University of Bedfordshire

Title : A smart home anomaly detection framework

Name : Edewede Oriwoh

A Smart Home Anomaly Detection Framework

E. ORIWOH

Ph.D

2015

UNIVERSITY OF BEDFORDSHIRE

A Smart Home Anomaly Detection Framework

by

E. Oriwoh

A thesis submitted to the University of Bedfordshire in partial fulfilment of the requirements for the degree of Doctor of Philosophy

1st March 2015

Declaration

I declare that this thesis is my own unaided work. It is being submitted for the degree of a PhD at the University of Bedfordshire.

It has not been submitted before for any degree or examination in any other University.

Name of candidate: Signature:

Edewede Oriwoh

Date:

Mother to son: “So, you say the *fridge* ordered all this chocolate?”

Wede

A SMART HOME ANOMALY DETECTION FRAMEWORK

E. ORIWOH

ABSTRACT

Smart Homes (SHs), as subsets of the Internet of Things (IoT), make use of Machine Learning and Artificial Intelligence tools to provide technology-enabled solutions which assist their occupants and users with their Activities of Daily Living (ADL). Some SH provide always-present, health management support and care services. Having these services provided at home enables SH occupants such as the elderly and disabled to continue to live in their own homes and localities thus aiding *Ageing In Place* goals and eliminating the need for them to be relocated in order to be able to continue receiving the same support and services.

Introducing and interconnecting smart, autonomous systems in homes to enable these service provisions and Assistance Technologies (AT) requires that certain interfaces in, and connections to, SH are exposed to the Internet, among other public-facing networks. This introduces the potential for cyber-physical attacks to be perpetrated through, from and against SH. Apart from the actual threats posed by these attacks to SH occupants and their homes, the *potential* that these attacks might occur can adversely affect the *adoption* or uptake of SH solutions.

This thesis identifies key *attributes* of the different elements (things or nodes and rooms or zones) in SHs and the *relationships* that exist between these elements. These relationships can be used to build SH security baselines for SHs such that any deviations from this baseline is described as anomalous. The thesis demonstrates the application of these relationships to Anomaly Detection (AD) through the analysis of several hypothetical scenarios and the decisions reached about whether they are normal or anomalous. This thesis also proposes an Internet of Things Digital Forensics Framework (IDFF), a Forensics Edge Management System (FEMS), a FEMS Decision-Making Algorithm (FDMA) and an IoT Incident Response plan. These tools can be combined to provide proactive (autonomous and human-led) Digital Forensics services within cyber-physical environments like the Smart Home.

Acknowledgements

I Thank God for His Grace throughout my studies and for bringing things to a Fruitful and Successful conclusion.

I also thank my Supervisor, Dr. Marc Conrad, for his guidance and unwavering support during my studies. I thank Mrs. Conrad for her kind support and encouragement too.

I thank my examiners, Prof. Christopher Nugent and Prof. William Buchanan, for their very helpful feedback during my Viva voce.

I am profoundly grateful to Dr. Antony Brown for his support with translating the Forensics Edge Management System (FEMS) Decision-Making Algorithm (FDMA) into code.

I thank too Dr. Mehmet Aydin, Dr. Paul Sant, Dr. Haider Al-Khateeb, Dr. Gregory Epiphaniou, Prof. Edmond Prakash and Dr. Xiaohua Feng for their support during my studies. Thank you all very much.

Contents

List of Figures

List of Tables

I Dedicate this Work to my Absolutely Fantastic Parents, Mr. and Mrs. Oriwoh, and to my Unbelievably Wonderful Siblings: your Love, Support and Patience know no bounds and I don't deserve such a wonderful family. Thank you all very much.

Chapter 1

Introduction

1.1 Overview

This chapter introduces the Problem Domain, the Definitions of Anomalies, Attacks and Errors as they are used in this thesis, and discusses the Motivation of the work. An early Overview of the Proposed Solution is presented and a Thesis Outline section concludes the Chapter.

1.2 Problem Domain

1.2.1 The Smart Home

Smart Homes (SH) are intelligent, autonomous networked spaces. They provide solutions that enable and support people - for example the elderly and those with health challenges - to live more comfortable lives, by assisting them with their Activities of Daily Living (ADL). SH are supportive Ambient Assisted Living Technologies (AAL) environments and they provide Assistive Technology (AT)[1] services, supporting a variety of people's needs including *Ageing In Place* ([3], page 8) needs, by making use of technological solutions and exposed interfaces to provide the home occupants with resources and services. Dementia sufferers with minimal family/carers support especially benefit from the support systems provided by SH such as reminders.

Since SH enable people to live with the support of technologies, it may mean that SH occupants will not always require physical human support for their healthcare

[1] According to [2], "Assistive Technology (AT) describes products or technology-based services which support those with disabilities or other limitations in their daily activities, enabling them to enjoy a better quality of life".

management, to provide reminders, and other such services. The elderly may therefore find that, because of this available alternative support, they end up living *alone* as sole occupants in their SH.

A typical (Smart) Home is comprised of two (2) major elements: zones (or rooms) and nodes (sensors or devices). In addition to these non-human elements are the human elements i.e. the SH occupants. Connecting these elements to the Internet exposes the home, via its now exposed interfaces, to threats that it hitherto was not exposed to. These threats, in combination with physical attacks (e.g. burglary) that are carried out against homes, form a group of attacks known as *cyber-physical* attacks. Cyber-physical attacks have negative impacts on people generally however the effects or the impact can understandably be worse if it is carried out against a person in the space of their own homes. From the perspective of an end-user, irrespective of how much functionality is incorporated into a SH, there will be a reasonable expectation for these smart spaces to provide - or at least support - adequate levels of both physical and cyber security for their occupants. Wilson, C. et al. [4] highlight that "the acceptability of Smart Homes to users is closely linked to issues of security...". Also, the authors of [5] identify "improved security" as being important to users. This research therefore aims to contribute towards security measures that can be applied in SH to reduce the potential for successful cyber-physical attacks. The development of this model also recognises the need for an easy to use tool which can be managed by home users whatever their level of expertise.

The attributes of SH components (nodes and zones) as well as the attributes of the SH itself are influenced in different ways by the actions of users *around*, and their interactions *with*, these elements. The three main ways in which nodes in an IoT (and Smart Homes) are interacted with and controlled are:

- Direct Touch
- Remote control and
- Learned habit (i.e. pre-programming with updated *learning*)

Different *relationships* can be formed between the SH components depending on how the nodes are interacted with. *Rules* that govern these relationships can also be derived and used to support the design and arrangement of SH to minimize the potential for false alarms being raised or true anomalies being missed.

1.3 Anomalies, Attacks and Errors: Definitions

For the purpose of this work, the following definitions are used:

Definition 1.1. Within the Smart Home (SH) environment, an **anomaly** is any occurrence that does not conform with expected or previously-known patterns or outcomes.

An anomaly can also be described as any element or set of elements in a collection of related or similar elements that does not fit the pattern of a previously-known normal. In a **single occupant home**, a *presence* anomaly can be described as having more than one *presence* in the home at a certain time of a certain day when there should only be one presence. Based on the definitions introduced in Chapter 4 of this work, an anomaly arises when the normality of the "Happy Home Network" elements significantly differ from their normal setting or deviate from a known threshold value.

Example anomalies include

1. A burst pipe;
2. A family member returning home an hour later than usual;
3. An aggressive network probe.

Definition 1.2. An **Attack** is a deliberate, malicious anomaly or an anomaly that is caused on purpose.

In the example anomalies given, an aggressive network probe that is carried out with the purposed of preventing legitimating access to a network is an attack. If a pipe bursts because of the pressure of frozen water inside it, this is an anomalous situation. If it bursts because someone physically destroys it with a weapon, then it is an attack.

Definition 1.3. An **Error** is a fault or a failure.

Example I For illustration purposes, consider the excerpt in Table 1.1 taken from one of the Washington State University Center for Advanced Studies in Adaptive Systems (WSU CASAS) Laboratory Aruba SH datasets (`http://casas.wsu.edu/datasets/`) [1]. The SH being considered is a single occupant SH. During the period when this data was collected, the occupant received occasional 'visitors'.

A visual observation of this data excerpt revealed that on 09/11/2010, during the highlighted periods, two sensors in two different zones were active within 2 seconds of each other. Based on the *Proximals* attribute described in Chapter 4 and, as long as the sensors are all functioning properly and this is not just as a result of a faulty sensor, this data output indicates the *presence* of more than one person in the house. If the assumption is made (for reasons of simplicity) that a second *presence* in a single-occupant home is that of an intruder, then it can be concluded that on that day and time there was a physical intrusion.

Date	**Time**	**Sensor**	**State**	**Activity**
09/11/2010	18:01:08	M018	ON	Meal_Preparation begin
09/11/2010	18:01:08	M022	ON	Motion
09/11/2010	18:01:09	M015	ON	Motion
09/11/2010	18:01:09	M018	OFF	Motion
09/11/2010	18:01:09	M021	OFF	Motion
09/11/2010	18:01:09	M028	ON	Motion
09/11/2010	18:01:10	M013	OFF	Motion
09/11/2010	18:01:10	M019	ON	Motion
09/11/2010	18:01:11	M014	OFF	Motion
09/11/2010	18:01:11	M017	ON	Motion
09/11/2010	18:01:11	M020	OFF	Motion
09/11/2010	18:01:11	M022	OFF	Motion
09/11/2010	18:01:11	M026	ON	Motion
09/11/2010	18:01:13	M018	ON	Motion
09/11/2010	18:01:13	M019	OFF	Motion
09/11/2010	18:01:13	M028	OFF	Motion
09/11/2010	18:01:14	M015	OFF	Motion
09/11/2010	18:01:14	M020	ON	Motion
09/11/2010	18:01:14	M027	ON	Motion
09/11/2010	18:01:15	M013	ON	Motion

Table 1.1: Part of a Smart Home Motion Sensor Dataset [1]

Example II As a second example, a Sensor Activity Chart (Figure 1.1) is used to illustrate how an anomaly can be defined. The anomaly depicted shows sensor M014 (shown in the red bar) being active at the same time as M001, M002, M003, M004 which it would not be under normal circumstances. This anomaly is better understood by looking at the layout in Figure 5.1 and the discussions in Chapter 4 and the work in [6].

The two example activities are recognised as anomalies through the use of a number of factors referred to in the remainder of this work as Attributes. Attributes are used to define the normal security baseline relationships between components of SH (i.e. nodes, zones and occupants) such that if there is a change in these baseline relationships or the thresholds that define them, the activities or events that led to the deviations can be investigated and ascertained as being the causes of the anomalies.

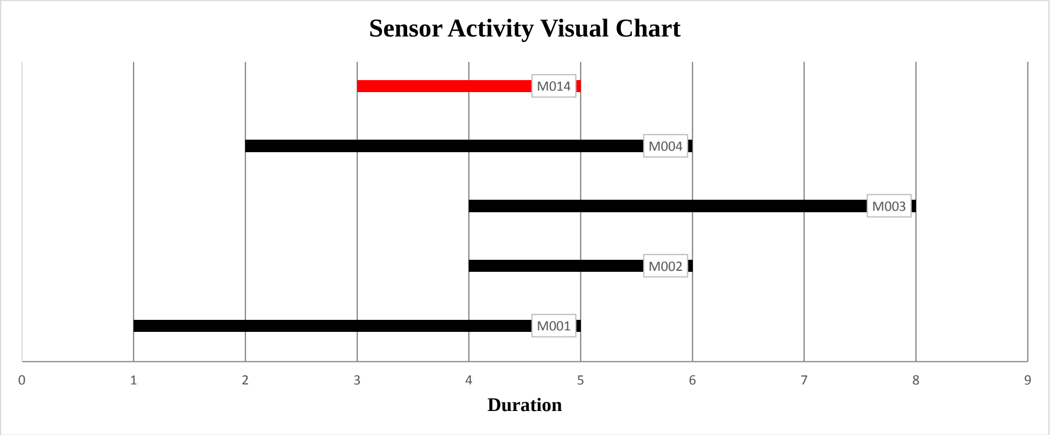

Figure 1.1: A Sample Sensor Activity Chart

Apart from the broad classification of anomalies into *attacks* and *errors*, another classification of anomalies puts them in four (4) classes [7] (see also [8]) :

- intrusive but not anomalous i.e. False Negatives
- not intrusive but anomalous i.e. False Positives
- not intrusive and not anomalous i.e. True Negatives
- intrusive and anomalous i.e. True Positives

Differentiating between attacks and errors is especially challenging in a home environment due to the highly unpredictable nature of humans (i.e. people may change their behavioural patterns thus affecting patterns learned by Machine Learning tools). In this light, Anomaly Detection (AD) systems for SH should therefore be designed to minimise the number of false positives triggered because responding to these false alarms could potentially lead to time, effort/man-power and other resource wastage. However, additionally, AD systems must not be tweaked to such an extent that, in order to reduce the number of False Positive anomalies detected, they end up allowing True Positive anomalies to pass through.

Another example of a Cyber-physical attack within a SH environment can be achieved by an attacker who controls the doors remotely and either locks the occupants in or out of the SH. Attacks that demonstrate that it is possible to control doors in this way was demonstrated in `http://resources.infosecinstitute.com/how-`

hackers-violate-privacy-and-security-of-the-smart-home/ where researchers exploited authentication and code execution flaws to control garage doors as well as obtain information “related to the presence of people in the house”. This real-life cyber attack - with physical manifestations - can have physical access consequences as well as result in data loss. These types of attacks - which were largely hypothetical during the preparation of this thesis - are becoming more prevalent. Another more recent attack involved the use of IoT devices (including cameras and baby monitors) to carry out a Distributed Denial of Service (DDoS) against the Domain Name System (DNS) provider Dyn (http://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/) which affected the services they provide to customers with the attack causing these customers’ sites to become temporarily unavailable and in some cases, intermittently available. The exact figures that demonstrate the prevalence of these attacks has not been collated as at the time of writing however the current trend indicates that due to inherent insecure practices by manufacturers (insecure designs and code), vendors (through insecure installations) and owners of smart Things (not changing default passwords or adjusting default security settings), the attacks are almost certainly going to increase in number.

Within the home, humans may act out of character and, without prior warning, choose to change or modify their schedules. These changes in behaviour, though a normal attribute of humans, may appear erratic and anomalous to Machine Learning algorithms which may erroneously interpret them as attacks. Even if people maintain certain behavioural patterns at their places of work, they may deviate from these patterns in their own homes because within the home settings people do not necessarily have to conform to set standards of behaviour. Falling asleep in the living room may not be anomalous in a home whilst falling asleep anywhere at all in the workplace, irrespective of the location or zone in the work place, is. Therefore AD solutions that are designed for public spaces such as offices may not be suitable for SH environments and tailored solutions that accommodate these characteristic of humans - and are not too restrictive when dealing with humans - may be required.

1.4 Motivation

Smart Homes (SH) are set to become some of the most ubiquitous smart spaces around in the near future. In addition to being intelligent, networked and autonomous spaces, SH primarily serve as abodes or dwelling places for their inhabitants. They have more exposed networking and communications interfaces in comparison with non-SH.

From the perspective of an end-user, irrespective of how much functionality is incorporated into a SH, there will be a reasonable expectation for these smart spaces to provide adequate levels of both physical and cyber security for their occupants. As can be observed from the recent demonstrations on smart car hacks (`http://uk.businessinsider.com/smart-cars-are-vulnerable-to-hackers-2015-7?r=US&IR=T`), exposing hitherto unexposed objects and interfaces to networks such as the Internet and other such networks provides a vulnerability channel that can be exploited by attackers with the right skills for various purposes. The exposed interfaces expand the threat landscape within places that are not normally seen as targets of cyber attacks. Therefore a comprehensive (logical and physical) approach to investigating anomalies within SH is required. Logical solutions can be in the form of security and Digital Forensics (DF) investigatory frameworks and/or methodologies, models and approaches whilst the physical solutions will include any and all cyber-physical security and forensics tools and solutions.

Presence-related anomalies are especially important because of the need to ensure the physical security of potentially defenceless, elderly patients living alone whilst preserving their privacy by avoiding the more intrusive video-based security surveillance tools. However, according to [9] patients would rather prefer it if all technologies designed to support their Activities of Daily Living (ADL) daily living in Smart environments are non-obtrusive. One of the ways in which this research addresses this is the by using the FDMA which detects a *presence* where one should not be; the approach identifies this from real-life SH datasets without visual data such as that from SH Closed Circuit Television (CCTV). This is especially helpful since CCTV is still largely considered to be relatively intrusive security.

This thesis uses original MathWorks® Matlab code to identify, without prior knowledge, the *presence* of an "other" from post-event, SH motion sensor dataset collected from a real-life SH.

1.5 Research Aim and Objectives

The Aim of this research is to develop a Smart Home Anomaly Detection (AD) Framework which supports AD by improving detection potential and/or reducing False Positive detections.

The Objectives of this research are to

- critically review existing Digital Forensic (DF) frameworks, models and methodologies for applicability within the Smart Home (SH) as a Cyber-Physical Environment (CPE);

- develop and validate an IoT Digital Forensics Framework (IDFF) suitable for automating DF processes in the SH with a view to ensuring that relevant evidence is acquired and investigations yield relevant, useful, forensically acceptable and legally admissible results;
- outline the design of an autonomous SH security system (the Forensics Edge Management System (FEMS)) that is based on the IDFF;
- develop a Forensics Edge Management System (FEMS) Decision-Making Algorithm (also written as Forensics Decision-Making Algorithm) (FDMA), as part of the FEMS security system. Demonstrate the applicability of the FDMA through the use of scenarios and analyses;
- combine the approach in this thesis with the work in [10] and, using this combination, demonstrate improvements in AD tasks - or a reduction in False Positive outcomes.

1.6 Overview of the Proposed Solution

The focus of this work is the effective detection of anomalous activities from SH datasets. The approach taken involves identifying a set of attributes, relationships, rules, and constraints which can be combined - alongside the typical AD approach based on user behavioural patterns and activities - to detect anomalies in SH.

This research acknowledges human unpredictability, that people can and do alter their behaviours, patterns and routines thus potentially affecting patterns learned by Machine Learning tools. The work thus avoids explicitly defining or creating tailored user profiles as normal training data for Machine Learning tools. It does not place restrictions on the Activities of Daily Living (ADL) of humans or on the way these activities are carried out. This is because by placing such restrictions there is a risk that alterations in human activity patterns, however minimal or even beneficial to a human subject, may be deemed anomalous behaviour by a Machine Learning tool. In addition, the potential retraining of the system that may be required in order for the ML and AD tools to re-learn the new behaviours might be significant.

In this work instead, an (adjustable) “Happy Home” Profile (Section 4.2) was developed as a support for existing AD approaches that focus mainly on user behaviours and activities. A SH profile is defined based on the data available from the relatively stable relationships between zones, nodes as well as taking the holistic view of the SH and other useful non-human relationships that exist in that context; the actions/behaviours and interactions between nodes as well as the anticipated (predominantly) fixed arrangement of rooms (zones), walls, and certain fixed-location nodes which should be available in most homes to derive stable, balanced (although still quite adjustable) relationships between them such that variations over certain thresholds (e.g. the distance between them, activation times and their response time to user input) can be used to infer anomalies (or otherwise) in the system.

This work considers node-node interactions in addition to user-object interactions and the relationships (or absence of relationships) between nodes and zones in SH. In summary, the following interactions are used:

Box 1.1. Interactions Considered

`Node` ⟷ `Node`, Zone ⟷ `Node`, Zone ⟷ Zone, user ⟷ `Node`, user ⟷ Zone

An overview of the Smart Home AD Forensics Decision-Making Algorithm is shown in Figure 1.2. More details on the elements of the Framework are available in Chapters 5 and 6.

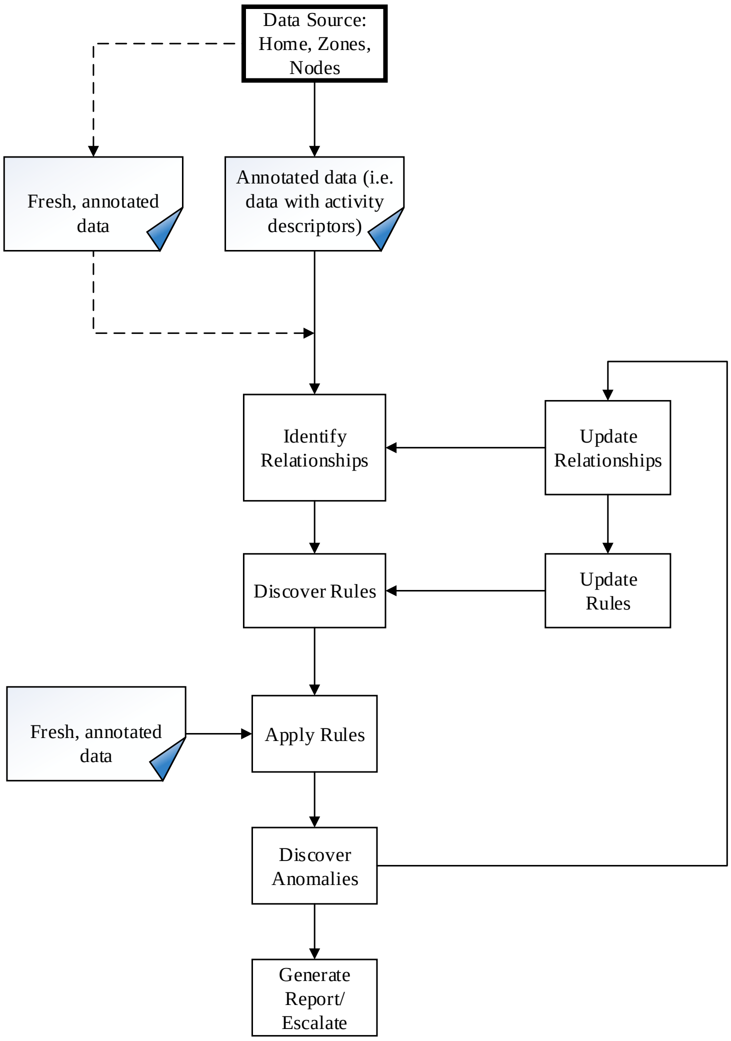

Figure 1.2: The Forensics Decision-Making Algorithm (FDMA) Overview

When a SH is set up, the attributes and relationships between the nodes and zones in the SH are identified and input into the SH FDMA system. This can be done manually to begin with after which the information about nodes and zones is updated automatically by the system. The manual entry can be done by the family and friends, carers, the SH solution providers and in some cases the SH occupants themselves. RFID tags on the nodes in a house can b automatically scanned and the information on them entered into the FDMA database.

The summary of the AD approach shown in Figure 1.2 is as follows: logs acquired from several sources in and around the SH are annotated, for instance using the method described in page 2 of [11]. The approach can be tailored to suit individual Smart

Homes because it is based on the relationships between zones, nodes found in each zone and the relationships formed based on user interactions with the nodes and zones in each home. The data with activity descriptors is then parsed for normal relationships - along with the normal settings input manually by the user. The rules that govern these relationships are harvested. They are applied to fresh data and any anomalies - data that does not fit the relationships, or breaches the rules, and/or are abnormal to user behavioural patterns - are flagged as anomalous. Appendix A shows a subsection of the FDMA - this subsection detects *presence*-related anomalies by making use of several of the relationships and attributes identified in Chapter 4 - location, proximality, nodes and edge node relationships.

For each event the following overall rule applies

Constraints + Attributes + Core Activities + ASeq $\Rightarrow$ *Decision*

1.7 Thesis Contributions

The following are the novel contributions of this work:

- The identification of the **Attributes** of nodes (sensors) and zones (rooms) in typical SH environments and the **Relationships** that these can have with each other. These can be used in conjunction with existing Machine Learning and AD approaches and tools for Anomaly Detection (AD) in Smart Homes.
- **Proximality** which is a term that describes the adjustable relationship that may exist between motion-detection sensors such that their response to stimuli over a given period is not deemed anomalous. Sensors that have this relationship are referred to as *Proximals (ρ)*;
- An **Internet of Things Digital Forensics Framework (IDFF)** which outlines the phases and stages that can be followed during a Digital Forensics investigation in an IoT environment;
- **The Forensics Decision-Making Algorithm (FDMA)**. The FDMA describes how data from and about SH nodes and zones can be used to describe relationships between these nodes and zones, how rules can be extracted from these relationships and how any breaches to these rules can be identified as anomalous. The overview of the algorithm is outlined in Figure 1.2;

- **Event and Activity Constraints**. Event and activity constraints set boundaries on what is acceptable during certain events and activities; these boundaries can include the ***time*** when an activity takes place as well as the ***location***.

1.8 Thesis Outline

The rest of this thesis is organised as follows:
Chapter 2 provides an overview of the Internet of Things (IoT) and Smart Homes (SH) as cyber-physical environments as well as a discussion on the cyber-physical threats to SH environments. It reviews current Anomaly Detection (AD) methods and approaches and highlights the contributions of this work with respect to the existing works.
Chapter 3 presents some of the preliminary research in the area of IoT Forensics that was conducted as part of this research.
Chapter 4 introduces and discusses the attributes that were identified as useful for AD and the AD model that was developed as part of this research work. In this Chapter, an overview of the proposed SH Security solution is also presented.
Chapter 5 covers the evaluation of the Model by applying it to a set of realistic SH scenarios. This is compared with scenarios where the Model is not applied in order to demonstrate any improvements gained through the use of the Model.
Chapter 6 combines the Framework with Core Activities [10] and applies this to anomalous scenarios in order to demonstrate the applicability and improvements gained because of the approach.
Chapter 7 presents conclusions of the preceding chapters, discusses the challenge faced and limitations of the research and discusses potential future work.

1.9 Chapter Summary

The need to support the elderly and those with health challenges in society by encouraging their health management in their own homes necessitates the development of secure Smart Home security models, solutions and tools that can be used to support and enable these solutions. The success of these solutions - from the perspective of health management - depends on how accurate they are at detecting anomalies (errors and attacks) when they occur thus potentially increasing user security and confidence. This work contributes to this goal through the introduction of an AD model for use within Smart Home environments. The model and associated algorithm are used to detect anomalies from datasets of Activities of Daily Living in Single-occupant SH. This research accomplishes this through the discovery of relationships that exist between different elements such as nodes in SH spaces and the application of these relationships in Anomaly Detection (AD). It is achieved by way of Hypothesis Testing. It therefore identifies and defines attributes that can be used to help optimise accurate AD in SH investigations.

Chapter 2

Related Work in Smart Home Anomaly Detection

2.1 Overview

This Chapter provides background information on the Internet of Things (IoT) and Smart Homes (SH). The Chapter also reviews existing multivariate data Anomaly Detection (AD) algorithms and approaches and provides a background on existing work in multivariate sensor data analysis techniques. This is because AD for SH environments will, of necessity, involve handling data from the different variety of *sources* which will be found within SH. This type of dataset made up of different data *types* (i.e. data from different sources) is known as multivariate data. Example data may include motion and temperature sensors readings and TCP/IP data from network monitoring tools. SH contain tools and technologies that support ageing in place and can therefore be useful for enabling the elderly, including those who suffer from dementia, to live supported lives in their own homes rather than being relocated to care homes for them to be looked after. It is therefore essential to ensure that, in addition to the comfort being provided to the home occupants, their cyber-physical security is also ensured with respect to the tools and algorithms which enable the "smartness" of the SH.

2.2 The Internet of Things (IoT)

Advancements in Radio Frequency Identification (RFID), wireless communication, Cloud, virtualisation, mobile computing, sensor technology, Wireless Sensor Networks (WSN), embedded computing and micro-electronics have contributed to the development of the Internet of Things (IoT). "Things" in this context range from traditional computing devices like Personal Computers (PC), to general household objects embedded with capabilities for sensing and/or communication through the use of technologies including, but not limited to, RFID, Bluetooth, Near Field Communication (NFC), Light Fidelity (Li-Fi), and Wireless Fidelity (Wi-Fi). Other systems that support the development of the IoT include social media, smart devices and improved wireless communications speed, cheaper electronics hardware, Big Data, rapid data analysis enabled by faster processing speeds and relatively easy app development tools. The IoT is enabled by the coming together of all these systems. The IoT describes the interconnection of entities (both tangible and intangible) for various purposes including identification, communication, sensing, and data collection.

