiria, J. C. Augusto, and A. Izaguirre, “Autonomous learning of user’s preferences improved through user feedback,” presented at the *Proc. 2nd Workshop on Behavior Monitoring and Interpretation*, Kaiserslautern, Germany, 2008, 396, 72–86.

[135] D. N. Monekosso and P. Remagnino, "Behavior Analysis for Assisted Living," in *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 4, pp. 879-886, Oct. 2010.

[136] G. Mokhtari, Q. Zhang, G. Nourbakhsh, S. Ball and M. Karunanithi, "BLUESOUND: A New Resident Identification Sensor—Using Ultrasound Array and BLE Technology for Smart Home Platform," in *IEEE Sensors Journal*, vol. 17, no. 5, pp. 1503-1512, March, 2017.

[137] N. C. Zakas, "How many users have JavaScript disabled?", Oct 2010, [Online]. Available: https://developer.yahoo.com/blogs/author/nicholasc--zakas/

[138] D. Jang, R. Jhala, S. Lerner, and H. Shacham, "An empirical study of privacy-violating information flows in JavaScript Web applications", *Proceedings of CCS 2010*, Oct. 2010.

[139] M. O. Oloyede and G. P. Hancke, "Unimodal and Multimodal Biometric Sensing Systems: A Review," in *IEEE Access*, vol. 4, no. , pp. 75327555, 2016.

[140] Unknown, "Browser Statistics", March 2016, [Online]. Available: http://www.w3schools.com/browsers/browsers_stats.asp

[141] V. Namboodiri, V. Aravinthan, S. N. Mohapatra, B. Karimi and W. Jewell, "Toward a Secure Wireless-Based Home Area Network for Metering in Smart Grids," *IEEE Systems Journal*, vol. 8, no. 2, pp. 509520, June 2014.

[142] A. P. Melaragno, D. Bandara, D. Wijesekera and J. B. Michael, "Securing the ZigBee Protocol in the Smart Grid," in *Computer*, vol. 45, no. 4, pp. 92-94, April 2012.

[143] B. Al Baalbaki, J. Pacheco, C. Tunc, S. Hariri and Y. Al-Nashif, "Anomaly Behavior Analysis System for ZigBee in smart buildings," *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, Marrakech, 2015, pp. 1-4.