The phrase "Internet of Things" was coined by Kevin Ashton [12] in 1999. Even though the concept of interconnecting devices and people for various reasons existed before that year - i.e. via the traditional Internet and Social Networks - this model of interconnecting devices and people and with the potential for interconnecting *everything* else is relatively new and is in its introductory stages [13]. In the IoT domain, objects such as baby monitors, cars and tablet computers are equipped with the capability to communicate with each other, providing improved efficiencies for those who use them. Objects that are not of themselves *smart* are being embedded with "smartness" and communication capabilities through the use of technologies such as RFID, sensors and other forms of embedded computing [14]. Communication with such objects will be done by

- direct-touch or physical control;
- remote control methods, for instance, over the Internet using web browsers; and
- 'learned' control or 'sequential actions': sequential actions describe situations where the actions of a Thing cause an action in another Thing; for instance, a user's alarm clock ringing causing their bed to vibrate in a combined "Wake Up" system for getting a user up in the morning.

Control over these interconnected devices will also be spread over a spectrum of stakeholders: owners, manufacturers, law enforcement, governments and other smart Things. IoT Things (also referred to in different literature as Spimes[1] and Blogjects[2]) are meant to be intelligent, autonomous and will be networked into any of a number of different types of *Area* Networks e.g. Personal Area Networks (PAN), Home Area Networks , and Metropolitan Area Networks (MAN) to name a few. The disparate technologies within the IoT are being interconnected in networks which are hybrid and

[1] Objects that exist in Space and within Time

[2] Objects that blog

evolving (i.e. changing their structure). For instance a user's X-box which is part of her HAN can become part of a neighbour's HAN if it is borrowed by that neighbour. These interconnections between smart disparate technologies and devices already offer various benefits and applications to end users, industry, companies and governments. These benefits are evident in areas of transportation, healthcare, Smart Cities, etc. [15, 16]. Cisco estimates that the IoT will offer revenue benefits of $14.4 trillion between 2013 and 2022 [17].

Some of the protocols that currently enable interconnectivity and "smartness" in devices and Things are Internet protocol version 6 (IPv6), Micro IP (uIP/uIPv6), *IPv6 over Low-Power Wireless Personal Area Network (6LowPAN)*, *Constrained Application Protocol (CoAP)*, *Near-Field Communication (NFC)*, *Radio Frequency Identification (RFID)*, *Lightweight IP (LwIP)*, *uC/IP*, and *Tiny TCP*.

These protocols and more are being developed to support the development of the IoT and to help realise the IoT vision.

2.3 Smart Homes (SH)

2.3.1 Overview of Smart Homes

Smart Homes (SH) are intelligent, networked, autonomous abodes or dwelling places [18] and they are set to become some of the most ubiquitous and prevalent smart spaces of the near future. When compared to non-SH, by virtue of their function, SH have an increased number of (**exposed**) communication, control and computing interfaces. These exposed interfaces make SH potential targets for cyber attacks - in addition to the physical attacks such as burglaries that are traditionally faced by homes. The combination of the two classes of attacks (cyber and physical) are referred to as *cyber-physical* attacks. As, arguably, the most personal smart spaces around, cyber attacks in and against SH will potentially have more direct (psychological and physical) impacts on their occupants. Examples of real-life SH as advertised by vendors include the **Loxone Showhome** (`http://www.loxone.com/enen/smart-home/case-studies/showhome.html`) which is described as being controlled from a Loxone miniserver through which the Showhome controls lighting, alarm system, music, air purification, monitors the garage door contact and senses rain and wind among other functions. Another example commercially available solution is **Webee** `http://www.webeelife.com/` which also makes use of a central SH hub that automatically recognises and connects any smart objects around the home. The user can then control any of the devices centrally from a *Webee* app on their phone. The Things which are controlled include "lights, temperature, motion, door locks". Other examples include the **Honeywell Evohome** (http://getconnected.honeywell.com/en/evohome) which allows users to remotely control multiple radiators in a SH; the **Philips Hue** which also provides a software tool that users can operate on their smart mobile devices to control their Phillips Hue lights; **British Gas Hive smart heating system** which allows its users to control their heating and hot water supply via the Internet; **Nest (Learning) Thermostat** (https://nest.com/uk/) and a Home automation provider **Cyberhomes** (https://www.cyberhomes.co.uk/). In addition to automatically monitoring and regulating the temperature in homes by adjusting its settings by itself, the Nest Thermostat learns a user's preferred temperatures and, according to the supplier's website, programs itself accordingly "in about a week". Other tools which are designed to enable the smartness of homes are **WiThings** (http://www.withings.com/uk/en/), **Robomow RC304 electronic Lawnmower** (http://robomow.com/en-GB/), **Sonos** (PLAY:3) smart wireless speaker (http://www.sonos.com/en-gb/home), **Netatmo weather station** (https://www.netatmo.com/product/weather/ and their general site with other SH solutions https://www.netatmo.com/en-GB/site), **Panasonic Home Monitoring and Control Kit** (http://www.panasonic.com/uk/consumer/smart-home/kx-hn6012ew.html), **Samsung Family Hub** (http://www.samsung.com/global/ces2016/familyhub/), **Dojo** (which glows to let users know when they are being hacked http://www.digitaltrends.com/home/the-dojo-protects-your-smart-home-from-cyber-attacks/), **Nesspresso Prodigio coffee Machine** (https://www.nespresso.com/uk/en/prodigio-machines-range?cid=SEM_B2C_UK-EN_LOC_R_Google_UK.Brand.Machines.BMM_Prodigio_nespresso.prodigio_Broad), **Philips Hue Phoenix LED lamp** (http://www2.meethue.com/en-us/productdetail/philips-hue-phoenix), **Dyson Pure Cool air purifier** (the manufacturers claim it "automatically monitors, reacts and purifies" the air in the home http://www.dyson.co.uk/fans-and-heaters/purifiers/dyson-pure-hot-cool-link.aspx), **Chamberlain MyQ Internet Gateway** is an Internet-based remote control of garage doors (http://www.chamberlain.com/smartphone-control-products/smartphone-connectivity/myq-internet-gateway), **Smarthings hub** (https://www.smartthings.com/uk/), Ubi (which is a voice-controlled device for answering questions and controlling SH https://www.kickstarter.com/projects/607691307/ubi-the-ubiquitous-computer-voice-activated-and-al-0), **Winkhub** which is a central control device for home automation products (http://www.wink.com/products/wink-hub/) and **Winkrelay** (http://www.wink.com/products/wink-relay-touchscreen-controller/), Lifx (Wi-Fi enabled LED smart bulb http://www.lifx.com/), **ismart alarm** (https://www.ismartalarm.com/), August smart lock (https://store.august.com/), **Lowe Iris** (https://www.irisbylowes.com/), **Canary all-in-one SH security system** (https://canary.is/) which, although claims to provide overall security for all nodes and all sensor types on a smart homes, requires a home to be empty to function.

Other SH are those set up by various institutions for research purposes. These include the WSU CASAS Smart Home http://casas.wsu.edu/ and the http://ailab.wsu.edu/mavhome/, http://lass.cs.umass.edu/projects/smart/, The Aware Home http://awarehome.imtc.gatech.edu/, PlaceLab , Duke Smart Home Program http://smarthome.duke.edu/, iSpace at the University of Essex (http://cswww.essex.ac.uk/iieg/idorm2/index.htm), GatorTech Smart House (https://www.cise.ufl.edu/~helal/gatortech/index2.html),

In non-SH, end users make decisions on a daily basis, however routine, which they themselves follow through on whereas in a SH, the users train the home either by giving it explicit commands and instructions when the SH network is being set up and as the nodes (devices) are being installed and configured; by their regular activities and behaviour patterns which the home can learn using any of several machine learning tools or; by modifying any learned patterns by explicitly changing the previously learned pattern of users. These learned patterns are then utilized by the home to maintain a habitable environment by managing all the conditions as required by its occupants. This habitable status is referred to as a "Happy Home" status in this work Section 4.2. The benefits of this kind of home are numerous: a time-saving benefit as users are free to work on other aspects of their daily living while the home manages and maintains their home life and schedule; cost saving - the home can get to work with refreshing itself through cleaning and doing the laundry during periods of the day when energy cost is lowest; better and timely maintenance because the home will more easily point out faults or damages depending on its design which a home owner may not observe until a while after; reduced potential for home accidents due to accidents or human error. For instance, a user might forget an iron plugged in, a kettle connected, their front door unlocked or their baby monitor remotely streaming video online (even when all family members are physically present in the house and there is no need for remote streaming). After the user's pattern has been learned these scenarios can be better managed by the home: it can turn off the iron and kettle after a fixed period has elapsed. It can lock the front door at a time that it has learned that the user is typically away and it can halt streaming of a 'live' baby monitor feed when it accurately detects that the baby's guardians or family are physically present in the home along with the baby that is being monitored.

Non-SH of today can and are being made smart through the addition and use of bolted-on technology. Alternatively, SH solutions might in future be provided as comprehensive offerings by single vendors who will offer and provide the different solutions (water, electricity, health, security, money, transportation, etc.) as a holistic package.

2.3.2 Threats, Attacks and Security Challenges

Homes typically serve various important functions; they serve as a place for short to long-term stay, a place to store property, a place to spend time with short- and long-term co-dwellers, a place to find privacy and even to find and enjoy some security. The home typically represents a place for rest, family, work, a space to age instead of going into care, a trusted domain and even a domain of absolute authority and control. Homes can be described essentially as the ultimate personal space. Threats and attacks against the home can therefore adversely affect people's confidence and comfort within this most personal of spaces. Countering these threats (especially the cyber and cyber-physical threats which are not the threats that homes typically face) will require technological solutions. Existing and emerging cryptographic, authentication methods being proposed by research are part of the effort being made to provide protections against these threats however other solutions are required such as effective detection tools against ongoing attacks and investigatory procedures which can be applied *after* successful attacks.

The development of strategies and solutions for securing the SH is important for several reasons. Humans now spend more time in their homes especially with the flexibility allowed them by an increasing number of employers who encourage working from home. People also spend more time working from home nowadays because it is now *possible* to do so and still be as effective as if they were in the office. Working from home saves on the cost of renting office space, minimises travel time to the office location, can provides savings on childcare and electricity bills, cuts out the need for other office maintenance staff such as the cleaners, minimizes the cost of utilities and stationery and eliminates the need to cater for the general security of the staff and the infrastructure. The home is therefore no longer the space where things e.g property are kept or the place where visitors are entertained and where family units can abide as, and when, needed. It is now increasingly a *functional* part of the family daily life - and is increasingly a potential repository and, therefore, source of company files and data. The value of the home as a potential target for corporate espionage and blackmail has increased with the introduction of smart technologies. In addition, the potential for assets within it to be accessed remotely has made it a more viable and useful target for malicious parties whether they wish to harm the family or the company that the family members are associated with.

Providing adequate security measures in a SH can be useful as part of a *Digital Forensics readiness* policy. The evidence from the nodes in smart homes could prove vital to digital forensics investigations of crimes carried out against or by nodes operating autonomously from the home. In addition, some insurance claims may require that SH users have a certain level of cyber security set up in their homes for their claims to be valid in the event that they are targets of successful cyber attacks. There is some overlap in these attacks.

From a security standpoint, there are many attacks that SH will face as they become more ubiquitous. Some of these Cyber-Physical attacks which may require Cyber-Physical forensics responses are discussed next. For the purpose of clarity the identified attacks are divided into groups. The security (confidentiality, Integrity and Availability or *CIA*) impacts of these attacks are also identified. With respect to real-life attacks, whilst there are particular examples of attacks against - or making use of - Smart Home nodes such as baby monitors and security cameras, as at the time of writing, there was no representative data which shows the prevalence of such attacks.

2.3.2.1 Smart Home Attack Groups

A. **Removal attacks**

These groups of attacks include any and all unauthorized extraction of legitimate assets from the home. Theft and burglary are examples of attacks in this group. Assets may be physical (cars, game consoles) and logical (digital family photos, electricity supply). Some existing United Kingdom laws that apply when such illegal activity is identified are the Theft Act 1968, the Data Protection Act 1998 and even the Computer Misuse Act 1990.

The CIA impact of this type of attack includes a general loss of trust in the entire home security system as well as a feeling of defilement and of being invaded because of the personal nature of a space like the home.

B. **Insertion Attacks**

An insertion attack can be physical or logical and can take several forms. For example, a rogue device such as a smart meter can be added to a home network for malicious reasons. Baig in [19] describes such a scenario as a “device implant attack”. Another potential attack approach is a situation where a trusted device is inserted into a network but ‘ghosted’ or hidden from, for example, the power grid so that the energy consumed by this device is not visible to the smart meters and thus the consumer is not charged for the electricity consumed. Botnets [20] which will require the insertion of malicious code on the bots or nodes in the home are another example of insertion attacks. A *presence* during an event or activity where there should not be one is another example. Shopping lists can be modified by an insertion attack through Man-In-The-Middle (MITM) attacks. This attack can be done to replace the list that is being sent between the fridge and the local grocery store so that the wrong items are ordered.

With respect to the CIA impact, a successful insertion attack carried out against a SH can lead to a general loss of trust in the security tools and solutions provided by SH security vendors. This type of attack compromises the Integrity of the SH network and any nodes that are directly affected (e.g. the shopping list).

C. **Modification attacks**

Any and all damages to home appliances fall within this group of attacks. Corruption of meter readings between the nodes and the smart meter in order to reduce the amount the user has to pay is also an example of a modification attack. The replacement of a trusted device with a rogue device to enable spying is a form of network modification. A secure, 2-way authentication scheme between the fridge and the local store is a potential remedy to this situation. Skimming, Cloning [21], Spoofing [22–24], Damage [25], Jamming [26] are modification-type attacks.

A loss of trust in the system (due to a loss of Integrity and Confidentiality) as well as a loss of access to resources (Availability) is affected by these types of attacks.

D. **Observation attacks**

A passive form of attack, the observation attack involves the monitoring of the smart space. An observation attack can form part of the reconnaissance phase of a physical attack i.e. preparations before actual attack. This can give the observing party an idea as to who is home, if anyone. Another observation method may involve installing remote video monitoring tools e.g. hijacking CCTV streams in order to physically view the home and its occupants. The reader is directed to [27] for discussions on Cyber-Physical security issues. An electricity meter can be sued to monitor their electricity usage.

A loss of Confidentiality and Integrity are the major CIA impacts of this attack. Additional negative impacts of this type of attack includes a loss of trust in the home security systems and general discomfort.

Attacks in Cyber-physical Environments can also be grouped into ***Physical Attacks*** (*e.g. Theft*) and ***Logical Attacks*** (e.g. DoS). This grouping is useful for investigators who may wish to analyse what assets (physical and/or logical) may have been impacted during an attack.

The effects of attacks on SH inhabitants will range from a general sense of insecurity and a loss of privacy to the erosion of trust in the SH solutions and the solution providers or vendors. These psychological impacts can lead to a negative perception of the companies that provide the devices and services especially those whose products are impacted most by the attacks. For example, if a live feed from a remotely accessible baby monitor is unlawfully accessed, the baby monitor supplier as well as the home network's ISP can potentially lose customers and, therefore, revenue.

More information on the security challenges faced by Cyber-Physical Systems (CPS) can be found in [27]. Example attempts to hack smart fridges are chronicled here `https://www.pentestpartners.com/blog/hacking-defcon-23s-iot-village-samsung-fridge/` and `http://resources.infosecinstitute.com/how-hackers-violate-privacy-and-security-of-the-smart-home/`. In the first link the attackers attempt, among other things, to add calendar entries to the Fridge's calendar - without permission - and to cause the fridge to accept a compromised firmware update. The second link discusses successful attacks against Smart Televisions, Smart Meters and Smart Light Bulbs. It also discusses the threats posed by insecure Baby Monitors (several brands have been found to have security flaws according to the article) in domestic environments.

2.4 Anomaly Detection: Multivariate Sensor Data

Anomaly Detection (AD) within SH environments will be based on data aggregation from the different data sources of the data evidence around the SH as well as data analysis. This necessitates the need for a background discussion on Data Mining, Machine Learning and a review of work which discusses data analysis for anomaly detection and even activity recognition because the mechanism for recognising activities can be extended to recognise activities including the ones occurring in anomalous contexts. Anomaly Detection within SH will be addressed through the analysis of data - whether live as the anomalous event is occurring or after an event has occurred. Therefore approaches which consider the analysis of data from sensor nodes is reviewed in this section as part of the background work for this research.

Multivariate Data is data that is made up of different data types and formats. An example of multivariate data is a dataset containing temperature readings and motion sensor data. Data outputs from SH will, of necessity, be multivariate because of the variety of sensors types available in SH. Any approach to AD for SH will therefore have to be able to accommodate different data types. Two main fields that are playing a key roles in the development of AD methods are *Data Mining* and *Machine Learning*.

2.4.1 Data Mining

Data Mining is described as the "core stage of the Knowledge Discovery process" [28]. It involves analysing data in such a way as to derive useful information from it.

This information may be a set of relationships or rules that exist within the data but which may not be apparent from any visual observation of the data when it is first encountered. Data Mining helps with the discovery of useful correlations with data or between different types of datasets: relationships based on the contents of shopping baskets from shopping basket datasets have helped stores make decisions about where to display certain products on the shelves in their shops and during what periods. This can help such stores increase revenue by helping customers find items they predominantly buy together faster as well as cut cost because there is a reduced need to heavily advertise and and promote different sets of products during certain periods when they are not likely to be in demand or bought. Data Mining is further defined and discussed in [28]. Data Mining is described as the "most important part" of the CRISP-DM process. CRISP-DM (Cross-Industry Standard Process for Data Mining Methodology) "defines the crucial steps of the Knowledge Discovery process". DM is being applied in several ways to find solutions to issues in SH and support SH occupants. [29] discussed and applied DM in Activity Recognition (AR). AR is useful if used to support AD. For example, if an unexpected activity is detected - or, conversely, an expected activity does not happen - and the AR system flags that as abnormal, then the AR system would have served the purpose of an AD tool as well.

2.4.2 Machine Learning

Machine learning is a field that involves the use of datasets to train algorithms in preparation for future data based tasks such as activity recognition and prediction. There are several tools used for Machine Learning tasks: Neural Networks including K-Nearest Neighbour (KNN), Support Vector Machine (SVM), Time-Series Analysis as well as the more recently developed method of Ontology-based modelling. There are two main approaches to Machine Learning: Supervised Machine Learning and Unsupervised Machine Learning; these are discussed in the sections that follow. Machine Learning algorithms are used to support Anomaly Detection (AD) by detecting outliers in datasets which can be from SH nodes.

According to [28], there are two approaches in ML: ***Symbolic*** and ***Statistical***. The Symbolic approaches deal with using symbolic tools to represent the relationships that exist within data whilst the Statistical approaches make use of machine learning tools to build models that are based on the relationships that exist within data. A detailed discussion on how the two approaches relate to each other is available in [28]. In Section 6.5 the Symbolic approach is taken in the induction of rules from two datasets. The rules represent the relationships that exist within the dataset analysed.

2.4.2.1 Supervised Machine Learning

In *supervised* Machine Learning, labelled data is fed into a system which is then trained to “learn” explicitly what is normal and what is anomalous. The aim is for a model to be built from this “learning” process so that when clean, unlabelled data is passed through the model, it will be able to identify which class any new data that is fed into it belong to. This type of learning is also known as *Classification* and the model developed is also referred to as a classifier. Example classes are normal and anomalous. One tool which is used for data classification is the Support Vector Machine (SVM). SVM implementations can be found in Matlab as well as in the R language software tools such as RStudio and Weka.

In data classification, each occurrence in a dataset is labelled with the class it belongs to e.g. data can be labelled as “fruits” or “vegetables”. This data is then fed to an algorithm which is trained to recognise the individual groups of each occurrence of the data. If similar but fresh, unlabelled data is made available for which the classes of some occurrences in the data are not known, this data can be fed into the algorithm which will then produce an output that labels the data as belonging to one class or another based on the rules learned by the algorithm from the original dataset. This type of learning is Supervised learning because the set of possible classes is known in advance and made available to the algorithm. This is useful for AD purposes as the anomalous data points are identified and known however some challenges with using this for anomaly detection is that sufficient, accurately-labelled training data has to be made available to the algorithm to be trained. In a SH, this type of prior data may not exist before the home is set up and anomaly detection will need to be initiated as soon as the home is set up i.e. all the nodes and zones are configured and interconnected.

2.4.2.2 Unsupervised Machine Learning

In *Unsupervised* Machine Learning, the system operates *unsupervised* which means, it is not explicitly trained with prior information about which instances in the dataset being analysed is normal and which instances are anomalous. Unsupervised Machine Learning typically passes the data through an algorithm which calculates the separation between the data points. The separated points which can be separated, for instance, based on their *similarity* to each other, are placed in **clusters** that best represent, based on the parameters made available to the algorithm, which groups the datasets belong to. There are several unsupervised learning algorithms that are available in software tools such as MATLAB and the R language. Unsupervised Machine Learning is also known as *Clustering.*

The inference of which grouping is normal and which is anomalous can then be made from a physical examination of the results or using an automatic labelling scheme. The aim of this thesis is to support unsupervised AD by proposing and testing several attributes and relationships as identified in Chapter 4 which can be used to build a baseline secure SH Model such that any deviation from the set normal is an anomaly.

2.4.2.3 Data synchronisation

Data synchronisation is also important for accurate AD. Synchronisation between the clock or timers on a central SH security monitoring system such as the FEMS and the nodes around the home that feed data to it. This will help eliminate any uncertainty about the times that any data was generated. In addition the FEMS should be synchronised to any external forensics companies systems or cloud backup point with regular updates of any data being collected sent through a secure route to the external system. This route can be encrypted and backups of data made; backups should not be made at fixed intervals to avoid theft or corruption of the data by any malicious parties monitoring the network who may detect the regular backup patterns which can make it easier for them to prepare for their attacks and also make it more likely for them to succeed.

2.5 Current Multivariate Anomaly Detection Techniques

The use of automated tools to expedite the detection of anomalies in real-time or from Smart Home (SH) datasets is an issue that is gaining growing attention for several reasons: the function of the home as a place of security and privacy; healthcare management reasons; time-saving; tools, services and product improvement for vendors; cost-saving, for example through better and smarter use of electricity; Green Living; supporting ADL within safe and secure environments, among others.

Existing AD methods focus on several methods of intrusion detection: the use of Machine Learning to recognise patterns using either supervised or unsupervised (classification or clustering) tools, methods and techniques to highlight the presence of outliers which are occurrences in a dataset which deviate from a particular more highly generalised and identifiable pattern. Some AD methods are discussed in this section.

2.5.1 A Data Mining Framework for Activity Recognition in Smart Environments

This work develops a framework for selecting and extracting useful features and uses these features to recognise activities from SH datasets. The part of their work that is closest to the work in this thesis is the features that they identified as being helpful in energy prediction which is their focus in this work. The features were identified using their feature extraction module in their Data Mining system which is comprised a Data Collection, Data Annotation, Feature Extraction and an Activity Recognition Module. The features are the Activity length (in seconds), Time of day, Day of week, weekday or weekend, previous and next activity, number of motion sensors involved, total number of times the motion sensors triggered, energy consumed in Watts, List of motion sensors and their states [29]. Some of these features are similar to the ones applied in this work (in this thesis they are referred to as Attributes).

2.5.2 A Knowledge-driven approach to Activity Recognition in Smart Homes

Chen at al. [32] introduce a knowledge-driven method for recognising activities from SH data acquired from different types of sensors (multivariate data). They highlight the fact that there is some correlation between ADLs and the contexts within which they occur. They introduce a generic conceptual sensor model which is used to create *domain ontologies* using tools such as Protege (`http://protege.stanford.edu/`) which enables the creation of classes, sub-classes, relationships between these, and carry out testing through the use of Reasoning tools such as Pellet and FaCT++ (`http://owl.man.ac.uk/factplusplus/`) and HermiT [33] (`http://iswc2004.semanticweb.org/posters/PID-ZWSCSLQK-1090286232.pdf` and `http://www.sciencedirect.com/science/article/pii/S1570826807000169`). They represent activities as “concepts”. Some of the work in this thesis is based on a similar premise - see Chapter 6. They make use of Ontological Modelling for activity recognition. They illustrate this system using the MakeDrinkADL class. They define several parameters for testing the accuracy of their system: these include the *Time per Recognition operation (TpRO)*, the *Fine-Grained Recognition Accuracy* and the *User-Object Interaction Recognition (UoIR)*. The TpRO is an indicator of how much time it takes the system to recognize an activity for what it is after a sensor has been activated. The Fine-Grained Recognition Accuracy is a measure of how well their system recognises an activity as a generic activity while the activity is on-going as well as how well the system identifies the specific *user* performing the activity and so can act as a reminder system which can remind the user which nodes might be useful for the particular activity they are performing. The UoIR is a metric for checking how reliable and efficient the activity monitoring system is. With all three metrics they achieved results in the highest percentile and thus demonstrate their system is relatively accurate as far as activity recognition is concerned.

This thesis agrees with their premise that activities can employ **certain** nodes and zones as part of their *normal* occurrence thus placing measurable constraints on activities and events. These constraints can be harvested and used to form part of a SH security baseline. The use of constraints to support AD is discussed in Section 4.2.2.

2.5.3 Using Temporal Logic and Model Checking in Automated Recognition of Human Activities for Ambient-Assisted Living

The work in this paper introduces some formalisms and an ARA (Automated Recogniser of ADLs) algorithm [34]. The formalisms are drawn up from the perspective of temporal logic and are are used for Activity Recognition. The approach considers the order of actions and recognises that some actions may be optional in certain activities. Using said formalisms, the authors developed several models which represent activities being carried out correctly. They also demonstrate the AD applications of the formalisms. This paper covers work that resembles the one in this thesis apart from the use of different formalisms, this thesis concentrates on building a model based on secure attribute states such that real-time network monitoring and anomaly detection based on any change of states is the goal. Temporal Logic was not used in this thesis because it is used to address relationships between activities based on time while this thesis addresses relationships between nodes, zones, activities, events based, not only on start and end times and durations of activities and events, but also on location, modes, proximality, among other attributes and relationships as discussed in Section 4.2.1. Rather than combine the Temporal Logic formalisms with the attributes and relationships identified in this work - which would have led to an overlap - the explicit time of day and date were instead used and the period was encapsulated in the proximality attribute.

2.5.4 Mining Correlation Patterns among Appliances in Smart Home Environment

The work by Chen et al. [35] identifies correlation patterns in SH datasets using the Correlation Pattern Miner (CoPMiner) algorithm that they developed. This work is the *previous work* of [36] which is discussed in Subsection 2.5.5.

2.5.5 Significant Correlation Pattern Mining in Smart Homes

The authors developed the Correlation Pattern Miner algorithm (CoPMiner) which probabilistically discovers the patterns in which appliances are used as well as the correlations among these appliances [36]. Their work is an improvement on most existing work which focus on the patterns of *single* appliances. The CoPMiner algorithm - which they tested on their synthetically-generated datasets and showed that it out-performed the other algorithms CTMiner [37], TPrefixSpan [38] and IEMiner [39] - transforms usage-interval data into what they refer to as **usage representation** (Page 6, Figure 2 in [36]) which they describes as an improvement on Allen's Thirteen Temporal Relations [31]. After the transformation into usage representations, CoPMiner prunes all infrequent usage points (based on a support threshold) from this usage representation. The time information for all the frequent usage points are then identified. Two other algorithms that they developed in support of CoPMiner discover all correlation patterns (UPrefixSpan) and output all the discovered correlation patterns (TPMiner). Pattern mining is important for AD because patterns of user activities scan be stored and compared against in real-time even as users perform their ADLs. Any deviations detected can trigger alerts.

2.5.6 Practical Anomaly Detection based on Classifying frequent Traffic Patterns

The work by Paredes-Oliva et al. [40] discusses the detection and classification of anomalies in network traffic through the use of *Frequent item-set mining* and *decision-trees* to detect anomalies. Their work produces results of up to 98 percent classification accuracy. Their work is however based on the traditional Train-Classify-retrain-detect approach, an approach which suffers from the same weakness of requiring accurate, substantial, training data sourced from the SH environment being investigated. In addition, the training phase(s) require adequate training periods (time) to be carried. Any malicious injection of anomalous data during the training phase - which will be accepted as a legitimate frequent item-set - will lead to a faulty baseline which can in turn lead to future anomalous network traffic being flagged as normal thereby giving rise to anomalous outputs form the SH AD model.

2.5.7 Detecting Anomalous Sensor Events in Smart Home Data for Enhancing Living Experience

In this work One Class Support Vector Machines (OCSVM) are applied as an AD platform by the authors [11]. Their solution is implemented using the LibSVM tool in Weka (`http://www.cs.waikato.ac.nz/ml/weka/`). The method is supervised and involves training their algorithm with labelled data and they were able to demonstrate positive results by feeding in un-labelled test data. Their main achievement in this work is the evaluation of the OCSVM as a tool for AD. Using this train-and-test method they demonstrate that they are able to detect several anomalies using OCSVM. They evaluate their approach using five (5) test measures (Precision, Recall, F-measure, Type I and Type II errors) - although they explain that Type II error are not considered. The approach in this thesis differs from the work in [11] in the fact that the work in this thesis does not require training data but that data analysis (Chapters 5 and 6) is based on the features and relationships present in the data.

2.5.8 A Model for Discovering Correlations of Ubiquitous Things

The work in [41] agrees with the premise in this thesis that Smart Things (also referred to as nodes throughout this work) have relationships with other Things based on several attributes. In addition, they support a premise in this thesis that user interaction with Things produce *events* (see Chapter 6 for a discussion on events and activities as applied in this thesis). These events are useful for capturing the relationships that exist between Things in a SH. In addition, similarly to the work in this thesis, they identify the spatial attributes (i.e. fixed and mobile) of nodes around smart spaces as being important for identifying their activities. They developed a model for discovering the relationships between things using Modularity-Based Community Detection [42] which they describe as “an effective measure for community structure in many complex networks”. They propose a method for discovering correlations between Things and accomplish this by capturing information about the user, temporal and spatial information of each Thing being used. They acknowledge the challenge of discovering relationships between Things. They build spatio-temporal (location-time) graphs to capture the relationships between Things as influenced by their location and the time, as well as a social graph that captures user interactions with Things. The definitions and further information on the graphs can be found in [42]. Their results demonstrate the importance of *location*, *time* and *user interactions* to the discovery of hidden relationships between Things and these elements are among the attributes applied in this thesis.

2.5.9 Dynamic Sensor Data Segmentation for real-time Knowledge-driven Activity Recognition

This work focuses on achieving continuous, real-time activity recognition through the implementation of a sensor data segmentation method [43]. According to the authors, their work extends the work by Chen at al. [32] by introducing the sensor *data segmentation* element. Their system is divided into three (3) main function areas (in addition to the sensor data segmentation time windows that they introduce): *Context Selection*, *Iterative Action inference* and *Activity Recognition* layers. The ***context*** of any action or activity within a Smart environment - and in the field of activity monitoring - covers elements of the action such as the *time of day*, *location*, *objects* used, *potential activity* that the accessed node might be used for (or, with respect to their work, which specific *ADL*). They test their system on synthetic SH data that they generated from the Synthetic ADL Data Generator that they developed.

They define a *time window* and its parameters such that they are able to adjust the window to accommodate or eliminate sensor data from a stream of input data sent to their system thereby segmenting the data as appropriate and based on feedback and output from the system, to enable an effective real-time activity recognition system for a SH. They implemented this in the form of a bespoke tool and their recognition accuracy results show that the system effectively recognises some activities to a high degree whilst others are not as accurately identified. They attribute one of the reasons of the low recognition accuracy of certain activities to the static nature of the time window used in their simulation; this is because data that should be in more than one ADL may become “merged within a time window”. This is a shortcoming for their method. The significance of this shortcoming can be demonstrated through a visual observation of the data from a real-life SH network (see the Aruba dataset excerpt in Table 1.1). From observation, the nodes in the home do not go **on** and **off** in neat, precise, activity patterns and a motion sensor, for example, may still be in state *on* for some time after a home occupant has vacated the location having passed by the motion sensor in question. This means that it is possible that even if a user has, for instance, stopped brushing their teeth and has moved to the living room to turn on the television, the *BrushTeeth* event might still be active as well as the motion sensors in the bathroom. This is not the case in this work because the Living Room and the two Bathrooms in the Aruba SH (which is the focal SH in this thesis) are non-proximal zones. Information on proximals is available in Section 4.2.1.3. The Aruba SH is described in Figure 5.1.

They also assume that certain nodes become *active* at certain (fixed) periods after each other. This is however not representative of how real sensors function and two nodes (e.g. `CoffeePot` and `Cup`) may go to state *on* 5 seconds after each other on a particular day and go to state *on* 20 seconds after each on another day and this disparity might adversely affect the output of their tool since the flaw is in the data generated. A better approach to their investigation may be to test their tool on any of the widely-available real-life SH sensor datasets. It is clear that no issue is taken with their solution or approach but with the data generation method they employ which can be better defined and developed to more realistically represent sensor behaviours. Another shortcoming of the tool is that it seemingly ignores the fact that certain nodes do not power *on* only when the user aims to actually use them and therefore a node coming **on** at a certain time may be seen as the sign of an actual ongoing activity and this may, in fact, be inaccurate. For instance, a mapping of activity *PickUpCup* to event *MakeTea* may be inaccurate because a user may carry out the activity, change their mind and pick up a flask instead in order to complete the same event i.e. *MakeTea* except that this time, they plan to store the tea in a flask.

Chapter 6 contains more details on how Events and Activities are applied to Anomaly Detection scenarios in this thesis.

2.5.10 Smart Homes for the Elderly Dementia Sufferers: Identification and Prediction of Abnormal Behaviour

Lotfi et al. in [9] explore the identification and prediction of abnormal behaviour in dementia sufferers. They make use of recurrent Neural Networks to "predict the future values of the activities of each sensor" thus ensuring that pre-emptive care can be taken for predicted anomalous behaviours. With respect to AD, prediction of anomalous behaviour is crucial for ensuring the safety of SH occupants whatever their age or health status. Although prediction is not applied in this thesis, the attributes and relationships introduced in this work can find potential use in existing prediction algorithms and used to fine-tune the outcomes of the algorithms.

2.5.11 Unobtrusive Anomaly Detection in Presence of Elderly in a Smart-home Environment

The work by Novak et al. [44] closely resembles the one in this thesis in that it highlights the need to recognise the use of several selected factors to aid the AD process, factors such as the sensor *location*, its *start time* and the *duration of an activity*. They employed *Self-Organising Maps* (SOM) in their solution for unobtrusively detecting outliers in their motion sensor datasets. They use their solution to detect five (5) types of anomalies:

- Unusually long inactivity period
- User absent when presence is expected
- User present when absence is expected
- Unusually short activity duration
- Changes in patterns of behaviour

Working with the MavHome project dataset (`http://ailab.wsu.edu/mavhome/`), they artificially introduced anomalies to their dataset and, only considering activities that ran for longer than fifteen (15) minutes, their system was able to successfully detect anomalies in seventy-five (75) per cent of the cases considered.

2.5.12 Anomaly Detection in User Daily Patterns in Smart-Home Environment

In [45] the authors make use of Neural Network Self-Organising Maps in the design of a system which observes and learns users' behavioural patterns and detects anomalies from these patterns. The anomalies they focus on are the same as in their previous work [44]: activities occurring at unusual periods, activities occurring for unusually long periods and activities occurring for unusually short periods when they should normally go on for longer. Also, similarly to their previous work, they make use of Neural Network Self-Organising Maps and they claim that their work is not as limited as those that focus on detecting only single anomalies and the detection of *inactivity*. They tested their work on real-life data from the MavHome (`http://ailab.wsu.edu/mavhome/`) project as well as on synthetic data which they generated using an ADL dataset generator. They manually added anomalies to the real-life dataset and, by generating anomalous reference points, were able to set their data generator to generate scenarios that included anomalies as part of the daily activities. They simulated ten (10) daily scenarios.

The results they obtained demonstrate that their system was able to detect ninety-six (96) per cent of the manually induced anomalies in the real dataset. Similarly to the work in this thesis, they focus on AD in a single-occupant home in which there are no pets. However, their simulation is based on a data generator which inserts anomalies based on data and not based on the context of activities and events which is something that was taken into consideration in this work Chapters 5 and 6.

2.5.13 Information Technology Supporting Daily Activities of seniors

The authors develop a tool (ZoneSURE) which can be used to detect anomalous presence or absence [46]. ZoneSURE operates by detecting if a user is in a particular location during a particular time interval as expected or if a user is indeed present in a certain location during a certain time interval when no one should be. Their work closely resembles the one in this thesis in its aim i.e. the detection of *presence* however, it does not acknowledge the fact that motion detection nodes being active do not always indicate presence but may be as a residue of a former presence in a particular location. They thus do not investigate the identification of multiple "presences" within the SH in non-proximal zones. Information on proximals is available in Section 4.2.1.3.

2.5.14 Anomaly-based Data Mining for Intrusions (ADMIT)

For this work, the authors applied data mining to design a real-time Intrusion Detection system which identifies legitimate users of a "computer terminal" from a non-legitimate user [47]. Their work is relevant to this thesis because it recognises that sensor "firings" or activations do not happen in a binary way. For instance, they explain that "some sensors will keep firing even though the same event triggered them." and they use as an example, sensors in a KITCHEN firing continuously whilst the user is moving around in the KITCHEN. These firings cannot be recorded as multiple *Cooking* or *MealPreparation* events (example events that may happen in a Kitchen) otherwise, an AD system set to identify individual sensor firings as separate events may label any and all occurrences after the first firing as suspicious and, potentially, anomalous firings thus leading to false positives.

2.5.15 Ontology-based Activity Recognition in Intelligent Pervasive Environments

Chen and Nugent in their work [48] introduce the use of Ontological Modelling, Representation and Reasoning to the problem of Activity Recognition. According to the authors, their work "exploits logical semantic reasoning for the purposes of activity recognition". They extend their work to the realm of continuous activity recognition and their approach - using ontological modelling tools and reasoner - yielded useful results according to them. One challenge to their work is the assumption that the activation of a sensor implies that the sensor is actually being used for a certain purpose. In reality a person may pick up a particular node (for example a cup) as part of the event MakeTea. They may then change their minds (for any number of reasons), drop the first cup and pick up a different cup and use the second one. In addition, users do not always stick to a particular pattern i.e a user may not always make coffee at a particular time. Thirdly, That a coffee pot was touched at a particular time of day that fits a pattern (e.g. lift coffee pot at 8.30 a.m.) does not imply it was used to make coffee even if the activity fits the event. That same activity might fit different event too. Just because a user picks up a coffee tin does not imply that the user intends to make coffee; a person sometimes picks up a coffee tin in the morning to clean their kitchen cabinet.

Therefore, even though their system is applicable in situations where highly regular ADL patterns are followed (such as in patient healthcare management), the system would not scale well where *unpredictability* is a factor to be contended with. Therefore, the context in which the action is carried about is also important in addition to the individual action.

The basis of Ontological Activity Modelling is that the **domain knowledge** of activities being performed in a particular space (e.g. hospitals, shipyard loading docks or even smart homes) is used to identify the different ways in which the contextual information come together to form different patterns for performing certain activities. Domain knowledge is very important for building a set of potential ways in which activities can be performed so as to identify the different expected activities that are part of the normal pattern and those which are not. After the activities have been identified, when a user attempts to carry out an activity which is not on the list and does not meet set constraints, it is flagged as an anomalous action.

This thesis supports existing AD approaches by highlighting the fact that nodes, zones and home occupants in SH ecosystems form relationships and that these relationships can be exploited to build baseline security models such that deviations from these secure models are identifiable as anomalies.

2.6 Chapter Summary

Datasets obtained from Smart Home (SH) will be multivariate in nature because of the variety of nodes and thus, sources of data which are found in SH. This Chapter provided an overview of current techniques used for AD from multivariate datasets. The techniques discussed can be broadly put in two groups: supervised and non-supervised methods. These methods apply different classification and clustering approaches and algorithms such as the KNN and SVM methods.

Supervised methods involve making use of explicitly labelled data to train a model. The output of this training is a classifier. This classifier can then be used to detect classes (or labels) of fresh (unlabelled) test data. Clustering techniques such as the K Nearest Neighbour (KNN) make use of the closeness between data points in a space to identify the groupings that data points best fit into.

Current AD methods in general detect anomalies by learning a *base normal* which subsequent data is compared against so that any variation is flagged as an anomaly. There are various variations to how each method reaches this goal however the general premise is the same (apart from with the Ontological Modelling method which makes use of patterns and constraints): the user's patterns are learned and used to build a model through which fresh user behavioural data is analysed. Any variations from the normal threshold are flagged as anomalies. Some of the existing methods assume that humans are binary systems and that it is enough to harvest information about them whilst ignoring information that pertain to the nodes that they interact with in and around their homes. Some of the reviewed methods also do not capture the interesting reactions of certain nodes to stimuli not directly applied to them such that their reactions appear anomalous.

This work proposes that, as a support for AD, the relationships formed by nodes and zones in SH on the basis of their attributes can be harvested, that rules can be formed to constrain these relationships and that these relationships, alongside user behavioural patterns - in the form of well-known, pre-determined events and activities - can be used to form baseline security settings for the individual SH such that any deviations from this baseline can be investigated and a decision reached about whether the deviation is an attack or an error, or if the anomaly is a False Positive, a False Negative, a True Positive or a True Negative. Using the range of anomalies allows for the reduction of an anomaly report to not always call for external parties especially if it is found not to be an attack i.e. a True Positive. More details on True Positives can be found in Section 5.2 in Chapter 5.

Chapter 3

Internet of Things Digital Forensics Framework (IDFF)

3.1 Overview

This Chapter presents an overview of the preliminary research that was carried out as part of the effort towards answering the research question. These foundational aspects contributed to the overall understanding of the research problem and contributed key aspects of the solution. The Chapter begins by proposing a definition for the word *Things*. This definition was proposed in order to provide an insight into what *nodes* in the Internet of Things (IoT) are, for the purposes of Digital Forensics (DF) investigations. In this regard, the words *Thing* and *Things* are used in this Chapter but they are interchangeable with the words *node* and *nodes.*

A discussion that addresses ‘acceptable’ behaviour (Code of Conduct) within the IoT as a cyber-physical environment leads on to the introduction of The Thing Commandments which, from a philosophical viewpoint, are a set of guiding principles and policies that can be applied by all the stakeholders involved in the IoT during the IoT’s introduction, deployment and thereafter. A discussion follows on the potential responsibility challenges that will exist in smart, autonomous ecosystems and a proposal for Responsibility Modelling to be applied within the IoT context as a way of mitigating these challenges is introduced. The Chapter concludes with the introduction of the IoT Digital Forensics Framework (IDFF), the IoT Incident Response (IR) Plan, and an example analysis of a hypothetical Incidence Response Scenario using a combination of the IDFF and the Incident Response plan.

The publications on which the chapter is based are:

Publication 1: Oriwoh, E., Sant P., and Epiphaniou, G. Guidelines for Internet of Things Deployment Approaches – The Thing Commandments. Procedia Computer Science, 21(0):122-131, 2013. doi: 10.1016/j.procs.2013.09.018 Conference: The 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2013) and the 3rd International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH), Volume: 21.

Publication 2: Oriwoh, E. and Sant P. The Forensics Edge Management System: A Concept and Design. In Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC), pages 544-550, 2013.

Publication 3: Oriwoh, E., Jazani D., Epiphaniou, G., and Sant P. Internet of Things Forensics: Challenges and Approaches. In Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013, 9th International Conference, Conference on, pages 608-615, 2013.

Publication 4: Oriwoh, E. and Williams, G. Internet of Things: The Argument for Smart Forensics. In Handbook of Research on Digital Crime, Cyberspace Security, and Information Assurance, pages 407-423, 2014, publisher: IGI Global.

Publication 5: Oriwoh, E., Sant P., Epiphaniou, G., French T., Maple C. (2013) Do Not Pass The Buck: the Need for Responsibility Modelling in the Internet of Things, Procs. of the International Conference on ICT LAW 2013 (Information and Communication Technology Law, Protection and Access Rights), Porto, Portugal.

Publication 6: Oriwoh, E. and Conrad, M. 'Things' in the Internet of Things: Towards a Definition. volume 4(1), pages 1-5, 2015. doi: 10.5923/j.ijit.20150401.01.

Publication 7: Oriwoh, E., al-Khateeb, H. M. (2015) 'Internet-Of-Things: Towards a Forensic Methodology', Digital Forensics Magazine, 2015(22): 34-37.

3.2 'Things': A definition

As part of carrying out research into the IoT, Smart Homes and security in both these areas, it was deemed necessary to proffer a generalised definition of the word 'Things' in the context of the Internet of Things (IoT). A brief summary of the related publication ([49]) that discusses this definition is presented here:

For anything to be classed as a ‘Thing’ in the IoT sense of the word, the following criteria are suggested:

- It serves a purpose;
- It can be interconnected as part of a network even though this might not always be. All Things possess the facility for interconnection either using technology (e.g. 802.11 or Ethernet connectivity) or via a natural method e.g. verbal, physical or other human communication; Things can also be fully functional, standalone nodes.
- It can have a physical or logical form;
- It is traceable physically or logically e.g. by eyesight or GPS location tags;
- It can be communicated with or it can communicate or both;
- It can be interfaced with;
- It can be living or non-living;
- It can be identified e.g. using a physical and/or logical identification marker;
- It has capacity for autonomous operation;
- It is tangible or intangible;
- It can be naturally autonomous (e.g. humans), enabled to be autonomous (e.g. self-driven vehicles) or even non-autonomous (e.g. software code).

Drawing from these discussions and criteria, ‘Things’ are described as ***‘anything at all, depending on requirements’***. An example of a Thing is a self-driving, autonomous vehicle.

3.3 Code of Conduct within the IoT

The IoT introduces a unique dimension to the relationships between humans and the (smart) objects that belong to them and/or which help them fulfil a direct or indirect objective. With extended and continuous interactions with these Smart Things, this relationship can be expected to gradually become one of reliance and maybe eventually one of trust. However, even though end users may increasingly rely on smart Things for the achievement of goal states, there should be a clear recognition by stakeholders (owners, users, designers, vendors, etc.) of where responsibility for Things, and actions by Things, should lie. This Responsibility question has recently started gaining some attention as part of public discourses. In early March 2015 the UK Parliamentary Transport Committee and the House of Lords raised the subject as one that needs to be addressed [50].

There are numerous beneficial applications of interconnecting objects in the form of an IoT in areas like emergency management, logistics, and medicine [51–53]. However, on-going research has exposed some privacy risks thus leading to some questions. Examples of these challenges include: how the owners of Things can accurately be identified? How to know when a Thing (as opposed to a person) has done something wrong? Who is to blame for a Thing doing something wrong e.g. for committing a crime in the form of a mistake or an anomalous activity? What should a society's expectations be of its government in relation to Things like sensors being deployed publicly or the expectations of individuals of other individuals? Should neighbours always inform each other about the presence of all sensors or of certain types of sensors outside their homes? When is it permissible not to inform anyone about the presence of sensors? Lastly, who is supposed to have control and what is the acceptable amount of control various stakeholders should (or should not) have over certain Things?

Before introducing the proposed Thing Commandments [54], a number of challenges, issues and concerns in the IoT domain that necessitate a conditioned approach to rights and responsibilities within the IoT are discussed in greater depth.

3.3.1 Legal

Novel attacks and cyber-crimes typically emerge with the development of new technologies. This can also be expected with the IoT [55]. The Law, for its part, tends to lag behind these 'novel' crimes; indeed according to [56], by its very nature, Criminal law lags behind technology and this legal time lag is inevitable because of the inability of state and federal statutes to keep up with Internet and technology developments. Also, Rolf H Weber in [57] and [52] argues that there is a need for governance and legal oversight over how the IoT operates even while it is in its early (formative) stages. As identified by [58], because of the far-reaching applications of the IoT in national and private systems, the legal and policy discussions around the IoT should be dealt with as *important for its development.* It is therefore important that the legal arm of society is prepared - and even contributes - to the development of the IoT by setting guiding principles that stakeholders would be expected to adhere to. Governments and lawmakers should be ready and willing to modify existing laws or create new ones, in order to safeguard citizens from the dangers of mismanaged or maliciously modified systems.

For instance, in order to deal with some aspects of computer crime, the UK government deemed it necessary to introduce the Computer Misuse Act 1990 [59] although certain other cyber-crime-related charges are still being brought under (previously existing) laws such as the Theft Act 1968 [60]. This approach can serve as a good precedent for dealing with IoT-related crime - old laws that are applicable can be retained and new ones or, at the very least, new guiding principles and policies such as the ones proposed by the United States National Highway Traffic Safety Administration (NHTSA) Policy to govern the use of self-driving cars - should be enacted.

Some existing UK-based laws that may apply within the cyber-physical environment of the SH include

- Theft Act 1968 This law is applicable to the authorised or unlawful removal of someone's property. With respect to the IoT as a Cyber-physical system, property includes all *cyber* and *physical* property. `www.legislation.gov.uk/ukpga/1968/60/section/9`
- Computer Misuse Act 1990, 2006 A relatively new law under which several prosecutions have already been successfully brought since it was enacted. `www.legislation.gov.uk/ukpga/1990/18/contents`
- Communications Act 2003 This act is used to charge people who piggy-back onto the wireless netwroks of others typically because the networks in question are relatively insecure. `www.legislation.gov.uk/ukpga/2003/21/contents`
- Data Protection Act 1998 `www.legislation.gov.uk/ukpga/1998/29/contents`
- Regulation of investigatory Powers Act 2000 `www.legislation.gov.uk/ukpga/2000/23/contents`

3.3.2 Accountability, Liability, Rights and Responsibility

The number of devices connected to the Internet is expected to reach 50 billion by 2020 [61]. This proliferation of connected devices may lead to management and oversight create some control concerns for governments and personal privacy concerns for citizens. Introducing discussions on rights and responsibilities at the development phase of the IoT is pertinent and is what the work in this thesis aims to do. As an example of proactive responsibility management, governments might be expected to indicate clearly to their citizens (e.g. through the use of publicly-placed sign posts) where smart sensors have been deployed in public even if these are deployed for the benefit of the public have deployed tags in public places.

3.3.3 Trust

Some concerns about trust include those expressed in [62] in which the authors raise concerns that deployed RFID systems could encourage widespread surveillance of members of the public without their knowledge or consent. They also argue that data produced might be re-used for a second purpose, again without the consent of those whose data was collected.

3.3.4 Regulation and Control

There are a number of questions around regulation and control over the IoT. Questions like who, if any **one** body should control the IoT? Should it be self-regulated or government controlled? *Should* there be a body governing the IoT? For management and marketing purposes, it may be deemed necessary to introduce a system for storing records of all tagged (or IoT-enabled) objects. This is similar to the system where customers who purchase contract (mobile) phones and lines are registered on to the provider's database. If a system like this is introduced and customers' details are collected as part of the purchase process, then access to these records would have to be strictly controlled.

3.3.5 Privacy

There are already concerns about the deployment of devices that can be communicated with over IoT-enabling technologies like RFID.

Consider the new e-Passports that the United States started issuing to its citizens since August 2007. This passport has an integrated RFID chip embedded into the back page of the passport; the chip digitally holds relevant information about the passport assignee for security reasons [63]. When grouped according to their source of power, RFID tags can be active, semi-active, or passive and they can be awakened and read by RF readers at adequate distances from them - some from distances of several meters [62]. However, the research to determine the best way to secure this communication link (i.e. to avoid indiscriminate awakening and reading of tag data, for instance in sensitive nodes such as the passport described above, by just any reader) is still on-going and has not yet produced a widely accepted result. Therefore, although RFID is beneficial since, by design, it enables automatic and time-saving identification, the benefits of being automatically identified may not outweigh the potential privacy and data loss disadvantages for an individual. Also, the time-saving benefits of being able to identify the contents of a traveller's suitcase by scanning and reading all the tags attached to each item - without the requirement of opening the suitcase and physically confirming the person's identity may lead to items with spoofed tags being used to provide false identities for their holders and thus erroneously tagged items could be allowed through security barriers such as border checkpoints.

In addition, [64] highlights "tracking" as a problem with the use of tags arguing that the identifiers provided by tags - which are usually predictable - can make it possible for associations to easily be established between tags and their owners. The same paper argued that privacy is one of the issues that are still to be addressed before a major roll-out of RFID systems.

Ethical questions about how any data produced by smart Things are used, by whom, for how long this data is stored and other Data Protection considerations are another aspect of the IoT which need to be addressed.

These questions highlight a requirement for a set of principles that can guide the deployment and application of Things as they become part of the autonomous, ubiquitous mesh of connected devices used every day in society.

Given the existing issues around the deployment of the IoT, a set of commandments are suggested in Section 3.4 that might address them. Some of these commandments may come across as obvious whilst others may be points of contention and probably even encourage debate around this subject area. These Commandments are published in [54]. They were developed to address the foreseen issues of privacy, ownership and use of smart Things, the dangers posed by malfunctioning autonomous, self-aware systems and the need to be able to identify the owners of Things no matter how autonomous or smart they become.

3.4 The Thing Commandments

The aim of drawing up a set of "Thing Commandments" was to introduce principles that can be applied to the IoT so that even as devices are being designed, manufactured and tagged and/or embedded with identification, communication and sensing capabilities, the long-term recognition of the implication of having intelligent, self-controlling (i.e. autonomous) Things - as opposed to non-autonomous things - is recognised by stakeholders including the manufacturers, consumers and governments. The recognition that objects are becoming connected and interconnected ([65], [66]) has led to the development of the Thing Commandments which should hopefully serve as a starting point and feed into discussions around the privacy and security rights and responsibilities of stakeholders where the IoT and connected things are concerned. If there are no guiding principles, there is a potential for the rights of stakeholders (governments, vendors and individuals) to be stretched too far or, conversely, for their responsibilities to be diminished.

Box 3.1. The Thing Commandments

1. Ownership: Things belong to, and are the responsibility of, their owners, and so too are the actions of Things;
2. Access and Authorisation: Accessing Things without permission is not acceptable;
3. The Co-relationship Principle: All Things that belong to a particular individual should belong to a common network and be easily and uniquely identifiable to each other and by each other;
4. Communication Policies: Secure mechanisms and channels of communication between Things should be established;
5. Ease of Use: Things should be made relatively easy to use and manage;
6. Control: All Things should be controllable by their owners;
7. Identification: It should be possible to identify the owner of a Thing from information available *on* or embedded *in* the Thing;
8. Freedom from Things (the Non-reciprocity Principle): People should be free to achieve goal states without the use of Things and at the same time not loose out on rights, privileges, goods or services as a result;
9. Disabling, destroying or disposing of Things: People are to be free to dispose of their Things when it suits them.

3.4.1 Ownership

This commandment argues that Things belong to, and are the responsibility of, their owners and, by inference, so are the *actions* of Things. Consider as an example a situation where a smart Thing owner, say Osas, did not send a specific request to a Thing to carry out an action, for example, “Open the front door” (a direct action A) or even a derived action e.g. do ‘A’ if ‘B’ happens else do ‘D’ (i.e. based on what it has learned), then Osas cannot assume that any negative action that his thing carries out will be excused without expecting any form of reproach to himself as the owner. This way, if a crime involves a Thing, the owner, identifiable by a label on or, incorporated in the Thing, bears some responsibility for the actions of the Thing.

3.4.2 Access and Authorisation

If anybody or entity accesses or controls any thing without the permission of the owners, such access must be regarded as illegal. Meingast et al. in [67] present an argument on how the new applications of RFID technology in transponders can introduce privacy risks. They contend that the objects that are permanently embedded with transponders could be used by individuals in public places where they may have little or no control over who can access them. The UK Computer Misuse Act (CMA) 1990 was introduced to deal with crimes that involve, among other things, unauthorised access to computer systems. Even if RFID tags and sensors have little computing capability, they are by extension, computing equipment and this law can be extended to include them.

In addition, even if it is possible for Things to be accessed by unauthorised (but somehow authenticated) users, it does not mean that the successful unauthorised access is permitted. An analogous situation exists in the the field of wireless networking where the general advice is that even if a Wireless Access Point is not adequately secured by an owner, a person with the requisite skill who gains access to it in order to use the WiFi (e.g. by piggy-backing) is doing something wrong.

3.4.3 The Co-relationship Principle

All Things that belong to a particular person should belong to a network of their Things and should be easily and uniquely identifiable to and by each other through the use of a mutual authentication system. The aim of this *trust* relationship is to enable security by ensuring that rogue Things attempting to become part of someone's personal network of Things (such as their Personal Area Network) would be denied access because establishing that relationship would require certain criteria to be met and authentication threshold to be crossed. These criteria can be based on a system of Authentication, Authorisation and Accountability (AAA) and may even incorporate existing AAA solutions.

3.4.4 Communication Policies

All nodes should be able to identify trusted *communications* made by other trusted nodes nodes. Although [68] argues conversely that no device (i.e. tagged) should be fully trusted by another in order to avoid a situation where every single device stands as a single point of entry to a network, this thesis argues that this a trust relationship, is essential for things to be able to identify *and verify* the identification [69] of each other. In order for this relationship to be established, a strong but manageable system of authentication and authorisation would have to be put in place. This way only trusted communication would be permitted between things. Therefore, any unauthorised “imposter” Thing that tries to communicate with an established network would be detected and such communication would not be allowed.

3.4.5 Ease of Use

It should be made relatively easy for users to configure Things and add new Things to or remove old Things from their network of things. If a user purchases a new Thing and wants to ensure it can communicate with their existing Things and vice versa, the user should be able to do so easily without the need for any special training or requiring the services of the specialist vendor. Vendors should be required to make Things easy to install. One level that they might aim for may be to make Things that can be installed as *plug and play* devices by consumers. This commandment would ensure that vendors don’t make unreasonable amounts of profit from providing unnecessary Thing-related services to their consumers since some consumers would obviously start out by being “Thing luddites”. This commandment is especially relevant to Thing designers, developers and vendors. It is supported by the assertion in [70] that home-owners must be able to get devices on to their home networks by themselves without any assistance from “installers”. Therefore the back-end and front-end technologies have to be designed with the end users’ access needs and requirements in mind. This can be achieved through collaboration with standards bodies such as the Institute of Electrical and Electronics Engineers (IEEE).

3.4.6 Control

It should be expected that all Things should be under complete or at least partial (shared) control of their owners. To meet this requirement, vendors and developers should ensure that all available technology in relation to Things is designed to be user accessible and user friendly as well as easily understandable. Rigorous tests and consumer surveys must be carried out to ensure that consumers are not just aware that they should be able to control their Things but that this is a requirement for owners of Things.

3.4.7 Identification

The owner must be clearly identifiable by some form of identification on, around or hard coded in the Thing. For instance, Things can be affixed with a sign that states they are the 'Property of the Government of a particular country or military'. Any Thing with no identifiable owner belongs to no one and can legally be confiscated or destroyed. This commandment would prove especially useful for law enforcement agencies since it would most likely be up to them to locate the owners of misplaced Things. This way, hopefully, any crime committed by a Thing can be traced to the owners. There might be the need to detect ownership by scanning a code but that code must be easy to locate by (or accessible to) law enforcement.

3.4.8 Freedom from Things (the Non-reciprocity Principle)

This commandment incorporates and introduces a caveat to the previous commandment. Simply put, it eschews "The Right to Freedom from tags and tagging". The commandment seeks to introduce to the IoT discussion the fact that not everyone will wish to use smart, autonomous Things. This commandment therefore ensures the freedom of individuals and society from tags of every form. It introduces the rights of individuals to refuse all or certain things e.g. implantable tags. RFID tags are being embedded inside pets, exotic animals and people [71, 72] but this thesis posits that this should only be done with the consent of the individual concerned or an appointed guardian, within recognised legal limits.

3.4.9 Disabling, destroying or disposing of Things

This commandment asserts that it must be possible and easy to disable Things even remotely. This commandment also requires that Things should be easy to dispose of and the data within them completely destroyed. Bernard in [73] argues that, as a fundamental right of citizens, the "silence of the chips" must be preserved. His argument supports the position of this commandment.

3.5 Responsibility Modelling (RM) for the Internet of Things

This section discusses the benefits of Responsibility Modelling (RM) for the IoT. It begins by discussing some of the potential responsibility challenges that may arise in mature, smart, autonomous environments such as the IoT.

3.5.1 Responsibility Challenges in Smart, Autonomous, Ecosystems

As part of the Internet of Things (IoT) design and development, ordinary objects are being equipped with the ability to sense and make decisions independently on behalf of their owners and of end-users. In the near future it can be expected that people will increasingly transfer daily decision-making tasks to smart non-human (or autonomous) agents and that Things will take over tasks as required by their owners or operators. Jiang et al. note in [74] that the effect of the IoT on daily life will be "transformational". For instance, parents may transfer the tasks of ensuring their children arrive at their school on time to non-human nodes such as self-driving vehicles that work in cooperation with smart traffic lights as well as other self-driving vehicles. It can be supposed, going forward, that tasks that will be assigned to smart Things will range from simple trivial tasks to those which can be described as being "safety critical" in nature. As a direct consequence of having smart Things take over a variety of mundane as well as safety critical Activities of Daily Living (ADL), people are likely to develop an increased reliance on these Things. One of the aims of the IoT is for it to become a structure for improving people's lives, one in which people find that they can reliably assign tasks to smart Things to take care of satisfactorily. However this reliance upon IoT elements (smart, sometimes autonomous, systems) to mediate transactions and thus achievement of goal states on behalf of their owners must not be construed as wholly handing over *responsibility for the outcomes of the actions* of these systems. It is therefore essential to recognise the limits of the IoT technology so that the transference of skills, obligations and responsibility for accomplishing and completing tasks on to smart Things is not viewed as the transference of responsibility for the *consequences* of their actions as well.

3.5.2 Benefits of Responsibility Models from the perspective of the IoT

In the light of the preceding discussion, it is essential that the question of *responsibility* within the IoT is addressed. According to Ashton [12], humans are intrinsically fallible, having limited accuracy and so introducing a high level of autonomic intelligence to Things can minimize delays and mishaps that occur due to human error. While this argument is not undesirable, it has potential shortcomings that can be highlighted using RM and this section discusses some of the potential benefits of introducing RM into IoT transactions.

3.5.2.1 Overcoming Responsibility Vulnerabilities

Responsibility vulnerabilities occur when agents do not properly discharge their responsibilities for any number of reasons. Six (6) responsibility vulnerabilities are identified and defined by Sommerville [75] (Table 3.1):

Vulnerability Type	Description
Unassigned	A responsibility is not assigned to any agent
Duplicated	More than one agent assumes each holds the same responsibility
Uncommunicated	Agent not aware of assigned responsibility
Mis-assigned	When a lack of tools or skills affects effectiveness
Overloaded	When an agent is given too much to do
Fragile	When no alternative agent is made available to take over if the first one becomes unavailable

Table 3.1: Responsibility vulnerabilities

In an increasingly automated world, the vulnerabilities described have to be guarded against. However, if and when they do occur, the people with whom the consequential responsibility lies must ultimately be identifiable. This identification process can be assisted through the use of consequential responsibility models. Consequential responsibility models are useful for identifying agents in a socio-technical environment that are ultimately responsible for consequences of outcomes.

3.5.2.2 Addressing Legal needs

The IoT, whilst it is in its early stages of development, is already providing beneficial applications in areas of health, transportation and security, among others. However, as with any new technology and new application of existing technology, the IoT introduces dimensions of crime that will undoubtedly provide opportunities to malicious parties to exploit with potentially harmful consequences. The possibility for owners to train smart Things to commit crimes may become an issue for existing legal systems. From the perspective of the United Kingdom, there are existing laws which might be applicable to crimes in the IoT including, amongst others, the Computer Misuse Act (CMA 1990) [59]. Other laws or acts that apply in other countries include the Health Insurance Portability and Accountability Act (HIPAA 1996) which regulates the access and use of individually identifiable health information in the US [76]. However, the notion of a legal system implies a recognition of responsibility. Lock et al. [77] state that the definition of responsibility inherently implies that all actions and systems should "work within the social framework of both legal and domain standards". Without a method of identifying responsible parties when the situation calls for it, there might be issues when it comes to pressing charges and this may lead to erroneous convictions or acquittals. As a result, there is an explicit requirement in the early stages of the IoT development for a method of identifying responsible parties.

In addition, legal systems have historically only had as their focus living and/or deceased human characters and establishments (e.g. companies) because it is to these that the concept of punishment and praise currently has meaning, relevance and effect. Legal systems obviously have no impact on conscience-free, non-human objects whatever the consequences of their actions. Therefore, it is imperative that a method of ascribing responsibility for the consequences of actions of smart Things to people is developed and made available to all the relevant stakeholders in IoT transactions e.g. lawmakers and law enforcement bodies. Having such a method of identifying responsible parties and for apportioning blame, if any situation arises within the IoT domain, the responsible non-humans can be correctly identified and ascribed causal responsibility (what did it?) whilst humans can be ascribed both causal (who did it?) and consequential responsibilities (who is to blame?). This will eliminate the possibility for a human player to pass the blame buck. From the perspective of consumer protection, this model will be useful for end-users since blame for the actions of faulty Things may lie with the manufacturers, vendor, and designers and there may be a need for them to seek redress of some sort.

3.5.2.3 As a discussion support tool

Responsibility Models are designed, not to be static, hard-and-fast, unchangeable tools, but rather as tools to facilitate discussion around subject areas that, although might still be subject to change and development, require some form of structure during their evolution. This is undertaken so that the concerned stakeholders and agents recognise what should be done if anything goes wrong [75]. As previously mentioned, Responsibility Models are useful for the purposes of eliminating any chances for blame to be passed around - a situation which might lead to a lack of resolution of issues when and if something goes wrong because the responsible party is not clearly identifiable. Designers of IoTware can also make use of RM through the discussion it generates to better understand user requirements.

3.6 The Forensics Edge Management System

The Forensics Edge Management System (FEMS) is briefly introduced in this section. More detailed information is available in [78]. The FEM is a system that is designed to provide real-time, automated and autonomous security and Digital Forensics (DF) services within SH environments. The FEMS operation is as illustrated in Figure 3.1. Thresholds are set by the user and continually updated by a SH Management system. If any breach is observed (based on a change in the "Home Happiness" status), the FEMS operation is triggered. It harvests any data acquired within a specified time window before and after the alert. This data is parsed and a decision-making algorithm (the FDMA) performs a DF investigation at the end of which it produces a report about whether the anomalous activity observed was an attack or an error. Example FEMS services include network monitoring, data mining, logging, timeline creation, alert management (incident escalation) and presentation of investigation results and reports in human-readable formats. Chain of custody management is also handled within the FEMS itself through a combination of authentication and time-stamp details i.e. details about ***who*** logged in to the FEMS and ***when*** are combined and used to track who handled the evidence. More details on the FEMS is available in a related publication [78]. The relationship between the IDFF, FEMS and FDMA is illustrated in Figure 3.2.

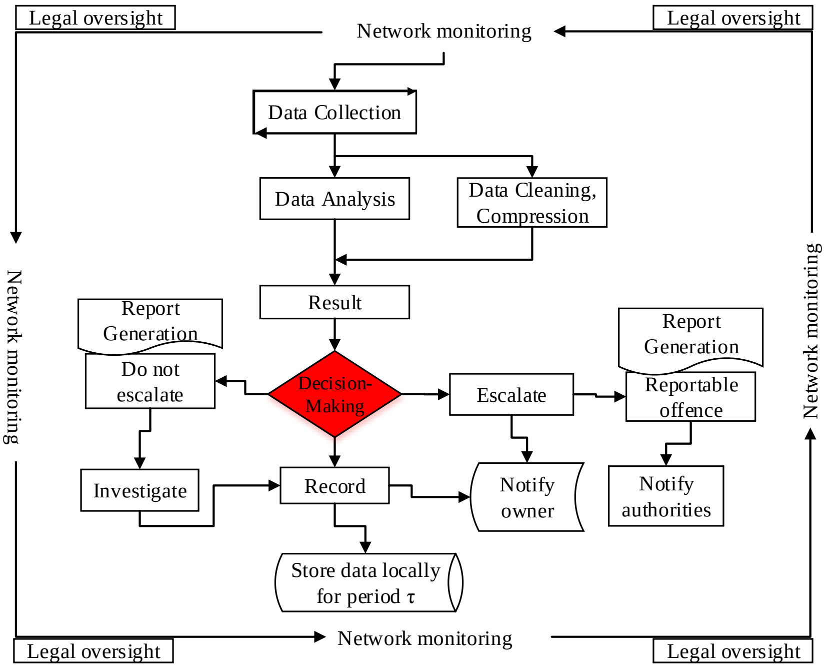

Figure 3.1: The FEMS Operation (summarised)

The FEMS can also be set up to perform integrity self-checks using methods such as hashing. Hashing involves using passing data through a hashing algorithm such as SHA256 which generates a hash of the data. This output is an irreversible value which can be subsequently securely stored. In order to find out if the data has been tampered with, a new hash is generated from the raw data and compared with the old hash and if the two resultant hashes match, then the integrity has been maintained. This can be used to ensure that any evidence collected by the FEMS is not compromised during an investigation. This method of maintaining integrity using hashing is used in digital forensics where investigators store an original copy of a piece of evidence then work on copies which are first hashed and their hashes compared to the hash of the original untouched evidence file to be sure that they are the same as the original and uncompromised.

In addition, delays to the system are avoided by making sure that even though the FEMS monitors the network continually and initially any anomalous pattern is detected in real-time, further detailed analysis of any acquired evidence is carried out offline. This is because a full analysis of captured data online and in real-time may adversely affect the performance of the FEMS as well as create bottlenecks on the overall SH network.

3.7 IoT Digital Forensics Framework (IDFF)

3.7.1 Overview

The IDFF framework (originally introduced in [78] and extensively modified in this thesis) is a Framework that is applied as part of the general incident Response solution operations for SH. The IDFF was drawn up from a review of existing DF frameworks and methodologies [79] and from input (gained through interviews) from academics and former DF field and industry experts with relevant, professional experience and qualifications. The interviewees from whom feedback was obtained included:

- Two (2) Professors, one of whom had extensive past professional field experience (United States);
- Six (6) Doctors i.e. PhD holders - three (3) UK, two (2) USA, one (1) Pakistan;
- One (1) co-author of a widely-used. widely-consulted, edited DF text book who also had experience of DF frameworks development which are actually being used.

Their input was obtained through 30-40 minute, qualitative, semi-structured, interviews conducted over the Internet using Skype$^{\text{TM}}$.

According to the Sample Dissertation Methodology Article available at [80], "A semi-structured interview is a qualitative interview that is defined by a pre-set question guide." Also according to [80], as part of deductive Qualitative Data analysis of

Research Findings, a pattern matching procedure "involves the development of an analytical framework, utilizing existing theory, and then testing the adequacies of the framework as a means of explaining the findings (Saunders et al, 2007). In the instance where a pattern is found as initially predicted, it would be evidence that suggests that there is indeed an explanation for findings." The interviews in this thesis were conducted after a preliminary framework had been drawn up and the feedback received was used to adjust and update the framework where necessary thus following the recommendation in [80].

The (theoretical) evaluation process of the framework involved drawing up an initial version of the IDFF framework, sending this out to the identified and selected group of DF professionals and academics and inviting them to give their feedback on the framework during an interview session. The interview questions are included in Appendix B. The interview questions were asked with the following rationale:

- to identify the interviewees' knowledge of the IoT and their overall Digital Forensics (DF) experience and expertise;
- to find out if there is a need for a DF framework for the IoT;
- to obtain feedback from them on the draft IDFF which was sent to them prior to the interview date, to gain some insight from real-world DF academics and (former) investigators into what an IDFF should contain and if they thought the IDFF as presented did not meet those ideals.

The Interview questions had several objectives and these objectives addressed several key factors: the expertise of the interviewees in IoT and DF fields, their views on the future of the DF as applicable to atypical nodes within the IoT and the subject of data explosion due to the interconnection of more nodes as part of the IoT and how this will affect DF going forward.

The interviewees' feedback overwhelmingly supported the argument that a good structured Forensics approach is indeed necessary for forensics in the IoT.

Drawing from the input from the interviews - and this thesis author's own domain knowledge - the following recommendations were made for strengthening the IDFF framework:

- The expansion of the framework beyond its initial remit;
- The inclusion of legal aspects in the discussion;

- Match the outcomes of the "Presentation" and "Report Generation" stages with applicable laws;
- Apply the framework to a theoretical case to demonstrate its suitability;
- Present the framework in the form of a flowchart to aid easy understanding;
- Inclusion of an "Analysis" phase during which logs that are not useful as part of the overall investigation may be identified and discarded;
- Identify clearly **who** trains the FEMS e.g. the home occupant, their family, friends, carers, cyber-physical security product vendors and installers, etc.

The next section discusses the Framework in greater detail.

3.8 IDFF Framework Breakdown

3.8.1 Phase 1: Preparation Phase (Setting up the Smart Home)

3.8.1.1 Stage 1: Physical Security

In order to adequately prepare for cyber-physical forensics, the *physical* security of the home has to be considered as part of the Forensics preparedness process. Some solutions that can be used to support physical security around SH include Closed Circuit Television (CCTV); security guard dogs; physical human and node identification methodologies (to avoid rogue nodes being introduced to the SH environment); SH physical boundary and perimeters protection e.g through fences; water leak/spill detectors; total number of nodes threshold to prevent piggy-backing on to electricity supply; extinguishers (foam, water, etc.); the average number of human home occupants per period or per day to avoid physical intrusions and to provide alerts about unauthorised congregations e.g. parties in the home; physical cable cut detectors; Wi-Fi piggy-backing; safes, physical intrusion trip-wires; keyed/coded locks and other window and door security solutions; protected and limited physical network access e.g. limited access to Ethernet ports; quick and sound waste disposal policies and procedures to prevent *dumpster diving*: this may involve shredding as necessary, training and information to home occupants about Social Engineering practices; monitoring Heating, Ventilation and Air-conditioning (HVAC) systems; security at air vents to prevent human entry or harmful gas and other chemicals being passed in through vents; secure, sturdy doors; biometrics security solutions and so on. Even though SH would not necessarily be expected to have all the same security features as, for instance, public places, having a minimal set of solutions will reduce the threat landscape. However, before any security measures are put in place the home occupants' input must be sought because of the cost and the relevance of the solutions; providing security dogs to a person who does not like dogs will, for example, be counter-productive. A useful way to determine what type and level of security to implement in each SH is to find out the requirements of the SH occupants through the use of approaches such as Needs and Wants assessments where the users identify in detail what is important to them and how tightly security measures should be tweaked or otherwise.

3.8.1.2 Stage 2: SH Network Preparation

During this stage, all the devices in the home that need to be networked are identified and then connected (i.e. networked). This stage completes the initial setting up of the *base* SH network. The home network may include nodes such as fridges, televisions, thermostats, cars, and IP-based routers and switches. Network testing and troubleshooting should be carried out during or at the end of this stage to make sure the network is functional. In addition, during this stage the user must install and configure any required Intrusion Detection and Prevention Systems (IDPS). The registration of nodes in and around the home can be achieved by scanning their RFID tags or other types of tags (bar codes or QR codes) where available, by manual registration and even self-registration by the nodes themselves. The identifiers of the nodes can be specific or generic. Example node identifiers include `BedroomFan`, `KitchenFridge`, `Node 001`, or simply, `Node A`. As part of this stage, the home occupant should identify the different zones Z in the home and the nodes η in each zone as well as the **number** of zones and nodes in each zone (where *zones* are rooms and other spaces in the Smart Home). An example zone identifier is BigBedroom. After the nodes and zones have been identified, any networks that individual nodes belong to should also be identified. As an example, the TV, games console may belong to a Living room entertainment sub-network. As part of the SH network preparation process, the following information would then be required:

- The attributes (e.g. Type γ and mode μ) of the nodes;
- All the Relationships (Peripherals, Proximals, etc.) between the nodes and zones;
- All External Access Nodes for the purposes of the Mutual Exclusivity Rule;
- Any initial thresholds as applicable. An example threshold can be *"Keep the Living Room temperature at a minimum of 30 degrees during the winter months of the year"*, and so on;

- All trusted firmware and software update sources e.g. websites;
- All trusted vendors, resources- and service-providers, suppliers and stores. This list can be, for instance as shown in Table 3.2;

Node	**Resource**	**Supplier**
Fridge	Groceries	Tesco
TV	Films	NetFlix
Car	Fuel	Local mechanic

Table 3.2: An Example Smart Home Supplier/Vendor Table

The physical and logical security preparation phases help with the *prevention* of attacks. However the measures put in place during these phases should also help make *detection* easier.

3.8.1.3 Stage 3: Forensics Preparation

During this phase, all the devices that may have digital evidentiary value are identified. If an evidence ranking method is available, these sources may be ranked and listed according to *order of relevance or importance* to investigations with regards to the evidence they hold. Devices that may hold only physical evidence should also identified during this stage. Lastly, devices that may hold cyber-physical (digital *and* physical) evidence should be identified. A data aggregator can be used to collect logs and data from the different nodes that are identified during this stage as being potentially useful for forensics investigations.

Apart from the physical nodes and the evidence they provide, the approach in this work proposes the use of node and zone attributes, relationships and activities and events constraints as aids to AD processes an systems. Therefore at this stage, the following should be identified:

- Potential activities that make up expected Activities of Daily living (ADL) or events;
- Core events per home occupant and core activities for each event;
- All the edge nodes[1], if any;

[1] For the sake of simplicity and streamlining the investigation process, it would be better if the nodes are arranged in such a way as to avoid having any edge nodes

- Trusted (third-party) cyber-physical security and forensics services company(ies);
- Logging frequency and logging amount or level. Example logging rules are given in Table 3.3);
- Storage location(s) for logs e.g. local or remote or both.

Log data	**Log duration**	**Log location**	**Storage Location**
Everything	One (1) month	Local Server	Living Room
IP addresses only	Two (2) weeks	Remote Database	Cloud-based Server
Everything except Motion sensor data	Daily	Home PC	BigBedroom

Table 3.3: Example Logging levels

Part of the preparation for forensics involves the identification of critical infrastructure (physical and logical) in the home. Critical infrastructure includes those that, for example, must never be turned off because turning them off will potentially endanger the SH or its occupant(s). Other such infrastructure includes nodes that hold critical or private data. If a ranking methodology is available, one may be applied at this stage to identify the nodes that are non-critical and those that are. This phase can be seen as the phase during which steps are taken to prepare the SH for Triage.

3.8.1.4 Stage 4: Training and Adjustment

The first step in this stage is to, as much as possible, maintain the Home status at “Happy” (i.e. normal) so that the FEMS learns what the “Happy Home Network” attributes and values are. During this stage, the SH system can be explicitly trained on what is normal **by the user** e.g. thermostat values and preferred wake-up times can be keyed in into the Home Management System (HMS) or even directly into the FEMS. After this point, the home, through the HMS or FEMS can subsequently **learn and update its records** on what normal is. The meaning of what is normal may change because of changes in user behaviour. The user may also choose to manually modify what the system has learned. These must be updated as soon as possible so they are not erroneously detected as anomalous by the FEMS.

Further details on Phase 1 is available in Table 3.4.

3.8.2 Phase 2: Detection Phase

This phase involves the use of the combination of any available home network monitoring tools (Firewalls, Intrusion Detection and Intrusion Prevention Systems) in tandem with the trained FEMS and its decision-making module (the FDMA) to detect any changes from normal to abnormal behaviour in the homes "Happiness" status. The FEMS, which by this stage should have been configured with baseline operating values of different nodes in the Home, would detect any significant changes from the learned baselines, within a set of constraints such as *time of day*, *location of event*, etc., . The Anomaly Detection takes place as follows. After the SH has been set up:

- As each event happens, the FEMS checks for the Activity Sequence (ASeq) followed, the core activities (as defined in [10]), the order of activities, and the normality of the attributes and relationships;
- An anomaly is recorded if a wrong sequence of activities is followed, the activities are in the wrong order, the core activity(-ies) is missing or if abnormal relationships occur;
- Alerts are sent as necessary.

Essentially, the FEMS attempts to answer the question **is the Home at a "Happy" State** i.e. are all attributes normal? If the answer to this question is no, it implies there is an anomaly and that it may be necessary to investigate. In order to carry out an investigation, relevant logs or data must be acquired. This can be achieved by identifying a start time of when the incident was detected and either tagging the data for later or immediately capturing the logs from that point forward until an identified stop time. Tagging the logs for capturing later can help reduce disruptions to the network but may however allow ongoing attacks to continue. When the log is acquired it can then be analysed by the FDMA for evidence.

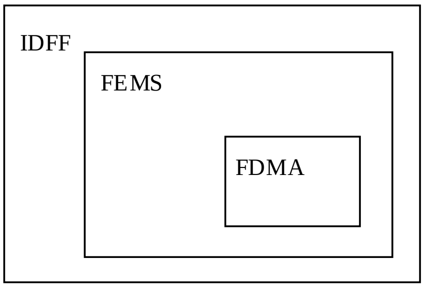

Figure 3.2: Relationship between the IDFF, FEMS and FDMA

3.8.3 Phase 3: Decision-making Phase

The Forensics Decision-Making Algorithm (FDMA) begins its operation in this phase. The acquired logs are parsed following the methodology as shown in Figure 1.2. If the anomaly is recognised as being within the "normal" threshold i.e. False Negative, no action is taken. However, if a higher threshold is reached, the *Investigation Phase* is called *and* the user is alerted. Network monitoring continues throughout the lifetime of the network and is carried out using any available firewall, IDPS, the FEMS, or any other available network monitoring tool as selected by the home occupant. If an even higher threshold is reached, the situation is contained and an alert is sent to previously identified external cyber-physical DF parties as well as the user. These rankings are similar to those discussed in [7]. A high ranking indicates that it has been established that what is happening is an attack. The detected anomalies are divided into different types as one of the four types of anomalies: True Positive (TP), True Negative (TN), False Positive (FP) or False Negative (FN) (see the Anomaly Rankings Text Box in Chapter 5 for information on Anomaly Ranking).

3.8.4 Phase 4: Investigation Phase

This phase involves the identification of the actual causes of the alarm/event triggers detected in Phase 2: Detection Phase. During this phase, the sources of the attacks are identified. Causes may be nodes such as `Kettles` or `Windows`.

3.8.5 Phase 5: Escalation

If the decision reached after Phase 4 (the Investigation Phase) is that the attention of external parties such as an external Digital Forensics (DF) unit is required, the FEMS may escalate the investigation. This escalation can be achieved using text or email alert messages or via remote alarms at the remote site of the third party security and home monitoring company.

3.8.6 Phase 6: Presentation Phase

At the conclusion of an investigation, the results obtained are prepared into human-readable form. This report may be made available to the home occupant immediately after the investigation (e.g. in the form of a text message) or stored and presented to them later. Depending on the outcome of the investigation, the same report can also be sent to third party security companies. The report may be in the form of a summarised message (text or email) or as an automated print-out from an available local printer depending on the user's (preconfigured) requirements. The output that is presented to the user is an explanation, briefly, of what the investigation uncovered e.g. "a presence was detected in the house while the user was away". The report which is generated may have the following fields: Date, Timestamp, Log Duration involved, Location of logs, Decision, Outcome of investigation, Action(s) Taken, Current state of home, Source of anomaly (if any), people involved (if any), End of Investigation Timestamp, End of Report: Date, Time, Signature, Case Officer or Officer-in-Charge. In addition, for preserving privacy, the data is stored locally with access granted to only a select few. An example report is shown in Table 3.8.

3.8.7 Phase 7: Storage and/or Disposal

The logs that were analysed as part of an investigation are stored for a pre-defied period of time after which they can be disposed of. How long they are stored for, where they are stored and how they are disposed of are decided upon based on preconfigured settings by the user. *Restoration and Recovery* of the home to its "Happy" state takes place as part of rounding up the investigation.

Phase	***Stage***	***Steps***	***Notes***	***Output***
1 Preparation	Physical Preparation	i. Identify home perimeter. ii. Install physical security measures. iii. Test physical security measures. v. Zone and node identification.	Registration of all nodes - old and new - in a database of Things; (process continues throughout network lifetime)	A List of zones, nodes. A physically secured home
	Network Preparation	i. Set up home network. ii. Connect the FEMS to the network. iii. Test network connectivity. iv. Configure FEMS. v. Node Registration.	Identification and Registration of all nodes - old and new - in a database of Things; (continues throughout network lifetime)	(Updated) Register of nodes and zones

	Preparation for Forensics	i. Set up logging and log expiration period
		ii. Determine attributes thresholds.
		iii. Define and set threshold for triggering alarms.
		iv. Define and set threshold for escalation.
		v. Identify Home Perimeter and zones.
		Thresholds that are defined can be, for example:
		A: normal network health,
		B: attack noticed and being investigated and
		C: attack beyond capabilities of the FEMS;
			Example threshold attributes are location in SH, temperature, humidity level.
			Threshold values are continually revised and updated according to the users' requirements.
			This should ideally continue as long as the owner wants their devices to be IoT-connected or connected to any network.
				i. Established logging thresholds;
				ii. Incident Response (IR) Plan;
				iii. Trigger levels.

	Training and Adjustment	i. Build the balanced “Happy[2] Network” status ii. Learn ii. Detect changes and learn what is accepted and what is abnormal based on user actions, reactions, input or change in behaviour or requirements.	Continue to learn user preferences This continues throughout the FEMS operational lifetime.	(Updated) trigger levels and SH “Happy” status attributes.
2 Detection	Detection and Acquisition	i. Monitor Home attributes for changes; ii. If yes, trigger FEMS forensic operation.	Attributes include time of day, node ID, zone, etc.	Captured (or tagged logs) with anomalous attributes.
3 Decision-making	The FDMA operation begins	i. Parse captured logs. ii. Unzip, decompress, etc. and parse the logs for evidentiary information. i. If threshold A reached, do nothing; ii. If threshold B reached, investigate, alert user; iii. If threshold C reached, continue network monitoring, contain situation, alert external Forensics Team and user.	Investigation of acquired logs can take place offline or in real-time	Decision i.e. A, B or C.

4 Investigation	Identification of causes and sources of event trigger e.g. kettle, dog, etc.	See Figure 1.2	Takes place, offline using the FDMA algorithm;
			Collect available data for investigation;
			continue network monitoring;
			perform investigation offline.
				i. Relevant evidence;
				ii. Source of attack;
				iii. Nature of attack.

5 Escalation	Escalation and Notification	i. If thought necessary during the investigation an alarm can be sent to an external service provider in addition to a warning message ii. Depending on the outcome, a message is sent to one of two agents.	This is done if the investigation is proving to be beyond the capabilities of the FEMS.	i. Text message and/or email sent to: a. user; b. external digital forensics service providers.
6 Presentation	Report generation	i. Generate a report based on the outcomes of Phases 2 - 5	Outcome of investigation and evidence can be made available to the end user on their return to the SH. Also, it is ready to be made available if queried locally or remotely by the end user or a trusted third party.	Forensics reports on i. Status of the Home ii. Outcome of the investigation.
7 Storage and/or disposal	Local and/or remote logs storage.	i. Logs stored for pre-defined period. ii. Log disposal after expiration period.	Pre-defined storage period.	Set of logs with respective expiration periods., Revised attribute values for “Happy” posture of the SH network.

Table 3.4: Internet of Things Digital Forensics Framework (IDFF)

3.9 IoT Incident Response Plan

In the context of this thesis, the Incident Response (IR) Plan as presented describes the stages and steps that can be followed by Incident Response teams and DF forensics investigators in the event of anomalous security incidents in Smart Homes. The IR Plan may be put into action if the decision taken at the Escalation Phase (Phase 5) of the IDFF is that an external party is required to take over the investigation from the FEMS.

According to the National Institute of Standards and Technology (NIST) *Guide to Integrating Forensics Techniques to Incident Response*, there are four basic forensics phases: *Collection*, *Examination*, *Analysis* and *Reporting*. The IoT Incident Response framework proposed in this thesis is similar to the NIST IR Plan. Figure 3.3 shows the mapping between the NIST and the IoT IR steps. This mapping was done just to show how the IoT Incident Response Plan compares to a well-known, standardised IR Plan such as the one proposed by NIST. The phases of the IoT IR Plan are ***Preparation & Preliminary Information Gathering***, ***Acquisition***, ***Investigation and Analysis*** and ***Reporting & Storage*** [79]. During the *Preparation & Preliminary Information Gathering* phase the investigators gather information from the person who made the report and from any other witnessing parties. They also perform a reconnaissance of the physical and logical SH set-up in order to gain an idea of what the potential sources of evidence are (in a generic sense). If list of evidence nodes are available from the home occupant, these can be consulted, with a certain degree of caution, as well. Evidence is acquired from relevant evidence sources or the next best things (see [81] for more details on IoT forensics and the use of the Next Best Thing or NBT) during the *Acquisition* stage. During the *Investigation and Analysis* phase the acquired evidence is analysed and a report is prepared during the *Reporting* phase. Also at this stage, if necessary, the evidence that was analysed is stored securely *Storage phase.* The SH is *Restored* to a Happy state as much as possible after the investigation so that the Home occupant may continue to use it (that is, assuming the investigation does not find that they themselves caused the problem). Chain of custody is maintained through signing and secure storage and handover of evidence between investigators in the IR Team.

NIST is a widely recognised Standards Agency. It is part of the United States Department of Commerce. It provides technology, measurement and standards that are used in products and services including computer chips, nano-materials, and even the National grid. The NIST Guide was selected as a comparator in this work because it the IR Phases are widely-known and applied globally, NIST is a well-recognised body and the phases in the NIST Incident Response Plan are similar to the one proposed in this work. There are other IR Plans such as those discussed in [82] and [83] however, as explained, the NIST Guide was used in this case because of the fact that NIST is a

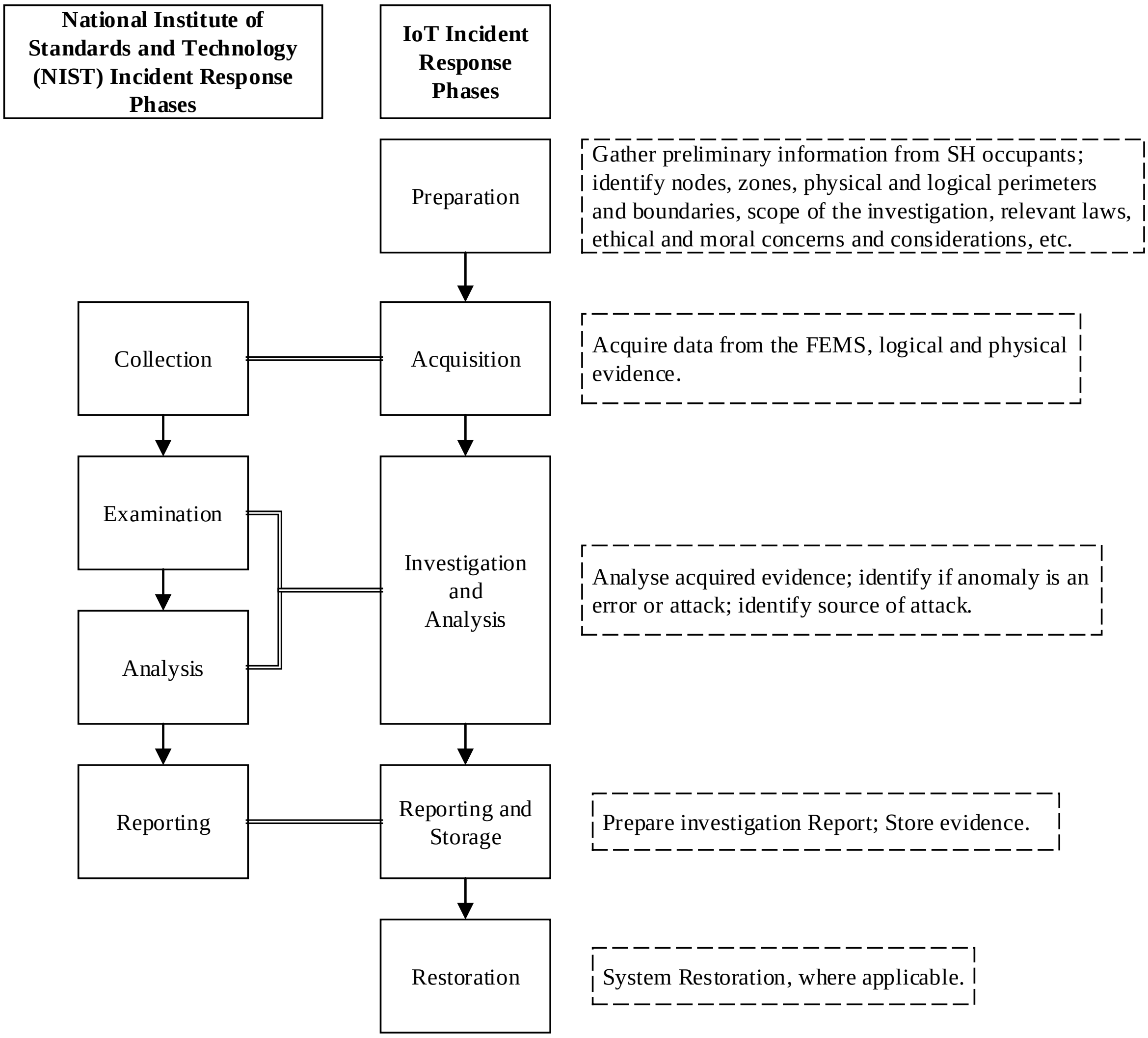
National Institute of Standards and Technology (NIST) Incident Response Phases
IoT Incident Response Phases
Preparation
Gather preliminary information from SH occupants; identify nodes, zones, physical and logical perimeters and boundaries, scope of the investigation, relevant laws, ethical and moral concerns and considerations, etc.
Collection
Acquisition
Acquire data from the FEMS, logical and physical evidence.
Examination
Analysis
Investigation and Analysis
Analyse acquired evidence; identify if anomaly is an error or attack; identify source of attack.
Reporting
Reporting and Storage
Prepare investigation Report; Store evidence.
Restoration
System Restoration, where applicable.

FIGURE 3.3: NIST to IoT Incident Response Plan Mapping

widely-recognised Standards Agency and its standards and systems such as its IR Plan are well-known and relied upon globally.

3.9.1 Purpose

In responding to incidents, IR Teams should ideally have IR Plans. According to Dell SecureWorks (see http://www.secureworks.co.uk/incident-response/), an IR Plan should detail "roles and responsibilities, procedures and communications". They add that if an IR Plan is not used during the investigation of a breach, poor decisions may be made which can "make the breach worse and delay its resolution". Adequate IR planning can therefore help to minimize damage and loss caused by an incident as well as aid systematic system (and data) recovery and preserve evidence should there be a need for legal action. Some of the challenges that DF teams may encounter if there is no structure in place to guide Incidence Response tasks and investigations within the IoT - challenges which an IR Plan can help overcome - include:

- **data loss** due to a lack of established, tested, repeatable, methodical steps;
- **varied outcomes per investigations** due to the absence of repeatable, standard processes and guidelines to assist future investigations;
- **time wastage** which may occur as a result of a lack of a pre-arranged strategy. This can in turn lead to a lack of focus and duplicated efforts or even cause investigators to focus on unnecessary tasks;
- potential **legal and ethical breaches** (e.g. breaches of personal privacy);
- potential **investigation scope creep**;
- potential **evidence volume overrun or mismanagement**. Without a proper Plan in place, investigators may apply a *seize-and-bag-all* approach which may then lead to an **unnecessarily large volume of evidence** that may or may not be useful to ongoing investigations;
- **delayed resolution of incidents** or breaches and even escalation of the effects of the breaches;

An IR Plan will help investigators maximise a response potential and minimise the duration and, potentially, the impact of an incident.

3.9.2 Scope

The IoT IR Plan was developed to be applicable to DF investigations in SH and, more widely, the IoT. It suggests the steps that can be taken during investigations in such environments but it does not prescribe tools that should be used during specific investigations.

3.9.3 Target audience and users

The intended audience for the IoT Incident Response Plan are IR teams, Digital Forensics (DF) investigators, IoT and Smart environments cyber-crime and security professionals.

3.9.4 Application

The IR Plan provides scope for steps that may be taken *before*, *during* and *after* incidents. This means that it can serve as an Incident Response Plan as well as an overall Framework for DF investigations.

3.9.5 Benefits and merits

The ways in which classical/traditional DF methods differ from the IDFF (discussed in [78]) are in the types of evidence, types of networks, types and amount of data (logical evidence), evidence sources, jurisdiction and network types and boundaries. The IoT IR Plan recognises and addresses the challenge of the cyber-physical environments domain with respect to these differences.

Logical evidence in the IoT and SH will be obtained from data from a variety of nodes including but not limited to mobile phones, TCP/IP network data, electricity readings, and data from location and motion sensors. Aggregating this to a useful and manageable form will be crucial for DF tasks because it will make the data easier to manage. One way that this can be addressed can be through the use of a network *Data Aggregation* module where the data from the different nodes are collected and centrally processed. Such a central location can subsequently be queried for evidence during investigations. A Home Management System or even the FEMS can serve as such a central log processing unit. Alternatively, this module may deal with the data about the raw data e.g. rather than store explicit electricity readings, the aggregator may hold information such as in Table 3.5:

Node	**Location**	**State**	**Duration**
`Kettle`	KITCHEN	ON	2 minutes
`Fridge`	LIVINGROOM	ON	10 hours

TABLE 3.5: Smart Home Log Aggregation Sample

By stripping the SH data records of specific, explicit information in this way, when investigators access the data, a degree of the SH occupants' privacy is maintained[3]. The aggregation unit must be adequately secured to avoid accidental or deliberate, malicious log corruption or destruction. In addition, it was developed with a focus on existing laws,

[2] Minimum setup requirements where everything is at a normal state.

[3] This was one of the suggestions from the interviews on the IDFF.

the aim being to identify how these may apply within the CPS investigations. The goal is to eliminate or at least reduce the potential for investigations to be abandoned due to lack of prosecutory provisions or due to potential breaches by investigators themselves during investigations.

3.10 Analysis of a Hypothetical Intrusion Detection Scenario

In this section the IDFF and IoT Incident Response Plan are used to analyse a hypothetical Incident Response scenario and the outcomes of the analysis are presented. The hypothetical scenarios are representative of attacks that SH may realistically face.

Zone	Zone Mapping	Objects
A	LivingRoom	`Television, Sofa, GamesConsoleOne, MobilePhones(2)`
B	Dining	`DiningTable, DiningChairs, FlowerPotOne`, `BabyMonitor`
C	Kitchen	`Fridge, Cooker`, `KitchenTelevision`
D	SmallBathroom	`BathTubTwo` `MirrorTwo`
E	Office	`Desk`, `Chair`, `Cupboard`, `Door`
F	SmallBedroom	`SmallBed`
G	BigBedroom	`BigBed, TableLamp, BabyMonitor, GamesConsoleTwo`, `CCTV Management Console, Home Management System`
H	Hallway	`Thermostat`

Table 3.6: Hypothetical Scenario: Smart Home Nodes

3.10.1 Example Smart Home Incident Response Plan

The SH selected as the focus of the scenario is the Aruba SH with zones and nodes as identified in Table 5.2. Additional Objects (i.e. Smart Home nodes) shown in Table 3.6 - but which are not confirmed as actually existing in the real WSU CASAS Aruba SH - are introduced hypothetically into the home for the sake of analysis and discussion. The investigation is carried out as described in the following sections.

3.10.1.1 Preparation and Preliminary Information Gathering

The source of attacks against SH may not be from single sources and may also not be of one particular type or against single, isolated targets. Adequately preparing to address different types of attacks and to gather as much relevant evidence as possible as part of an incident Response strategy is a key first step in any investigation.

Information gathering about an incident should begin as soon as the Incident Response Team receives the user's complaint and agrees to investigate the reported incident. Preliminary information can be gathered by asking the person(s) who made the report some specific **questions** about the particular incident as well as asking general questions. Some example questions can be *What did you observe?*, *Could* ***you*** *have done anything that may have triggered one or another of these incidents?*, Bearing in mind that the target space is a home, *Since the incident occurred what, if anything, did you do to try to address the issue(s) by yourself? Please provide as much detail as possible.*

It is important that the **scope** of the investigation is ascertained as early as possible during an investigation. Consider for example an incident involving a Hospital. The scope for a hospital investigation can be the **everything within the hospital's physical grounds only** or **all remotely-managed patients fitted with pace-makers**. The scope of an autonomous, self-driven vehicle investigation can be **all cars** or **all cars that have particular software and hardware** installed.

Determining the scope of an investigation involves identifying if the investigation is **digital, physical or cyber-physical** in nature and, where possible, what the physical and logical perimeters of the investigation are. Identifying the perimeter will help to determine if the incident involves physical nodes **local** to the home or if there are **external** nodes involved as well as help identify any logical elements that may be involved. In addition, this stage helps identify if external parties such as family members and friends may need to be contacted as part of the investigation. A potential scenario in this regard can occur if a friend places an order without informing the SH occupant. After, receiving the groceries, the occupant may make a complaint to the store about groceries being wrongfully delivered to their home. Therefore, where possible, every human, non-human, autonomous, non-autonomous entity, node (typical and atypical) involved in an investigation should be explicitly identified. With respect to the sources of logical evidence, the nodes being identified must be nodes that may or *should* contain evidentiary data (referred to as Objects of Forensic Interest (OOFIs) in [78]).

Determining the perimeter of an investigation also helps identify what laws apply to an investigation e.g. do international laws apply or national laws only?

During this phase, the following should also be identified:

1. specifics of individual incidents, how many incidents there are and any links or overlaps between them where applicable;
2. where any SH data logs are aggregated and stored, if at all;
3. the affected nodes;
4. the affected zones;
5. items that can be removed (for *off*-site analysis where necessary) and what **must** be analysed *in situ*;
6. what logical and physical evidence each affected node holds;
7. which of the evidence is volatile. Volatile evidence should, ideally, be acquired first or, as early as possible during investigations;
8. any laws, ethical and moral boundaries that apply for example, whether to potentially interfere with an ongoing CT scan by accessing the PC that is receiving data from it in order to acquire evidence.

This information should be gathered manually as well as from output from the FEMS.

For this scenario, assume that the occupant of the home is physically away from her home and that she left at 8 a.m. From the user complaint and the report of the FEMS the following information was made available to the IR Team and represents the user's **complaint**:

- Incident One: The FEMS "detected" a *presence* in the home at 11a.m;
- Incident Two: The SH network monitoring system reported unusually high network traffic between 10 a.m. and 2 p.m.;
- Incident Three: An unanticipated delivery of groceries was made within this time period. The user claims she did not place this, or any other, order.

The team should ask the home occupant the questions as suggested earlier in Subsubsection 3.10.1.1.

The IR Team will need to find out if any of the attacks are ongoing. If the network is still being accessed and orders placed then they may choose to limit communications on the network by blocking the offending IP addresses or disconnecting the entire network thereby limiting the impact of the attack. However, if this will impact the home occupants or SH itself negatively, they will need to reach decisions about what is the best way to respond e.g. limiting only certain types of traffic and not all or trying to investigate whilst the attach is going on. This, may help them gather even more evidence and verify the source of the attack depending on how skilled the IRT is and the capabilities of their tools. With respect to Incident One, if the person is still present then efforts should be made to apprehend them physically.

During this phase, the Team can find out information about the types of protocols available on the network, the types of nodes - as mentioned earlier. If there is comprehensive list of the nodes in the SH, it would be helpful to the team to acquire this from the home occupant.

This service is provided after the incident occurs. The incident may still be ongoing while the investigation is being carried out. It is cost-effective because it is only carried out when incidents occur which the FEMS tool escalates.

3.10.1.2 Evidence Acquisition

During this stage the IT Team acquires evidence from the nodes, network, zones and humans (as necessary and feasible, under certain constraints). They can do this by connecting directly to the sources of evidence through any available interfaces (physical access) or wireless communications. SH nodes, whether large or miniature, typically have interfaces that can be connected to and used to query for the data in any internal memories that the nodes have. As much as possible investigation teams should try to interface with the nodes and acquire data from them. Wireless (BlueTooth and WiFi) and wired (e.g. Ethernet) sources must all be accessed where possible, bearing the volatility of certain evidence sources in mind.

Evidence of interest will also include IFTTT and other app schedules, app logs and app activities, sensor node meta-data such as node ID, hardware addresses e.g. MAC addresses, if any, node activity details such as when (i.e. time of day) they are turned on and off and the actual sensed data (stored centrally on sink nodes or locally on the individual nodes themselves), smart thermostat logs, humidity readings, car location and/or operating history, motion sensor logs, IP addresses, location data, data stored on any cloud locations, Social Media feeds,and upload locations and schedules for nodes that are programmed to upload data periodically, etc.

Network logs will include network activity data such as log-on and log-off times, meta-data, port details, number of ongoing connections, details about any authentication attempts and using which usernames, grocery and other purchase history, websites visited, and so on.

Some useful sources of data in Smart Homes will include any available networking systems (routers, switches, servers, etc.) some of which may dually serve as log storage systems and/or even Home Management Systems (HMS) [84]. Other sources of data will include mobile phones, FEMS, Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), Firewall, games consoles, pens, smart electricity meters, and fridge memory.

The data that some sensors hold is of a relatively small quantity. For example some wireless sensor nodes have Flash program memory sizes as small as 128 Kilo Bytes and 4KB of RAM. However, the quantity does not matter and if the data can be accessed then the investigators should attempt to access it. Nodes such as fridges and cars may have more data than tiny sensors. They will potentially also have larger processing capabilities. Some larger nodes also carry out some on-board processing and any data stored on them may be useful to a case and this data should therefore be acquired.

The team may need to pair up with a physical Incident Response Team so that not only the logical evidence (for example network logs) is acquired but the physical evidence as well such as fingerprints and fabric.

As explained in Section 1.6 (see also [78]), as soon as any changes in the SH "Happiness" status is detected the FEMS collects the logs from all events just before the time-stamp when the changes were detected. The time frame of when to begin the capture can be estimated based on the time frame given by the home occupant when they made a complaint as well as the investigation team's domain knowledge and experience. The issue of corrupted logs is still an open question as far as this research is concerned. Logs can be encrypted immediately they are acquired and stored however, this may cause the system to slow down thus leading to a potential self-inflicted DoS attack.

The autonomous service provided by the FEMS is a useful part of the evidence acquisition stage because it reduces the potential that relevant evidence is lost because the investigators arrive some time after the attack and, in addition, investigators are afforded the opportunity to complete the key step of preliminary information-gathering from the person making the report. It is also an invaluable source of evidence for investigating teams.

The team will also need to identify what potential physical sources of evidence (typical and atypical) are not connected on the SH wireless or wired network and whose logs would therefore not be available on the FEMS e.g. an analog, non-networkable alarm clock. The IR Team will need to identify the communication protocols (a typical protocol is Bluetooth, a less common protocol is Rime) being used and whether the Team has the tools to communicate with the nodes using those protocols.

In addition, the relationships that the nodes in a SH form with other nodes can be used to determine what activities the nodes were part of. Analysing the attributes, relationships and constraints of these nodes, etc., anomalous relationships can be detected as described in Chapters 4, 5 and 6.

Lastly, if data is not available from certain nodes directly, some information may be available from nodes that they are attached to or with which they interact. This is described as using the *Next Best Thing* (NBT) approach ([81]).

For this particular scenario, evidence is acquired from the FEMS and from the relevant nodes as identified in the Evidence Analysis part of the investigation.

3.10.1.3 Evidence Analysis

Based on the initial complaint made by the home-owner (and any available report from the FEMS) the three incidents [4] identified are analysed here:

Incident One:

The FEMS "detected" a presence *in the home at 11 a.m. The three zones in which the presence was detected are the* Dining*, the* Hallway *and the* Garage*. The home occupant was away at that time. The home is a single-occupant home. No visitors were expected that day so any presence while the home occupant was away was anomalous.*

The IR Team should physically go to the SH and check to see if any logical or physical evidence that proves that there was - or still is - a *presence* in the home is available. They should also look for evidence about what zone the presence was/is in. Note that by identifying the affected zone(s), the FEMS helps to narrow the scope and perimeter of the investigation but investigators may still choose to gather physical evidence (e.g. from CCTV recordings) for themselves and use that to corroborate the report from the FEMS. The IRTs will also need to also look at the logs of External Access Nodes.

[4] It should be noted that there might be an overlap between incidents especially considering the fact that, from the user's initial complaint, the three incidents occurred within a common period on the same day in the same SH

Assume for the purposes of this investigation that the FEMS - using the presence detection capabilities of the FDMA algorithm (Appendix A) - detected the *presence* in the DINING.

Incident Two:

The SH network monitoring system reported unusually high network traffic between 10 a.m. and 2 p.m.

From the FEMS output, the team can find out which nodes were active on the network during the period in question i.e. between 10 a.m. and 2 p.m. They can also find out details of any outgoing and incoming traffic from any network monitoring nodes or software like Intrusion Detection and Prevention Systems (IDPS), Firewall and the FEMS where they are available. If the excessive traffic (represented on the FEMS log as Nx = Up-High, the network state that indicates an attack) is discovered to be in-bound traffic from a single or multiple sources the investigators might be able to identify the source and destination IP addresses of the traffic. Even if the source address were spoofed, it might still be useful to collect this information as a way of evidencing the attacker's skill-level, their intention and their modus operandi for potential future profiling of the attacker. The investigators will also need to check to see what ports were active around this time and if the ports were on the list of allowed ports as well as if the number of active ports was greater than the maximum allowed i.e. Ptmax or not even on the list of identified or allowed ports.

Since the third incident involves the unexplained placing and delivery of an order, the team may also choose to search for which websites, if any, were being accessed. The search history will also be important as it might include search terms that include the grocery items that were physically delivered as well as delivery companies and potential grocery shops. Of particular importance will be any **activities** that involve access to online grocery stores, delivery companies and bank purchase.

Incident Three:

An unanticipated delivery of groceries was made within this time period. The user claims she did not place this, or any other, order.

The IR Team may begin by analysing any physical signatures that were used to sign for and accept the delivery if any are available. They may also analyse purchase receipts because these may have card details and details about the name of the person who placed the order. The shopping bags in which the items were delivered might give an indication of which store(s) the order was placed from. This is especially useful if the delivery driver is not identifiable (e.g. if they left after making the delivery and before the home owner returned to the house) and no receipt is available.

The team can check for which nodes are typically used to place orders for groceries. If orders are typically placed directly from the `Fridge`, then they might choose to access the `Fridge` logs for potential evidence of websites that were communicated with, shopping history, and which activities from the Activities List in Table 6.1 were performed. They may choose to find out if the `Fridge` was scheduled to place any orders by itself and if perhaps a software glitch caused it to place the order. In addition, it will be important to find out if any orders were placed from remote locations using the fridge itself (see Subsection 6.3.2 which discusses three potential ways in which groceries can be ordered using a smart fridge).

If the order was placed from the fridge, locally or remotely, the login details (usernames) used may also give some indication of who was logging in. In addition, the *number* of login attempts on the fridge may indicate if it was an authorised user or not. To eliminate any possibility that it was the home occupant, they may need to look at her bank statements (payment history) that cover the period that the order was placed during according to the records from the store.

Since the FEMS output provides a time-stamp that indicates when the *presence* was detected (Incident One), the team can look up what nodes were active around this time. Assume that the nodes that were identified in Incident One are the `DiningTable, DiningChairs, FlowerPotOne`, the motion sensors `M021, M022, M029, M030, D004` as well as an extra, unregistered node of Type *Generic Mobile Communications Device.* The last node on the list indicates that there was one more node than expected on the network. However any active, non-proximal sensors may require further analysis. If it turns out that there was no outgoing traffic from the fridge, the IRT will need to investigate the unrecognised device as a possible suspicious source.

The FEMS also holds a record of how many active, networked, nodes (N_TAct) there were in the home including during this time period (see Section 4.2.1.16) so that if more nodes than expected were active, the system can send an alert warning of a possible physical intrusion in the home.

Since the *presence* was detected in the DINING at 11 a.m., assume for this incident that the network traffic logs showed an unknown *Generic Mobile Communications Device* making an outbound connection to a well-known grocery supplier's website. This could be the reason for the unusually high traffic because there was some network activity when none was expected thereby causing the network reading to be at state Up-High instead of Up-Normal (Section 4.2.1.12).

For this incident assume that no one physically signed for the delivery and the delivery driver had placed the shopping outside the home before making the call to the home-owner about the order. Also assume for the purposes for this analysis that the unrecognised node was found to have accessed shopping websites around the time of the incident. This was based on evidence from websites visited which showed that orders were placed and banking details entered. Lastly assume the firewall reported that the high network traffic was coming from multiple IP addresses external to the SH but that the destination was the IP address of the fridge in the kitchen.

The variety of potential data sources and, therefore, potentially, data types, demonstrates that stripping of the data and aggregating it using tools like the FEMS will prove useful for Digital Forensics purposes as it will save time and reduce the need for investigators to have tools to interface with the different interfaces and analyse the different data types. If this investigation was to be conducted without a system like the FEMS it would have been more challenging. The potential contributions of the FEMS to Incident Response and DF have been demonstrated here and include time-saving and a more efficient approach to investigations.

3.10.1.4 Investigation Conclusion

One conclusion is that there was someone in the house when the legitimate home occupant was away. The person came in with an unrecognised device which was used to place an order for groceries. During this time, there was a separate attack going on. The second attack involved a potential network-based (i.e. not physical) Denial of Service (DoS)[5] attack on the Smart Fridge.

[5] An physical, logical or cyber-physical attack that causes resources to become limited or completely unavailable to those who legitimately need it.

Factors	Incident one	Incident Two	Incident Three
Perimeter and network boundaries	Dining	Home* network and external network	Home* network and external network
Relevant Evidence Sources	Motion sensors (may also include CCTV as backup)	Network, affected nodes, router(s), switch(es)	Routers, switches, motion sensors, receipt, physical items delivered, shopping bags, fingerprints
Affected zones	Dining, Hallway, Garage	Home* and external	Home* and external
Affected Nodes	`M014, M021, M022, M030, D004`	FEMS	Outside the home
Cyber Evidence	Motion Sensor data	TCP/IP data, IP addresses, ports	FEMS network data , network activity of the unrecognised node, any available purchase history from groceries supplier with destination physical address of the SH in question, Internet History,
Physical Evidence	Fingerprints on `D004`, `DiningTable` and `FlowerPotOne`; other evidence of physical intrusion[6]	None	Physical items delivered, fingerprints, Fabric, Location information, receipt, activity logs
Conclusion from evidence	Intruder in home at 11 a.m. in Dining and moved between Dining and Garage only; touched objects as listed in *Physical Evidence*.	Someone placed an order using the unrecognised device.	n/a
Legalese	Theft Act 1968	CMA 1990 (c 2006),	CMA 1990, Communications Act 2003, Theft Act 1968, Telecommunications Act

Table 3.7: Hypothetical Scenari

Incident Response Report	
Report Author:	**Mary** (name and signature)
Start of Report metadata:[7]	**00/00/1900; 10:00 a.m.; Osas, case officer**
Beginning of Investigation metadata:[8]	**00/00/1900; 10:00 a.m.; Osas, case officer**
Incident(s) Reported by:	**Home Owner** (verified)
Required Permissions obtained:	**Yes**
Estimated log duration involved:	**5 hours**
Size of evidence e.g. logs:	$\sim$ **20Mb**
Current Location of logs:	**FEMS & USB drive**
Decision One:	**True Positive**
Source of anomaly (if any identified):	
Decision Two:	**True Positive**
Source of anomaly (if any identified):	
Decision Three[9]:	**True Positive**
Source of anomaly (if any identified):	
Action(s) Taken:	See Case Officer's Investigation Notes
Reason(s) for Action(s) Taken:	See Case Officer's Investigation Notes
Current Smart Home state:	**"Happy"**
People involved, if any:	**Home Owner**
Applicable Laws[10]:	**CMA 1990**, **Theft Act 1968**, **Communications Act 2003**
End of Investigation metadata:	**00/00/1900; 11:00 a.m.; Osas, case officer**
End of Report metadata:	**00/00/1900; 11:30 a.m.; Osas, case officer**

Table 3.8: Hypothetical Scenario: Incident Investigation Report

3.10.1.5 Reporting and Storage

During this stage the Team prepares a report based on the outcome of their investigation. The logs that were investigated are stored for a period. The report for this hypothetical incident is given in Table 3.8. An example logs storage table is Table 3.3.

3.10.1.6 System Restoration

Where feasible, after an investigation, the SH network should be restored to its healthy state. This can be achieved through installing software patches, removing offending nodes or applications from the network, updating anti-malware solutions such as anti-viruses, installing network monitoring tools if none were previously available and if necessary, moving nodes back to their correct locations, registering new previously unrecognised nodes that have been added onto the SH network, among other measures.

[6] CCTV footage as backup.

[7] Date, Time, Signature.

[8] Date; Time; Name; Signature of Case Officer or Officer-in-Charge.

[9] There are three (3) decisions because there are three incidents.

[10] e.g. which laws apply to the incident? Is the incident a *reportable* offence under, for example, the UK law?

Part of completing the investigation may also include putting incident prevention strategies in place. All of this has to be done with the home occupant's agreement especially since there will most certainly be a cost dimension to implementing these solutions however this cost might be part of a pre-existing Service Level Agreement that the home occupants may have with their SH solution providers or home insurers or even cyber-physical security and forensics companies.

3.11 Chapter Summary

The Chapter proposed a definition of the word "Things" in the context of the Internet of Things (IoT). An increased reliance on autonomous smart Things by humans and end users to obtain improved efficiencies in everyday life might be a consequence of advances in the reliability of smart Things as part of the IoT. This Chapter therefore proposed the use of Responsibility Models to demonstrate that even though there might be an *abstraction* of responsibility to Things when they perform tasks and respond to requests on behalf of end users, these end users (e.g. owners of self-controlling, autonomous systems and objects) should still be aware of their potential responsibilities for the *consequences* of the actions of Things. The Chapter also discussed the Internet of Things Digital Forensics Framework (IDFF), the Forensics Edge Management System (FEMS) and the Forensics Decision-Making Algorithm (FDMA) and illustrated how the three relate to each other in Figure 3.2. The IDFF is a framework that describes, in phases, a potential approach to cyber-physical Forensics within SH environments. The FEMS is a system that is designed to provide security and forensics services within SH environments as part of an overall Digital Forensics preparedness strategy. The FDMA is the decision-making module or component of the FEMS. It analyses SH datasets - and, based on several factors (attributes listed in Chapter 4 and core activities and relationships, explained in Chapters 5 and 6) - makes a decision about whether an observed occurrence on a network is an anomaly or not. This Chapter also introduced the Internet of Things Incident Response Plan and demonstrated its applicability by analysing three hypothetical Incident Response scenarios using the IoT Incident Response Plan.

Chapter 4

Definitions and Methodology

4.1 Overview

The thrust of this work centres on the fact that nodes and zones within and around Smart Homes (SH) have **attributes** that are non-changing or fixed (such as the *location* of a fridge), semi-static (e.g. the *locations* of window blinds and cars) and, predominantly variable (e.g. the *locations* of mobile phones), that these attributes can be used to identify **relationships** between zones, nodes and users and that **rules** can be drawn up to represent and constrain these relationships.

This Chapter introduces the formalisms used in the Smart Home (SH) Anomaly Detection (AD) framework to define the attributes and relationships that exist between nodes and zones in Smart Homes. These relationships can be temporal (related to time) or spatial (related to locations in the Home) or a combination of both. Realistic scenarios can be created to represent events and activities in SH such that if the relationships that are created within these scenarios do not conform to the *learned* or *known* Normal then the "Home Happiness" state is affected (an anomaly). In Chapter 5 several realistic scenarios are analysed to demonstrate the use of these relationships in anomaly detection.

4.2 "Happy Home Network"

This section introduces the concept of a "Happy Home Network". A "Happy Home Network" is one in which all the conditions for normality obtain, with respect to specific thresholds. These Normality conditions may be the **activities at *time of day***, or ***number of occupants at time of day***. In summary, a "Happy Home" is a set of "Happy Zones" which are made up of "Happy Nodes" where *Happy* is defined by attributes that meet certain criteria within defined thresholds. Some example attributes are **Node** η, **Type** γ, **State** S, **Location** Z, **Proximals** ρ, **Network State** Nx, **Peripherals** Per, **Mode** μ. Subsection 4.2.1 and Table 4.4 provide more details about each attribute. Each zone (i.e. room) and node (sensor, device, object) in and around Smart Homes can have its own set of corresponding attributes. Not all possible attributes apply to all zones (or nodes) and not all of the attributes are required to determine the "Happiness" status of a zone (or node). For *Happy* nodes and zones all the applicable attributes fall within defined acceptable thresholds. Therefore, the set of all the applicable attributes per zone (and node) at their normal state represents the *Security Baseline* of a SH where an IT Baseline, according to [85], is a "checklist against which systems can be evaluated and audited for security posture".

The following Formalisms or representations are used in the rest of this work:

Box 4.1. Formalisms

Zones are written in all Small Capitals font: e.g. Kitchen

`Nodes` are written in the `teletypefont` font: e.g. `KitchenDoor`

Events and *Activities* are written in italicised *red coloured font*

Attributes have *threshold-constrained* sets of values: the **Network State** (Nx) attribute can be Up-Normal, the **Location** (Z) can be Zone C or, more specifically, Kitchen (see Figure 5.2), the **Type** γ can be `Fridge` and the **State** S of the node can be *open*. Some attributes are *fixed*-value attributes while others can be adjusted. Consider the example of a `Fridge`: the attribute *Colour* is fixed while the *Location* attribute can be changeable, adjustable or non-fixed or non-static or variable; the *Size* attribute has a fixed value since the size of the fridge is not an adjustable quantity. Lastly, the *Mode* of a `Fridge` is also fixed.

The behaviours, user interactions with - and attributes of - nodes form well-defined relationships as users interact with the nodes. For instance, if a light comes ON as a user enters their Kitchen, the action of the user caused a state change in the node `Light`, in the Zone Kitchen and in the readings of Motion Sensors around the home (for instance in the occupant's previous zone (-Z)). These behaviours, interactions and attribute values can be captured and harnessed for Anomaly Detection Purposes.

Smart **Home** (Networks) are made up of **Zones** and **Zones** are comprised of **Nodes**. A generic SH Attribute Tree Structure is illustrated in Figure 4.1 while a specific example SH Attribute Tree Structure in which η = `Fridge` is illustrated in Figure 4.2.

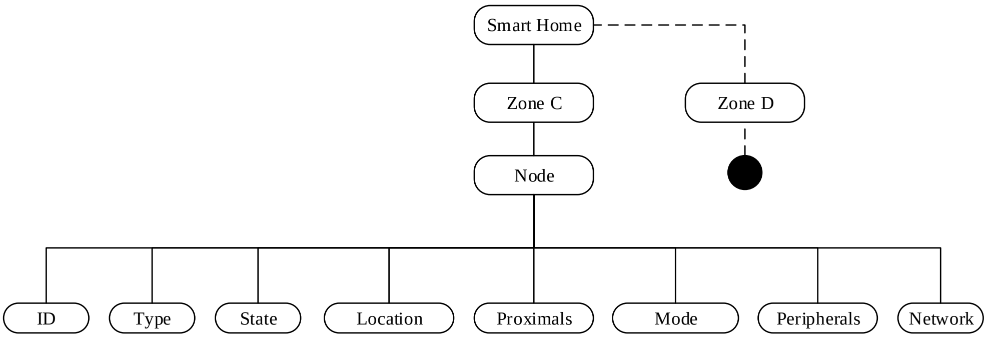

FIGURE 4.1: *Home-Zone-Node* Tree

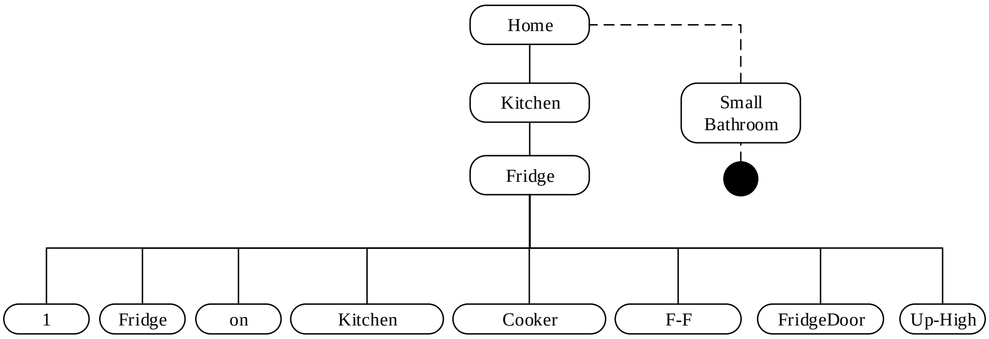

FIGURE 4.2: *Home-Kitchen-Fridge* Tree

4.2.1 Attributes and Relationships

Within the "Happy Home Network" Framework, several attributes are defined; these are discussed in this section (also see Table 4.4).

4.2.1.1 *Device or Node ID (η)*

This is a unique identifier assigned to a device when it is first registered on a Home network. Example Node Identifiers (η) are: ***001, 002,..., 100*** and ***a,b,c,d***. The ID is useful for easy identification of nodes both by the human occupants and by any networking or IT security/forensics system that is in place.

4.2.1.2 *Mode (μ)*

In this work, a node's location attribute is referred to as its *Mode* (μ). At any given time, the μ of each Zone and Node can *only* be one (1) out of a set of four (4) values (Table 4.1). A change in a node's μ from its original value can be used to detect a location change as well as potentially identify the activities surrounding the change of location. For instance, a user's presence next to a Fridge before it changes location might indicate that the user moved the Fridge. Conversely, the absence of any user activity next to a node that changes location could mean that the reported location change is a false alarm (i.e. an error by a location-detection system) or a *simulated location modification* which might be indicative of a network-based cyber attack.

Mode (μ)	Short Form
Fixed-Fixed	F-F
Fixed-Mobile	F-M
Mobile-Fixed	M-F
Mobile-Mobile	M-M

Table 4.1: Types of Smart Home Node Modes

By way of examples, the following Mode Types apply to the following nodes (Table 4.2):

Entities	μ
Fridge	F-F
Bed	F-F
Car/24-hour day	M-F
Motion sensor	F-F
Kitchen	F-F

Table 4.2: Examples of Smart Home Node Modes

All **zones** in all Smart Homes have a μ of F-F (unless the home is a caravan or other movable or reconfigurable home e.g. a home with movable walls).

Illustration of AD using a node's Mode (μ) value

Box 4.2. Anomaly Detection using the Mode (μ) attribute

Given
normal location = Z
previous location = $(-Z)$ and
current location = $\acute{Z}$
For all nodes with μ = **F-F**,
$Z = -Z = \acute{Z}$
and any change in this relationship for the duration of an event will trigger an anomaly.

In this work the focus in on the two Mode (μ) values F-F and M-M because the assumption is made that most nodes are either predominantly Stationary (F-F) or predominantly mobile (M-M). For *stationary*, fixed-zone sensors, their proximals and proximal relationships never change. For *non-stationary* or mobile sensors, the reverse is true and their proximals may vary over time thus altering the proximality relationships over time.

An AD system can adapt to special situations, for instance by recognising that certain nodes may have either a Mode (μ) value of M-F or F-M and that a change in location of nodes of this kind would not immediately be indicative of an anomaly; such a system instead can calculate how long they have been static in comparison to how long they have been mobile and therefore is able to make a decision about whether a reported incident is the sign of an anomaly or not.

An AD system can adapt to special situations, for instance by recognising that certain nodes may have either a Mode (μ) value of M-F or F-M and that a change in location of nodes of this kind would not immediately be indicative of an anomaly; such a system instead can compare a current situation of the Mode of a node with how the same node is normally static and how long it is normally mobile and therefore make a decision about whether a reported incident during a certain period is the sign of an anomaly or not.

4.2.1.3 ***Proximals (ρ)***

The Proximality relationship between nodes i and j is written as

$$\mathbf{i}\triangle\mathbf{j}$$

Proximality describes the relationship that exists between nodes that respond almost simultaneously when stimuli is applied to only one of them directly. This is especially observable in how motion sensors fire in the Aruba Dataset (available at `http://casas.wsu.edu/datasets/`).

Consider, for example a situation where a user passes node a (or η_a) on their way to η_c. If after the home occupant passes node η_a but before she gets to node η_c, node η_d comes *on*, even if it was not interacted with directly by the user, a number of factors may influence the decision (normal or anomalous) taken by an AD system such as the FDMA. One such factor is the *relationship* between the nodes. If $\eta_a \triangle \eta_d$, then such a response is not anomalous. However in the case where the responding nodes (or zones) are non-proximal (e.g. $\eta_a \bigtriangledown \eta_d$), then, this type of reaction to stimuli applied to either (but only) one of them is anomalous.

The conditions for Proximality are:

- Nodes in the same or different zones can be proximals. Proximal nodes in the same zone are known as intra-Zonal Proximals and sensors in different zones share an inter-Zonal proximality relationship (Figure 4.3);

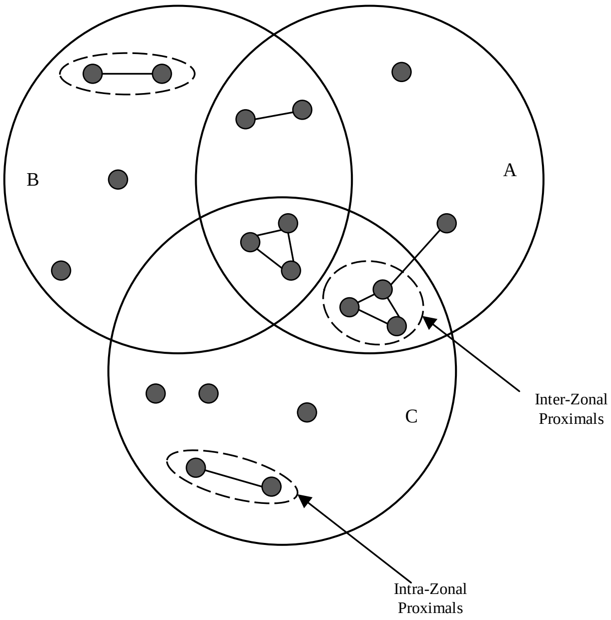

FIGURE 4.3: Intra- and Inter-Zonal Proximality

- Observe from Figure 4.3 that even though zones are proximal it does not mean that *all* the sensors *within them* are.

- All nodes within the same zone are proximal unless they are separated by a distance greater than a maximum ξ and/or by a significant solid obstacle such as a through wall. See Subsubsection 4.2.1.21 for further details;

- Proximality cannot be *transferred*. There is neither a forward nor a reverse transference of Proximality between non-Proximal zones (or sensors) even if there exists a common zone (or sensor) to which each of them is separately Proximal (see Figure 4.4; the cross indicates the action is disallowed; the tick mark indicates an allowed or permitted action);

- Modes with μ = F-F always have a non-changing set of proximals (whose μ are also = F-F) unless the proximals in question are of any other mode such as M-M. Therefore if a `Fridge` has the attributes μ = F-F, the proximals of the `Fridge` which have μ = F-F will have a static, unchanging proximality relationship with the `Fridge`.

Bearing these conditions in mind, from a visual observation of a section of the WSU CASAS Aruba SH dataset (see Table 1.1) the BIGBEDROOM proximals table (Table 4.3) can be drawn up. This is an example of a table of proximals. To populate this table and identify the Proximal relationships, an observation-based investigation was carried out. The investigation focussed on how the sensors responded to the user(s') actions inside

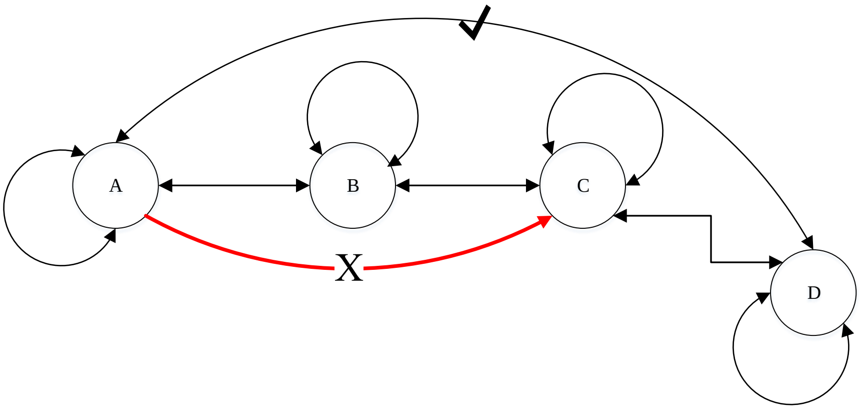

FIGURE 4.4: Illustration of Proximality rules

and around the BIGBEDROOM. The focus of this observation is on the motion sensors. It was observed that every time a sensor in a particular room (zone) was activated (*on*) some of the same sensors around it also became activated, either immediately after or a few seconds after. This was observed to exist between the sensors as shown in Table 4.3. It was also observed that sensors between different zones, but separated by a small enough distance (e.g. the sensors M014 and M021) came *on* within the same time frame. The sensors in different zones do not trigger when one o the other is triggered. This is observed to be the case throughout the section of this large dataset which was perused. It was therefore considered anomalous when sensors did fire in this way. Observe from the WSU CASAS Aruba dataset (*No.17* `http://casas.wsu.edu/datasets/`) that the motion sensors in zones A and E or zones C and D or zones C and F never fire together normally.

Thus, a table of proximals for a SH can, initially, be created from observation and domain knowledge about the SH and the user(s') actions. Building the table involves observing sensor reactions or making assumptions based on physical proximity. However, subsequently, such a table of proximals can be updated automatically by a SH security system.

Node (η)	**Proximals** (ρ)
M001	M001, M002, and M007
M002	M001, and M002
M003	M001, M002, and M007
M004	M029 and M031
M005	M025, M026, M027 and M028
M006	M023 and M024
M007	M001, M002, M003, M004, M005, M006, M007 and M008
M008	M022 and M023

TABLE 4.3: Big Bedroom Proximals

4.2.1.4 *Current Proximals ($\acute{\rho}$)*

This is the current set of nodes that are proximal to a focal node. The normal proximals of a node with μ = F-F also have μ = F-F. Take as an example a `Fridge` in a KITCHEN next to `cooker`. For AD purposes, the proximals list at the beginning and at the end of an event are compared. If the proximals on the list have changed - or the *number* or proximals has changed - the system can infer an anomaly such as an anomalous change in location or faults in the 'missing' proximals. The number of proximals that a sensor normally has is denoted by N_{ρ}.

Box 4.3. ACTIVE ZONE ⇒ PRESENCE

Let ρ = List of Proximals; $\acute{\rho}$ = List of Current Proximals; $\hat{\eta}$ = a node; $\hat{\eta}$ = a proximal of $\hat{\eta}$
For nodes with μ = F-F, $\rho = \acute{\rho}$ *iff* the proximals of the nodes also have Mode = F-F over the duration of the event being considered
Therefore, over the duration of an event, given $\eta \bigtriangleup \hat{\eta}$
For every η in Home* (with μ = F-F), $P = \acute{P}$
and
For each $\hat{\eta}$ in Home* (with μ = F-F), $P = \acute{P}$

!

S/No.	Symbol	Description	Example values
1	η	Node ID or Device ID	A whole number e.g. 1
2	μ	Mode	Fixed-Fixed (F-F), Fixed-Mobile (F-M), Mobile-Fixed (M-F), Mobile-Mobile (M-M)
3	ρ	Normal Proximals of each node/zone	$\eta_1, \eta_2, ...\eta_n$
4	$\acute{\rho}$	Current Proximals of each node/zone	$\eta_1, \eta_2, ...\eta_n$
5	Per	Peripherals	`FridgeDoor`, `CatFlap`
6	Pr	Presence	$0, \geq 1$
7	Z	Normal Location of a node	KITCHEN
8	$\acute{Z}$	Current Location of a node	GARAGE
9	$-Z$	Previous Location of a node	DINING
10	γ	Node Type	`Fridge`
11	S	State	*on, off, open, closed, present, absent, active, inactive*
12	Nx	Network	Up-High, Up-Normal, Down
13	Pt	Allowed Ports	port 161; port 21
14	N_T	Total Nodes	A whole number e.g. 5
15	$\mathrm{N}_T\mathrm{Act}$	Total Number of Active nodes	A whole number e.g. 2
16	N_ρ	Total Number of proximals	A whole number e.g. 10

TABLE 4.4: Attributes Table

4.2.1.5 ***Peripherals*** (Per)

These are nodes which are physically attached to each other. Peripheral nodes can be used to deduce more about the activities around nodes than can be obtained from the nodes themselves alone. Zones can also have peripherals: one peripheral of the BigBedroom can be a node `BigBedroomDoor`. Specific node peripherals are denoted by η_{per}. Other examples of peripherals are `FridgeDoor` and `CatFlap`. Peripherals are always proximal to their main nodes. Triggering the action *open* on the Fridge peripheral `FridgeDoor` will cause the main node `Fridge` to go to state *active*.

Identifying peripheral nodes is useful for Intrusion and Anomaly Detection purposes because accounting for the states of peripherals can be useful in eliminating - or at least reducing - False Positive alarms. For example, if a `FridgeDoor` becomes active while a user is accessing the `Fridge`, the FDMA anomaly detection system will not automatically consider these to be two separate and unrelated events because of the relationship between the `FridgeDoor` and its peripheral. This therefore reduces the potential for one of these activities to be identified as an anomaly.

4.2.1.6 ***Presence*** (Pr)

This value defines if there is a user in a zone or not. It is denoted by Pr and the value is either ≥ 1 or 0. To determine a user's presence within a SH - for the sake of simplification - the following assumption is made: an active motion sensor in a zone Z implies an active zone and is therefore the current location of the home occupant. Therefore, an when a motion sensor detects motion in a zone Z of a single-occupant home, it means there is *at least* one *presence* in that zone Z.

4.2.1.7 ***(Normal) Location*** (Z)

Example locations around a SH include Bathroom and LivingRoom. The normal location of a `Fridge` is the Kitchen and the predominant location of a `Car` is, for example, a Garage. As stated previously the location attribute of a node with $\mu =$ Fixed-Fixed (F-F) is always Fixed-Fixed (F-F). The normal location is a fixed value that is recorded per node and which can be compared against for AD purposes. The normal location is sometimes the same value as the Current Location ($\acute{Z}$) of a node.

4.2.1.8 *Current Location ($\acute{Z}$)*

The current location attribute can be used to determine if a node has changed zone. In conjunction with other factors such as a node's Mode (μ), the current location value can be used to make decisions about whether this detected location change is normal or abnormal. This can be useful for detecting anomalous physical relocation or theft. For example, consider a node `Fridge` with μ value of F-F: if a change is reported in the values of its normal location and its current location such that the normal location is not the same as the current location, this will indicate that the node has been abnormally relocated.

Another location value, **Previous** Location -Z, is also used for determining where a node or person has *been* such that it can be compared with their current location or normal location for the purposes of anomaly detection.

4.2.1.9 *Type* (γ)

The *Type* value gives the description of what the Node actually. For example, the γ of the node `D002` is door (see Figure 5.1) and the γ of a device `Fridge` is fridge. The node type of every node is stored during the registration process i.e. when nodes are first registered on the home network. This data can be added to the security system's database manually or automatically through scanning a tag such as an RFID tag which will have the information of the node stored on it.

4.2.1.10 *State (S)*

This is the current state or status of a device. This attribute describes the different states that nodes (η), Zones (Z), networks (Nx), etc. can be in. Some example states include *on, off* for motion sensors and *Up-Normal, Up-High, and Down* for network.

4.2.1.11 *Powered Nodes*

Powered nodes can be in either states *on* or *off*. This is a useful attribute for AD. An example rule that can be set can be that a device of Type (γ)“*Fridge*”*mustalwaysbeonor*“*offforlessth*
uplistcanthenbeupdatedbythesystemwhichwillinferthisinformationbasedonthepowerandenergyread

Node γ	Hours	Powered Node?
Mobile Phone	500	No
Fridge	620	Yes
Car	20	No
Door	n/a	n/a

Table 4.5: Sample Average 30-day power usage readings

4.2.1.12 *Network (Nx)*

The Network (Nx) value can be measured in the form of maximum and minimum throughput value e.g. 10Mbps. The *up-High*, *up-Normal*, *Down* network readings are opted for in this work for reasons of abstraction and simplicity. The state *up-High* can be indicative of a Distributed Denial of Service (DDoS) attack on a SH network; *up-Normal* indicates normal activity for instance, a user placing an order; *Down* indicates the network is down. An alternative way to describe this will be to use the phrase *Network Throughput*. The Average Network Throughput measure is used to monitor for excessive packet activity on or across a network which may be a sign of a DDoS attack. The alternative, more specific measures of average bytes per packet (bpp) [87] can be used as a baseline against which any packets traversing the network are compared. The packets per communication of flow (ppf) [87] can also be used to determine if there is an attempt at flooding the network.

An abnormally high Nx value could be a sign of a cyber-attack. In the *up-High* case, there is too much activity (this could be because there are too many packets passing than normal, or too many (active) connections, or too many unrecognised ports being opened) on the SH network compared to a normal baseline. An average or normal network baseline can be established over a period of months. This measure is based on the *whole* network and not on the network activity of a particular node. However, the network value can be applied on a Node by Node basis.

4.2.1.13 *Allowed Ports (Pt)*

This is a list of *all allowed* ports on the network. Example ports include 21, 22 (File Transfer Protocol or FTP), 161 (Simple Network Management Protocol or SNMP), 25 (Simple Mail Transfer Protocol or SMTP), 110 (Post Office Protocol III or POP3) and 25 (Telecommunications Network or Telnet).

4.2.1.14 *Open Ports (Pt_n)*

This is a list of all currently *open* or *active* ports on the network. The number of open ports (Pt_n) must be $\leq \mathrm{Pt}_{max}$, where Pt_{max} is the maximum number of (known) ports that may be open; if the number of ports that are open is higher than Pt_{max}, an anomaly will be flagged because this can be the sign of an anomaly such as a IP-based network attack.

4.2.1.15 *Total Number of Registered Nodes in the Home (N_T)*

Total number of registered Nodes in the Home(N_T) is used as a measure of how many nodes are actually on the home network. This can be used to decide if the total nodes exceed the maximum registered or expected nodes.

4.2.1.16 *Total Number of Active nodes (N_TAct)*

This is a list of all nodes which are active on the home network. The number can be ascertained using a node recogniser or other such identity management on a Home Management System. [88] and [89] describe some Energy Management Systems and similar systems exist for device identity management.

4.2.1.17 *Day/Date (D)*

The value provides information about the current date and day.

4.2.1.18 *IP Networked Nodes (N_N)*

These are nodes that are on a TCP/IP based network and to which IP addresses have been assigned either manually or using automated methods such as the Dynamic Host Configuration Protocol (DHCP).

4.2.1.19 *Thresholds*

These are the attribute values beyond which an alarm is raised. Different attributes have different thresholds and thresholds can be a function of time or location and so on.

4.2.1.20 *Edge Nodes*

These are nodes which have no physical barrier or obstacle between them but which are separated by the least distance between 2 or more zones. Therefore, *Edge Nodes* occur at the intersections between two or more sets of zones.

Box 4.4. Identifying Edge Nodes

```
Given two nodes $\eta_i$ and $\eta_i$
where $\eta_i \neq \eta_j$
if $|\eta_i \ \eta_j| \leq d_{Th}$          ▷ the distance between them is less than a Threshold value e.g. $\mathrm{d}_{Th}$
$Z(\eta_i) \not\equiv \mathrm{Z}(\eta_j)$,          ▷ they are in different Zones
and $\eta_i \bigtriangledown \eta_j$ then          ▷ they are not proximal to each other
$\eta_i$ and $\eta_j$ are Edge Nodes.
```

The relationship between edge nodes is used in the development of the FDMA to increase the accuracy of the algorithm by reducing False Positives.When two motion sensors are active within a short period of each other such that they appear to be proximal or peripheral - but are not, and they are in different zones but separated by the closest distance between the two zone with no obstacle between them, the fact that they stay on even after the user has passed form on to another is not a sign of an anomaly. the two nodes are Edge nodes and this is a common behaviour of edge nodes as observed from the Aruba SH dataset.

4.2.1.21 *The Obstacle Factor (δ)*

For instance, nodes M018 and M031 in Figure 5.1, even though they are close and thus may appear to be proximal based on distance, are separated by a barrier and any action that triggers one of them cannot be expected to normally trigger the other. If this happens, it is anomalous. The OF is important from the perspective of real-life DF and Incident Response aims and goals because it will help reduce false alarm triggers.

Although nodes that are *physically* close (i.e. distance < threshold) may be proximal (see M002 and M011 in [1]), those that are separated by an obstacle such as a wall are not. For two nodes not separated by an obstacle the distance between them is calculated as a linear value whilst for two nodes separated by an obstacle, the distance between them is the linear distance multiplied by an Obstacle Factor (OF). This Obstacle Factor is introduced to account for this obstacle between two nodes that are separated by a distance that meets the proximality requirement but which cannot logically be proximal to each other due to the obstacle. The *OF* is therefore used to determine the ***effective distance*** between two nodes.

Illustration

If
ξ = the maximum distance between nodes beyond which they can no longer be proximals
δ = the Obstacle Factor (*OF*)
x = the distance between two nodes separated by an obstacle where
x is always $\leq \xi$

i.e. distance $\mid \eta_i,\eta_j \mid \leq \xi$ is a condition of Proximality

If the *linear* distance between two nodes is $\leq \xi$ then the nodes can share a proximality relationship. Thus the *OF* value, δ, is calculated as follows:

$$x \text{ x } \delta > \xi$$

For the distance between two nodes separated by an obstacle to always be greater than the minimum distance between them, then

$$|\eta_i \eta_j| > \xi$$

Box 4.5. Obstacle Factor (δ) Illustration

If $\xi = 10$ and $x = 4$, then
where ξ is the maximum distance and x is the distance between two nodes. The obstacle Factor (δ) is calculated thus:

$$x\delta > \xi$$

Alternatively[a],

$$x\delta + 1 = \xi$$

Therefore

$$4\delta + 1 = 10$$

Further calculations show that:

$$\delta = 2.25$$

[a] The approximate value of '*1*' was selected because it is the smallest, whole, natural number.

The δ value therefore varies, depending on the value of x and the minimum value varies by home arrangement and setup. An example scenario that illustrates how the OF can be used is in Subsubsection 5.4.2.3.

4.2.1.22 *Access Nodes*

These include doors and windows, cat-flaps, garage doors, among other entry- and exit-ways. These can be in states *open* or *closed*.

4.2.1.23 *External Access Nodes*

Just as with access nodes, External access nodes, are entry- and exit-ways of SHs. External Access Nodes however have special significance with respect to the physical security of SH. This section discusses the relationship and constraints that can be applied between External Access Nodes (for the purposes of this work, these are the *Doors*) in a single-occupant SH in order to aid AD purposes. The relationship and constraints can be explained as follows: When the `FrontDoor` is *open*, the `BackDoor` must be in the opposite state (i.e. *closed*) else there is an anomaly; the specific type of anomaly (i.e error or attack) can be ascertained from the use of the various methods presented in this work. Note, intermediate states such as *closing* or *opening* are not considered in this work because these two states are both considered to be variations on the state *open*.

Therefore, in a home where `D001, D002, D004` are the External Access Node Identifiers, and given **T** = **True** or **open** and **F** = **False** or **closed**, the Truth Table in Table 4.6 illustrates the allowed/permitted and disallowed relationships between the nodes, in this case the doors.

	D001	D002	D004
Normal	T	F	F
	F	T	F
	F	F	T
	F	F	F
Anomaly	T	T	F
	F	T	T
	T	F	T
	T	T	T

Table 4.6: Access Nodes Relationships

A sub-routine in the FDMA as shown in Appendix A is designed to detect and raise an alarm if two or more External Access Nodes are in state OPEN at the same time or within a certain time frame. It also recognises when an External Access Node is *open* for an unusually long time or when one appears to have been forgotten in the state *open*.

4.2.2 Constraints

As part of this research, several types of *Constraints* that can be placed over activities and events to aid AD were identified (Figure 4.6). These are: Presence-, Location-, Time-, Date-, Node- and Network-bound constraints. These constraint were considered relevant and adequate for the purposes of the research in this thesis; more or fewer constraints, as necessary, may be considered in a wider context.

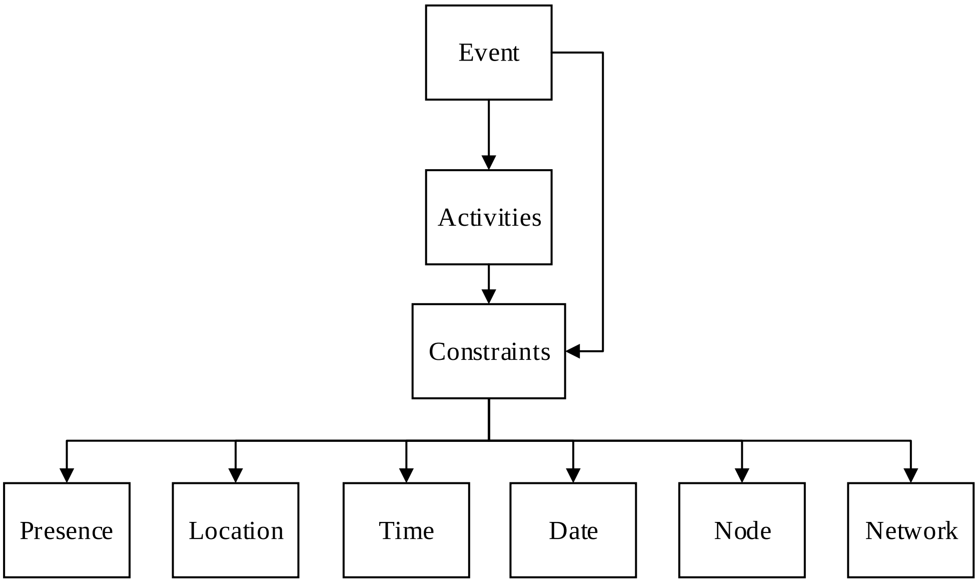

FIGURE 4.6: Relationship between Events, Activities and Constraints

Consider the event *Sleep*: it is not location-bound event but it is Presence- and Time-bound. However, if the *presence* is missing, the anomaly rating is **True Positive**; but if the sleep Time is wrong, the alarm rating may be reduced to **False Positive** or benign, depending on the setting at which the home occupant put them..

Binding activities and events to attributes in this way is useful for detecting anomalies. For instance, presence-bound Activity, though they may be a small pool or sample set of activities, can be used to detect anomalies. Subsubsections 4.2.2.1, 4.2.2.2, and 4.2.2.3 discuss how the constraint n is determined for each n-bound activity or event.

4.2.2.1 To determine n: Illustration One

It is important to determine what the constraint (n) is, per event, because it helps to identify which attributes are the most important and which ones make little to no difference to the outcomes of AD processes and can therefore be ignored so that AD analyses can be completed quicker. n is not a numeric value but represents an attribute or relationship such as network Nx or *presence* Pr. Thus, given the Decision table

Presence (Pr)	Node, State (η, S)	η, S	**Decision**
0	(`FridgeDoor`, Open)	(`M003`, Off)	**Abnormal**
0	(`FridgeDoor`, Closed)	(`M003`, On)	**Abnormal**
0	(`FridgeDoor`, Open)	(`M003`, On)	**Abnormal**
≥ 1	(`FridgeDoor`, Open)	(`M003`, Off)	**Abnormal**

Table 4.7: Scenario Decision Table One

representing the values in the table using numerals will generate the following baseline table:

1	2	3	X
1	4	5	X
1	2	5	X
6	2	3	X

Table 4.8: To determine the Constraint n: Illustration One

where (Pr) is represented by a '1', (`FridgeDoor`, Open) is represented by a '2' and (`M003`, Off) is represented by a '3' and so on. The Decision *Abnormal* is replaced by an **X** (a *normal* will be replaced by a **Y**). Next, the number of occurrences of each value is counted and the one with the highest value is the constraint. The highest occurring value in this table is a '1' i.e. (Presence, 0) with three (3) occurrences. This is followed by the sets (`M003`, On), (`FridgeDoor`, Open) and (`M003`, off) which each have a tally of two (2) occurrences. Therefore the event in this case is *presence*-bound.

To prevent the user from sleeping just anywhere (i.e. zone) in the home, the zones in which it is acceptable for the user to sleep may be explicitly programmed into the FDMA. In addition, the nodes used in these selected zones may also be explicitly identified and restricted to a limited set. This way, when the user falls asleep in one of the acceptable zones, the system will check to see if these nodes are being used and if not, then the event can be described as anomalous.

4.2.2.2 To determine the Constraint n: Illustration Two

Given the event *DriveAroundTown*, the task here is to determine which nodes are important and how important the specific node `Car` is. In order to achieve this, a rule needs to be derived that leads to a **Normal** outcome which involves explicitly the use of the desired node η `Car`. This is for a machine-driven scenario.

Given the scenario Decision Table 4.9.

Node (η)	Presence (Pr)	Time (τ)	Location (Z)	**Decision**
`Car`	≥ 1	Evening	Global	**Normal**
`BathTub`	0	Evening	Bathroom	**Anomaly**
`Car`	0	Evening	Global	**Normal**
`Car`	0	Morning	Global	**Normal**
`Car, Cooker`	0	Afternoon	Global	**Anomaly**[1]

Table 4.9: Scenario Decision Table Two

From Table 4.9 the following baseline table is derived

1	2	3	4	Y
5	6	3	7	X
1	6	3	4	Y
1	6	8	4	Y
1, 9	6	10	4	X

Table 4.10: To determine the Constraint n: Illustration Two

Condensing the 3 Normals (represented by Y in Table 4.10) gives

1	2	3	4	Y
1	6	3	4	Y
1	6	8	4	Y

Table 4.11: To determine the Constraint n: Illustration Two (condensed)

It can be observed from Table 4.11 that node `Car` (represented by figure **1**) and Location Global (represented by figure **4**) are of equal weighting or cardinality of

[1] The nodes are non-proximal.

occurrence. This is followed by Time (3, Evening) and Presence (6, no user). After this comes the Presence (2, *1 or more* users) and Time (9, Morning).

The conclusion from these outputs is that for a machine-driven scenario, *DriveAroundTown* is normal if and only if node = `Car` and Location = GLOBAL. This means that *DriveAroundTown* is node- and Location-bound and that *presence* is not a strict requirement for this event to occur successfully and/or normally.

4.2.2.3 To determine *n*: Illustration Three

For this illustration, the aim is to determine how important the Location attribute is in relation to the *Cook* event. Success for this event requires that the Decision is *Normal*.

Presence (Pr)	Node (η)	Location (Z)	**Decision**
≥ 1	`Bed`	BEDROOM	**Anomaly**
0	0	0	**Anomaly**
≥ 1	`Shoe`	TOILET	**Anomaly**
≥ 1	`Cooker`	KITCHEN	**Normal**

TABLE 4.12: Scenario Decision Table Three

Condensing this gives

1	2	3	X
4	4	4	X
1	5	6	X
1	7	8	Y

TABLE 4.13: To determine the Constraint n: Illustration Three

The greatest weighting in this case[2] = 1 i.e. *Presence* or Pr. If emphasis is placed on the scenarios which lead to a *Normal* output, the presence attribute, even with its weighting, is ignored and the other relationships that lead to a *Normal* outcome are identified. Extracting those relationships therefore give

1	7	8	Y

TABLE 4.14: To determine the Constraint n: Illustration Three (condensed)

[2] Row 2 is an Odd Row representing a non-event or an impossibility; it is therefore ignored.

The other two identified attributes are η (represented by 7) and Z (represented by 8). Therefore, Presence (Pr), Node (η) and Location (Z) are all important to Event *Cook*.

4.3 Methodology

This research makes use of both the Inductive and Deductive Reasoning processes. As explained by David E. Gray in *Doing Research in the Real World*[3], Induction involves the discovery of a "binding principle", derived after the examination of multiple cases with care being taken to avoid basing principles on just one case but instead basing it on an evaluation of multiple cases or instances. During the Inductive process, after the data is collected, it is analysed with the aim being to detect relationships between variables. Deductive Reasoning involves the testing of pre-formed principles, induction involves the development of a principle or principles after analysis of background information (e.g. collected data). For the development of the IoT Digital Forensics Framework (IDFF), an initial framework was proposed. This initial framework was then validated through communication with experts (**interviews**) (Subsection 3.7.1) and **Deductive Reasoning** through scenario analysis (Section 3.10). For the rule induction and analysis, the PRISM rule induction tool available in Weka is used (**Inductive Reasoning**, Section 6.5).

4.4 Chapter Summary

In this Chapter, the concept of a "Happy Home Network" as a security baseline of a SH network was introduced; the attributes that can be used to define this security baseline were identified and defined and the relationships that exist between the zones and nodes in a SH with respect to Anomaly Detection were also identified. The attributes and relationships that were identified in this chapter are applied in Chapters 5 and 6.

[3] Pages 4 and 5 and Figure 2.1 in `https://uk.sagepub.com/sites/default/files/upm-binaries/58626_Gray__Doing_Research_in_the_Real_World.pdf`

Chapter 5

Anomaly Detection Scenarios and Discussions

5.1 Overview

This Chapter discusses the application of the attributes identified in Section 4.2 in Anomaly Detection (AD) tasks. It also discusses how the use of these attributes can reduce False Positives detected. The Chapter introduces the focal Smart Home and Datasets used in this work. The FDMA algorithm and relationships are applied to several real and *realistic* scenarios to illustrate how they can be applied to meet real-life AD needs. The scenarios described range from the relatively simple to the more involved ones.

For the real scenario, real-life Smart Home (SH) data containing non-simulated anomalies is fed through the FDMA algorithm. This was done to detect *presence*-related anomalies. These anomalies were selected because of their significance to the real-life security of especially unsupported elderly people who may choose to remain (in some cases, alone) in their own homes rather than move to public care homes or communities. In addition, other attributes are combined together in different ways to analyse hypothetical scenarios and make decisions about whether they are anomalous or not. Detecting 'presence'-based anomalies is important because the SH occupant might be a defenceless elderly patient and this system will support their security.

5.2 Combining FDMA and Classical Anomaly Detection Methods

This section demonstrates how the FDMA can be used in conjunction with the train-and-test approach to Anomaly Detection (AD) so that a finer-grained approach is taken to detect anomalies that may be mis-classified by typical *train-and-test* AD methods. To demonstrate how AD tools may mis-classify instances and anomalies, consider, for instance, the real-life SH occupant's **total** sleep and toilet durations in hours:minutes:seconds as shown in Table 5.1. Support Vector Machines (SVM), if trained with a dataset like this one, have the capabilities to identify and detect any anomalous sleep/toilet patterns from this table. These methods will accurately identify that the user *typically* sleeps for between 6 to 8 hours per night but did so for over 10 hours on **Day 3**. The data shows that on **Day 21** the home occupant spent over six minutes in the toilet and that on **Day 10** it appears that the home occupant did not got to the toilet at all - and these two instances (**Day 10** and **Day 21**) do not fit in with the typical toilet durations.

Consider the former scenario: from an AD perspective if the threshold set for toilet visit is that each visit must be less than 6 minutes, then this occurrence is anomalous. Generally speaking, there are several possible explanations for this scenario: it is possible that the user went to the toilet more times than usual or, spent too long in the toilet in one visit. However, making use of the attributes identified in Chapter 4, analysing data from other data sources, it might be discovered that the user had a stay-over visitor who also went to the same toilet during the night which made the **total duration** of all visits to the toilet abnormally high.

Considering the case where the total toilet duration was **0 minutes**, location and *presence* data may show that the user went to *a* toilet but went to a different toilet than usual (this can be determined from information about which zones were active during the night). Alternatively, an incontinent SH occupant may have not actually physically changed location to go to the toilet; in this case a reminder system may be more relevant to this user, as opposed to a Digital forensics Incident Response Team.

Therefore, considering other features such as the number of *presences*, active zones, etc. (referred to as *attributes* in this thesis) can provide a much more detailed picture of the goings-on within datasets and thus provide better support for AD tasks whilst avoiding triggering unnecessary alarms. Very importantly, they can help reduce the likelihood of a benign event being classified as an anomaly and triggering a false alarm.

Day	Activity	
	Sleep	Toilet
1	07:54:20	00:02:39
2	07:45:01	00:05:50
3	10:05:32	00:03:09
4	07:23:07	00:05:48
5	08:10:47	00:03:04
6	08:38:58	00:05:01
7	06:33:01	00:00:00
8	09:29:22	00:03:25
9	08:51:24	00:03:47
10	07:23:19	00:00:00
11	07:34:21	00:03:28
12	08:57:21	00:01:58
13	07:19:16	00:03:52
14	09:33:04	00:04:21
15	06:46:21	00:00:00
16	07:57:46	00:04:06
17	06:35:01	00:03:18
18	08:32:29	00:04:29
19	08:06:30	00:03:04
20	07:59:38	00:05:18
21	08:05:29	00:06:23
22	08:09:01	00:03:36

Table 5.1: Sleep and Toilet Activities Table

Consider another scenario in which the *Sleep* event for the day did not start at the normal time or that the Bedroom is empty when a user is supposed to be inside it. In this case, if location data - and other behavioural information - is analysed using the approach proposed in this work, this *absence* will not automatically be flagged as an attack. Instead, based on the analysis, the days when the user performs the *Sleep* event in an anomalous but benign location, such as a LivingRoom will be logged as anomalous but will not raise any alarms. This is because *Sleep* is not a location-bound event and users may sleep wherever they wish in their own home although, for safety reasons, there are only certain locations in which sleep may reasonably happen (see Subsection 4.2.2). The absence of such contextual information may lead the system to trigger an alarm (False Positive) and even call for assistance (healthcare, security, etc.) where none is required thus leading to resource wastage.

Therefore, considering certain factors such as current and previous Locations ($\acute{Z}$ and $-Z$), Time or period of day (τ) alongside user behaviour - and taking into consideration the fact that humans change their minds - can reduce an anomalous event from a True Positive (attack) to an error or even a True Negative i.e a non-anomaly. Additional factors including duration, Core Events, Core Activities, *happens after* and *happens* *before* relationships between activities among others, can also influence the outcomes of AD analysis processes. Therefore, one way to avoid triggering false alarms is to identify and apply the correct factors to events. For example, based on the preceding discussion, for the *Sleep* event, the following attributes and relationships will be required:

only is not enough to classify events as anomalous or normal.

- **Location**: BigBedroom
- **Total number of human occupants i.e. Presences**: = 1
- **Sleep start Period**: 10 p.m. to 12 midnight
- **Core events**: M003 and M008 on
- **Happens before, happens after relationships**: *SleepBegins* happens before *SleepEnds*

If any of these are not as expected then an anomaly can be said to have occurred where an *anomaly* can be attack or an error. More information on combining the attributes and relationships with and activities for the purposes of AD is available in Chapter 6.

Since it is clear that not all anomalies are the same, the following anomaly rank mappings have been drawn up with the highest ranking being the most severe (the True Positive anomaly) and the rank of True Negative corresponding to the least severe.

Box 5.1. Anomaly Rankings

Nothing, Benign[a] **True Negative**

”

System/Algorithm Fault: **False Positive**

”

Error: **False Negative**

”

Attack: **True Positive**

[a] Benign anomalies are those which are excusable or relatively dismissable.

5.3 Datasets

In order to test the FDMA Algorithm at detecting *presence*-related anomalies, a publicly-available real-life SH sensor dataset was sourced and obtained from the Internet and fed into the algorithm. A brief description of the selected dataset is presented in Section 5.3.1. The nodes, zones and proximality relationships that exist between the elements in the Aruba home are also identified. These proximals are first identified based on observation of the node activities however it is possible for the process to be automated and for nodes that share proximality relationships to be automatically identified based on the description of proximality in Subsubsection 4.2.1.3.

In addition, a simulated dataset was used to create several anomalous scenarios for analysis using the AD approach developed in this work. The simulated dataset is described in Section 5.3.2.

There are several other real-life datasets publicly and freely available however the next section describes the Aruba SH dataset and why it was selected for this research. Some of the other real-life datasets include the datasets (available at BoxLab `https://boxlab.wikispaces.com/List+of+Home+Datasets`), Public Datasets for Non-intrusive Appliance Load Monitoring (NIALM) (available at `http://blog.oliverparson.co.uk/2012/06/public-data-sets-for-nialm.html`) and the University of Massachusetts Amherst (UMass) Trace Smart datasets (available at `http://traces.cs.umass.edu/index.php/Smart/Smart`).

The challenge of using datasets are that they might not contain the information required for a particular research need and therefore the data may need to be manipulated with new information introduced or the focus of the research adjusted so that the data available is adequate for requirements. As an example, in this work, a preferable situation would have been for a dataset that includes IP data such as IP addresses, and data from other IoT enabling protocols (see Section 2.2 for examples of IoT enabling protocols). These datasets are freely and readily-available, the way they were generated has been demonstrated in peer-reviewed papers and time and money is saved in not having to build a smart home set up in order to generate data and then validate it before using it. For example, their dataset took years to generate which was not available considering the duration of this research work.

5.3.1 The Aruba SH Dataset

The real-life Aruba Dataset was obtained from the **WSU CASAS** website at `http://casas.wsu.edu/datasets/`. The dataset is from a SH set up with three main types of sensors: the sensors labelled **M001 to M031** are **Motion Sensors**; the sensors labelled **T001 to T005** are **Temperature Sensors** and the sensors labelled **D001 to D004** are **Access nodes** (i.e. Doors). This dataset was selected for several reasons. Firstly, it is a dataset from a real SH. Secondly one of the main focuses of this work is on *presence* detection from motion sensor data and the dataset contains data from motion sensors. This dataset, which is usefully annotated, has over 100,000 data points (represented by *rows*) and relevant features (in the form of *columns*). The features recorded in are time, date, location in the SH and motion sensor reading. These were useful attributes for setting up presence detection scenarios. The presence-detection algorithm presented in this work is targeted at supporting the security of a single-occupant home. This is done in recognition of the fact that more and more people including those who require support and care, will increasingly be enabled by SH smart solutions to live independently at home. With this solution, the presence of an intruder can be identified after-the fact from analysing motion sensor data alongside the time, date data. In addition, the SH is described on the WSU CASAS website as a single-occupant home and that the occupant had visitors during the duration of their stay in the home. This was useful combination because the occupant could be treated as the legitimate occupant while the visitors - the data did not provide any information about *when* the visitors came in or left - could be treated as intruders as part of the physical intrusion detection scenarios. The third reason is because it was discovered, based on a visual observation, informed by domain knowledge, to inherently contain *presence* anomalies[1]. The fourth reason it was selected is because the dataset is also annotated with descriptions of which **activities** were being performed at which times in such a way that the volunteer's activities were recognisable and relatable to the sensor activations and deactivations. This was useful because it added some context to the sensor statuses.

More details on the dataset and a copy of the dataset itself (available at the time of writing) can be found on the WSU CASAS website (Dataset number *17* on the list at `http://casas.wsu.edu/datasets/`). The layout of the WSU CASAS Aruba Home is as shown in Figure 5.1.

5.3.2 Synthetic Dataset

In order to generate a synthetic dataset, anomalies were simulated by making use of the Aruba SH layout but simulating based on the Aruba SH, scenarios that mimic realistic, anomalous SH situations. The scenarios created were designed to influence the attributes identified in Chapter 4 so that the "Happy" state of the selected SH is affected. The scenarios are analysed (Chapters 5 and 6) to demonstrate the applicability of the attributes, relationships and constraints to AD.

[1] For this purposes of this work, more than one *presence* in the SH is seen as a sign of an anomalous intrusion by a human

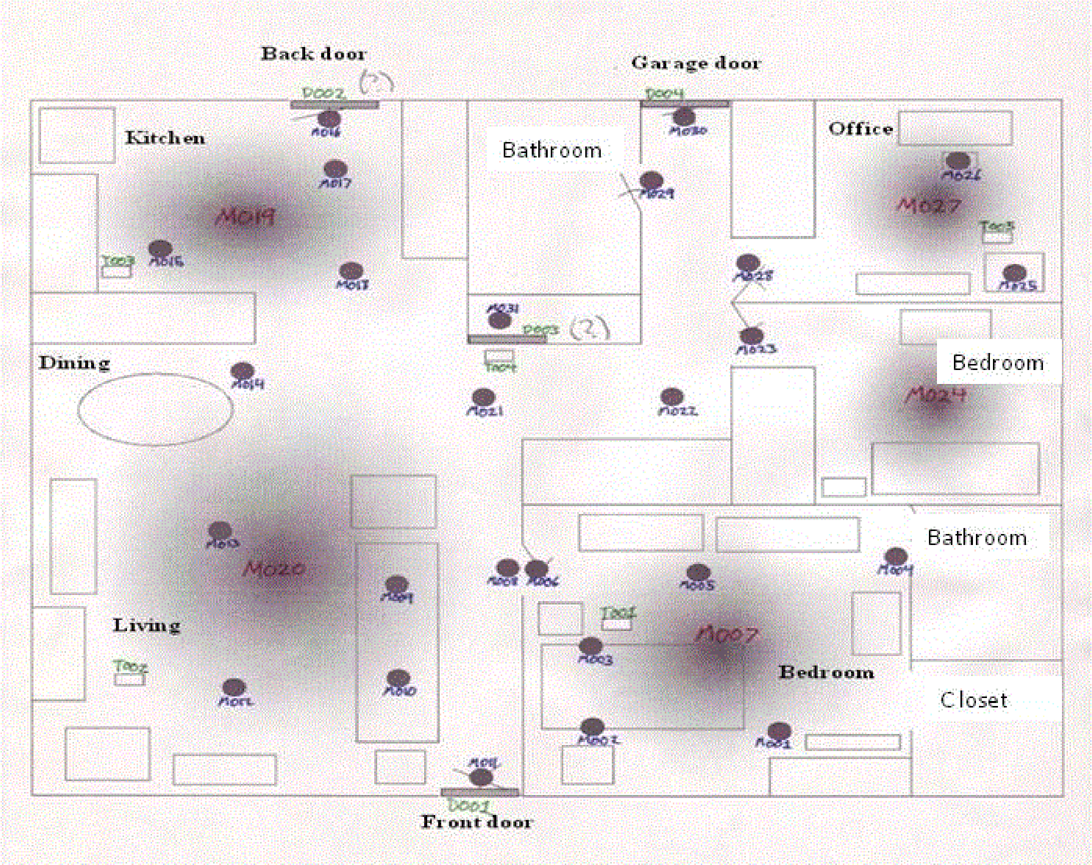

FIGURE 5.1: The WSU CASAS Aruba Smart Home [1]

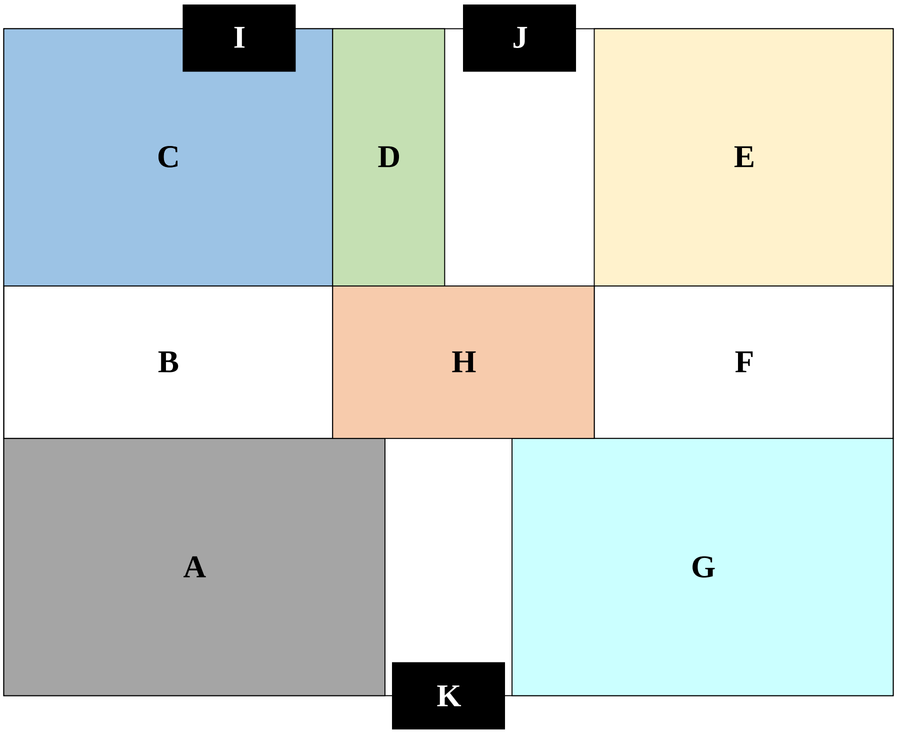

FIGURE 5.2: The WSU CASAS Aruba Smart Home [1] - divided into distinct, (coloured) Zones, A to K

Following the steps in the AD preparation phase, the nodes, zones and proximality relationships are identified and the Tables 5.2 to 5.15 can be drawn up for the Aruba SH. Tables 5.3 and 5.13 show specific motion sensor nodes of the Aruba SH in the first column and their corresponding *inter*-zonal proximals in the second column. A '–' means that the corresponding node has no *inter*-zonal proximals.

Other relationships and attributes can also be identified and a similar set of tables can be generated for any other SH. See Appendix C for the full Proximal Sensors table. In this SH[2], there are 11 zones in total and 34 nodes including 3 external access nodes. These details are fed into the FDMA and are used as part of the decision-making process. The Core of the FDMA Algorithm is available in Appendix A.

Zone	**Zone Mapping**	**Sensors**
A	LivingRoom	M009, M010, M012, M013 and M020
B	Dining	M014
C	Kitchen	M015, M016, M017, M018 and M019
D	SmallBathroom	M029 and M031
E	Office	M025, M026, M027 and M028
F	SmallBedroom	M023 and M024
G	BigBedroom	M001, M002, M003, M004, M005, M006, M007 and M008
H	Hallway	M021 and M022
I	BackDoor	*D002*
J	GarageDoor	M030 and *D004*
K	FrontDoor	*D001*

Table 5.2: Aruba Smart Home Sensors Map

In Tables 5.14 and 5.15 a '1' indicates proximality while a '0' represents non-proximality.

[2] Some imaginary nodes, not listed in this table, are added to the SH later in Chapter 6 in order to simulate hypothetical scenarios for analysis of the AD approach that is introduced in this thesis.

Zone A Sensor	**Proximal**
M009	–
M010	–
M012	–
M013	–
M014	–
M020	**M008 (Zone G)**

Table 5.3: Aruba Smart Home *Living Room* Proximals

Zone B Sensor	**Proximals**
M014	**M018 (Zone C)**

Table 5.4: *Dining* Proximals

Zone C Sensor	**Proximals**
M015	–
M016	–
M017	–
M018	**M014 (Zone B)**
M019	–
D002	–

Table 5.5: *Kitchen* Proximals

Zone D Sensor	**Proximals**
M029	M030 (Zone J)
M031	–

Table 5.6: *Small Bathroom* Proximals

Zone E Sensor	**Proximals**
M025	–
M026	–
M027	–
M028	**M022 (Zone H)**

Table 5.7: *Office* Proximals

Zone F Sensor	**Proximals**
M023	**M022 (Zone H)**
M024	–

Table 5.8: *Small Bedroom* Proximals

The Aruba SH dataset was introduced in this section. The FDMA algorithm is used to analyse a section of this dataset and based on the proximals and location, anomalous presence is detected.

5.4 Scenarios and Discussions

In this section, scenarios are set up (i.e. simulated) to represent different events, activities and node arrangements that may realistically occur within SH and even non-SH. Scenarios are used here because of the lack of access to a real-life SH in which these different permutations of events and activities could have been set up.

Zone G Sensor	Proximals
M001	–
M002	–
M003	–
M004	–
M005	–
M006	–
M007	–
M008	**M020 (Zone A)**

Table 5.9: *Big Bedroom* Proximals

Zone H Sensor	Proximals
M021	**M018 (Zone C)**
M022	**M023 (Zone F)**

Table 5.10: *Hallway* Proximals

Zone I Sensor	Proximals
D002	**M016 (Zone C)**

Table 5.11: *Back Door* Proximals

Zone J Sensor	Proximals
D004	**M030 (Zone J)**

Table 5.12: *Garage Door* Proximals

Zone K Sensor	Proximals
D001("M034")	**M011(Zone K)**

Table 5.13: *Front Door* Proximals

Zones	A	B	C	D	E	F	G	H	I	J	K
A	1	1	0	0	0	0	1	0	0	0	1
B	1	1	1	0	0	0	0	0	0	0	0
C	0	1	1	0	0	0	0	1	0	0	0
D	0	0	0	1	1	0	0	1	0	1	0
E	0	0	0	1	1	1	0	1	0	1	0
F	0	0	0	0	0	1	0	1	0	0	0
G	1	0	0	0	0	0	1	0	0	0	0
H	0	1	1	1	1	1	0	1	0	0	0
I	0	0	1	0	0	0	0	0	1	0	0
J	0	0	0	1	0	0	0	0	0	1	0
K	1	0	0	0	0	0	0	0	0	0	1

Table 5.14: Aruba Smart Home Zone Proximals Table

For the simulated scenarios, an imaginary set of realistic sensors - a `Fridge`, `FridgeDoor`, `Cooker`, `BathTub`, `Bicycle` and a `Car` - are introduced into the SH environment in order to create realistic anomaly detection scenarios within the selected SH. These imaginary nodes were selected based on the likelihood that they would be present in real-life homes including SH. In preparation for a DF investigation, the details from the home are first acquired according to the steps in the SH Network Preparation Stage. Other attributes (e.g. modes, types, distance, peripherals) may be included if applicable.

Sensors	**M001**	**M002**	**M004**	**M006**	**M008**	**M009**	**M010**	**M011**	**M013**	**M020**
M001	**1**	**1**	**1**	0	0	0	0	0	0	0
M002	**1**	**1**	0	0	0	0	0	0	0	0
M004	0	0	**1**	0	0	0	0	0	0	0
M006	0	0	0	**1**	0	0	0	0	0	0
M008	0	0	0	0	**1**	0	0	**1**	0	**1**
M009	0	0	0	0	**1**	**1**	0	0	0	**1**
M010	0	0	0	0	0	0	**1**	0	0	0
M011	0	0	0	0	**1**	0	0	**1**	0	0
M013	0	0	0	0	0	0	0	0	**1**	0
M020	0	0	0	0	**1**	**1**	0	0	**1**	**1**

Table 5.15: Aruba Smart Home Sensor Proximals (partial Table)

5.4.1 Zone Z, Node η, Mode μ, Proximals ρ

A location-based anomaly is said to have occurred if a node changes location abnormally and is discovered to be in a wrong location. The *Mode (μ)* attribute of a node and its proximals can be used to detect this abnormal change in location. A node with a F-F mode should not, as part of its everyday function, change location unless its circumstances change e.g. a door being removed for repairs or replacement.

This randomisation was achieved manually. The aim of this scenario was to demonstrate the use of location and mode attributes to detect anomalies. In order to detect the *anomalous location change* for nodes with μ = F-F, the normal and current proximals of the node are compared over time. In addition, the normal location and current location at different times of the day are compared. The scenarios in Table 5.16 are used as a basis for a discussion on how changes in location can be detected using these attributes. The headings of the attribute columns are from the attributes introduced in Subsection 4.2.1.

Scenario	Period τ	Node η	Mode μ	Proximals ρ	N_ρ	Zone Z	Previous Zone $-Z$	**Decision**
1	Morning	`Bathtub`	F-F	`ToiletDoor`	1	Bathroom	Bathroom	**Normal**
2	Afternoon	`Bathtub`	F-F	`ToiletDoor` & `Bicycle`	2	Bathroom	Bathroom	*Inconclusive*
3	Night	`Bathtub`	F-F	`ToiletDoor`	1	Bathroom	Bathroom	**Normal**
4	Night	`Bathtub`	F-F	`Fridge`	1	Bathroom	Bathroom	**Abnormal**
5	Night	`Bathtub`	F-F	`ToiletDoor`	1	Kitchen	Bathroom	**Anomaly**
6	Morning	`Cooker`	F-F	`ToiletDoor`	1	Kitchen	Bathroom	**Anomaly**
7	Afternoon	`Cooker`	F-F	`ToiletDoor`	1	Kitchen	Bathroom	**Anomaly**

Table 5.16: Fixed-Fixed Mode Table of Scenarios

The scenarios are analysed individually and the decisions reached in each case is explained next.

Scenario 1 is **normal**. This is because in the morning, the active node is the `Bathtub`, which has a a F-F mode. The system detects that the proximal of the `Bathtub` is the `ToiletDoor` and the zone the user is in is the Bathroom. In addition, the previous (-Z) and current (Z) zones are both the same, and for a node with $\mu =$ F-F, this is as required.

Scenario 2 can be either **normal** or **anomalous**. The scenario can be normal because even though a `Bicycle` is proximal to the `BathTub`, a location change anomaly with respect to the `BathTub` has not occurred. More information such as the on-going activity or event will be required to determine which node is being given priority in terms of consideration. The application of events and activities to Anomaly Detection is discussed in Chapter 6.

On the other hand assuming that the user does not normally store the `Bicycle` in the Bathroom, an anomaly can be inferred based on user preferences or, even, domain knowledge.

Scenario 3 is **normal** for reasons similar to Scenario 1. The difference between the two scenarios is that the first one occurs in the morning while the second one happens in the afternoon. This result implies that for location-based anomalies (as explicitly defined in the Definition of Location-based anomaly in Subsection 5.4.1) to be detected, the time of day does not play a significant part.

The decision in Scenario 4 is **anomalous** because of the fact that even though the node (i.e. `BathTub`) is in the Bathroom and does not change location (Z is the same as -Z) for the duration of the scenario, its proximal is a `Fridge` which is in a different, non-proximal, zone (see Figure 5.1 and Figure 6.2).

Consider Scenario 5. This scenario is **anomalous** because even though the other relationships are normal, the previous and current zones are different, indicating that the node which should have a F-F mode has changed location.

Bearing in mind that Scenario 6 can represent any on-going activity or event (none is explicitly specified here), if the active node `Cooker` is taken as the cardinal attribute, the scenario is **anomalous** because the `Cooker` is in the Kitchen (Figure 6.2), the `Bathtub` is in the Bathroom and the two zones are non-proximals (see Table 5.14).

Scenario 7 accurately resolves to an **anomaly** because the only attribute by which it differs from Scenario 6 (i.e. the Period τ) is inconsequential to the overall decision in the scenario being analysed.

A similar approach to the preceding (i.e. using scenario analyses) is used in the rest of this Chapter and in Chapter 6 to explain how the individual Decisions for each scenario is reached.

5.4.2 *Presence*-bound Anomalies

A *presence*-bound anomaly occurs when a presence is detected where one should not be or no presence is detected where there should be at least one. Some example scenarios are used to illustrate the use of *Presence* (Pr) in Anomaly Detection.

5.4.2.1 Presence Pr, Network Nx, Peripheral Per

This section involves demonstrating the use of the Pr, Nx and the Per attributes in anomaly detection. Consider a scenario where a user receives a report that their `FridgeDoor` is *open*. Assume in this scenario that the user lives alone and is away from the house. Combining these attributes, several potential scenarios can be drawn up which capture this situation.

Scenario	Pr	Nx	Per	Node	Decision
1	≥ 1	Up-Normal	Kitchen	Active	**Normal**
2	≥ 1	Up-High	KITCHEN	Active	**Anomaly**
3	≥ 1	Down	KITCHEN	Active	**Anomaly**
4	≥ 1	Up-Normal	Kitchen	Active	**Normal**
5	0	Up-Normal	KITCHEN	Active	**Normal**
6	0	Up-High	KITCHEN	Active	**Anomaly**
7	0	Down	KITCHEN	Active	**Anomaly**
8	0	Down	KITCHEN	Inactive	**Normal**
9	≥ 1	Up-Normal	BIGBEDROOM	Active	**Anomaly**
10	≥ 1	Up-Normal	na	Active	**Anomaly**
11	0	Up-Normal	na	Active	**Anomaly**

TABLE 5.17: Presence and Network Attributes Scenarios Table

Box 5.2. PRESENCE Pr, NETWORK Nx, PERIPHERAL Per

If (Pr, 0) & (Nx[a] , Down) & (Per, Inactive) ⇒ (No Anomaly)

[a] Inbound-Outbound, Remote Connections, etc.

This may indicate that the report is potentially the sign of an anomalous remote access attack involving the `FridgeDoor` which is being made to appear open even if it is physically shut and no one is home. However, if the logs show that the `FridgeDoor` was left open before the user left the SH, then an error (which was missed by whichever SH monitoring system was in place) will be the most likely explanation of the situation.

5.4.2.2 Presence Pr, Network Nx

The attributes network, *presence* and location can be combined to detect anomalies. To demonstrate this, consider the scenarios in Table 5.18. If the scenarios each begin with the report that the `FridgeDoor` is open in a home that is empty (i.e. $Pr = 0$). Given: `FridgeDoor` is open and `Fridge` zone is Kitchen

Scenario	Pr	Nx	Z	`Fridge`	`FridgeDoor`	**Decision**
1	≥ 1	Up-Normal	Kitchen	Active	open	**Normal**
2	≥ 1	Up-High	Kitchen	Active	open	**Anomaly**
3	≥ 1	Down	Kitchen	Active	open	**Anomaly**
4	≥ 1	Up-Normal	Kitchen	Active	closed	**Normal**
5	0	Up-Normal	Kitchen	Active	open	**Normal**
6	0	Up-High	Kitchen	Active	open	**Anomaly**
7	0	Down	Kitchen	Active	open	**Anomaly**
8	0	Down	Kitchen	Inactive	closed	**Normal**
9	≥ 1	Up-Normal	BigBedroom	Active	open	**Anomaly**
10	≥ 1	Up-Normal	na	Active	open	**Anomaly**
11	0	Up-Normal	na	Active	open	**Anomaly**

Table 5.18: Presence and Network Attributes Scenarios Table

Rule Induction is discussed in Chapter 6 but an example rule which is specific to this particular Aruba SH scenario and which considers the attributes *presence* and location is given thus:

Box 5.3. Presence Pr, Location Z

If (Pr, ≥ 1) & (`FridgeDoor`, Open) & (`FridgeDoor`, Shut) & (Z, Kitchen) $\Rightarrow$ (No Anomaly)

The rule which is drawn from domain knowledge based on the data in the table states that there is at least one person in the KITCHEN, the peripheral was opened then shut.

In this example, considering that the `FridgeDoor` can be shut automatically, if that part of the rule is removed, the decision will still be the same.

5.4.2.3 Presence Pr, Obstacle Factor δ

Given a scenario where two motion sensors `M018` and `M031` are active at the same time. To determine if this is an anomalous situation i.e. is there more than one presence in the home, check the following:

- Are the two nodes proximal? In this case the two nodes are in two different, non-proximal zones (zones C and D in Figure 5.2). Also, from the data in Appendix C the two nodes are not proximal.
- Are they edge nodes? The two nodes in this scenario are not because there is a significant, solid obstacle - a through wall - between them.

Therefore, if two nodes such as `M018` and `M031` are both on at the same time, even if they are only separated by a distance that means that they can be proximal nodes, as long as they are also separated by a through wall or a significant, solid obstacle. the distance between them will have to be multiplied by the obstacle factor δ in order to obtain the real distance between them.

5.4.2.4 Presence Pr, Proximals ρ

For this scenario - which makes use of proximals to detect anomalies in physical presence - it was not necessary to simulate any anomaly or make up any scenarios in the Aruba WSU CASAS Smart Home dataset which was used as experimental data to test the *presence* detection function of the FDMA algorithm.

As mentioned before, an active motion sensor in a zone implies a *presence* is in that zone. The data from motion sensors around the Aruba home were used for this analysis i.e. the detection of at least one anomalous *presence* based on the proximality rules.

First of all the data is analysed using the SVM algorithm and the result obtained from that analysis is as shown in Listing One: Anomalous Presence Detection output - SVM. The SVM approach is a *supervised* method. this means labelled data is used to train the algorithm on what is normal or anomalous. A Clssifier is built using this training data after which fresh unlabelled test data is analysed using the same Classifier for it to determine which of the occurrences (in this case, data rows) is anomalous.

```
STATE  SENSOR    HOUR    MINUTE   SECOND  ANOMALOUS?
1        26       15       47       48        0
0        22       15       47       49        0
0        28       15       47       50        0
1        27       15       47       52        0
0        27       15       47       54        0
0        26       15       47       58        0
1        26       15       48       0         0
0        26       15       48       2         0
1        26       15       48       2         0
0        26       15       48       4         0
1        26       15       48       8         0
1        27       15       48       8         0
0        26       15       48       9         0
0        27       15       48       9         0
1        27       15       48       10        0
1        26       15       48       11        0
0        26       15       48       12        0
0        27       15       48       12        0
1        26       15       48       13        0
0        26       15       48       15        0
1        26       15       48       17        0
0        26       15       48       19        0
1        26       15       48       27        0
0        26       15       48       29        0
1        26       15       48       41        0
0        26       15       48       43        0
1        26       15       48       50        0
0        26       15       48       52        0
1        26       15       49       11        0
0        26       15       49       13        0
1        26       15       51       27        0
1        27       15       51       27        0
0        26       15       51       29        0
0        27       15       51       29        0
1        26       15       51       30        0
1        27       15       51       32        0
0        27       15       51       33        0
0        26       15       51       34        0
1        26       15       56       3         0
0        26       15       56       4         0
1        26       15       56       46        0
0        26       15       56       47        0
1        26       15       58       19        0
0        26       15       58       21        0
1        26       15       59       32        0
0        26       15       59       33        0
1        26       16       0        12        0
0        26       16       0        14        0
1        26       16       6        47        0
0        26       16       6        50        0
```

```
1       26      16      7       29      0
0       26      16      7       31      0
1       26      16      8       32      0
1       27      16      8       33      0
0       26      16      8       35      0
0       27      16      8       35      0
1       26      16      8       58      0
0       26      16      9       0       0
1       27      16      9       36      0
1       26      16      9       37      0
0       27      16      9       39      0
0       26      16      9       44      0
1       26      16      9       46      0
0       26      16      9       48      0
1       26      16      10      23      0
1       28      16      10      29      0
1       18      16      10      34      1
1       21      16      10      34      1
0       26      16      10      34      0
0       28      16      10      34      0
1       19      16      10      35      1
1       16      16      10      36      1
0       21      16      10      36      0
0       18      16      10      37      0
0       19      16      10      37      0
1       19      16      10      38      1
0       19      16      10      39      0
0       16      16      10      40      0
```

Listing One: Anomalous Presence Detection output (SVM-derived)

A ‘1’ in the “ANOMALOUS?” column means that the particular scenario is anomalous. The anomalies detected are based on the proximality relationship. However, the need for explicit labelling makes this method cumbersome. On the other hand, for each SH that applies the FDMA, a set of relationship tables can be built - which are updated - such that if any of the relationships are breached (e.g. a relationship based on proximality) an alarm can be triggered.

Therefore, the anomalies detected are correct however the challenge is how to efficiently and explicitly label data when it runs into the tens of thousands of instances. Bearing in mind that identifying proximality is an arduous task that requires careful scrutiny of the node firings and then labelling the ones that ire according to a pattern a reaction.

A section of the dataset was fed into the *presence* detection engine of the FDMA algorithm and the results obtained are shown in Listing Two: Anomalous Presence Detection output - FDMA. As mentioned, the method does not involve using the classical AD approach of *training* and then *testing* because the rules are explicit unchanging rules - i.e. not predominantly based on user behaviours which can understandably change -

but on the attributes of nodes and zones that can be preprogrammed to behave in particular ways.

```
S/No. -STATE: CURRENT LOCATION CURRENT SENSOR = NODEID DECISION
10 -Active: Office  current sensor  = 27     ''

11 -Inactive: Office  current sensor  = 27     ''

12 -Active: Office  current sensor  = 27     ''

13 -Inactive: Office  current sensor  = 27     ''

14 -Active: Office  current sensor  = 27     ''

15 -Inactive: Office  current sensor  = 27     ''

16 -Active: Office  current sensor  = 28     ''

17 -Inactive: Office  current sensor  = 26     ''

18 -Inactive: Office  current sensor  = 28     ''

19 -Active: Office  current sensor  = 26     ''

20 -Active: Office  current sensor  = 27     ''

21 -Active: Corridor  current sensor  = 22 : Warning!  current sensor  = 22 last
    active sensor = 27 Previous Active Location was Office  And current active
    Location is Corridor  0 seconds travel time : Non-Proximal!!!!!!     ''

22 -Active: Corridor  current sensor  = 21     ''

23 -Inactive: Corridor  current sensor  = 22     ''

24 -Inactive: Office  current sensor  = 26     ''

25 -Inactive: Office  current sensor  = 27     ''

26 -Active: Kitchen  current sensor  = 18     ''

27 -Active: Office  current sensor  = 25 : Warning!  current sensor  = 25 last
    active sensor = 18 Previous Active Location was Kitchen  And current active
    Location is Office  0 seconds travel time : Non-Proximal!!!!!!     ''

28 -Inactive: Corridor  current sensor  = 21     ''

29 -Active: Office  current sensor  = 26     ''

30 -Inactive: Office  current sensor  = 25     ''

31 -Active: Kitchen  current sensor  = 19 : Warning!  current sensor  = 19 last
    active sensor = 26 Previous Active Location was Office  And current active
    Location is Kitchen  0 seconds travel time : Non-Proximal!!!!!!     ''
```

Listing Two: Anomalous Presence Detection output (FDMA-derived)

The *Active* location is the same thing as the *current* location ($\acute{Z}$).

The FDMA detects anomalies using a combination of the Proximality and (Active) Zone attributes where *presence* is determined by a zone being active. The FDMA (Core) algorithm in this work currently only detects *Presence* and external access nodes anomalies without any labelling (*unsupervised*).

One shortcoming of the method is its inability to determine if there are more than two (2) *presences* in a space at a given time. Furthermore, the system can also be enhanced so that it can be used to determine if the *presence* detected is that of a pet or of a human being. The ability of the system to tell the difference between a human and a pet will be useful in avoiding any false alarms in a home where the home-owner has a pet where the pet can be mistaken for an intruder. One suggestion for how this can be done is proposed under further work in Chapter 7, Subsection 7.4.2.

5.4.3 Node η, Peripherals Per

When two nodes are peripheral to each other, they are both activated when stimulus is applied to only one of them. Nodes that share peripheral relationships include fridges and fridge doors, cookers and oven doors.

Nodes and their peripherals can be arranged in such a way that, in response to a common stimuli, one is activated *before* the other so that any time this pattern is not adhered to, an anomaly can be inferred[3].

Between nodes and their peripherals the following apply: they have the **same location** (Z) and respond to stimuli applied to **either one of them** even if not at precisely the same time, are active for approximately the same duration, and they can have common **proximals**. A peripheral relationship is the type of relationship that exists between the `Fridge` and the `FridgeDoor` in Figure 6.2 and it influences the **Decisions** reached in the Scenarios in Table 6.4.

[3] In some cases it is enough for the peripheral to be active (at all) *while* the node itself is active; this is because of how quickly the triggering happens such that it is not easy to observe which one happens first therefore the happens *before* relationship can be ignored as long as they are both active.

5.4.4 External Access Nodes η

For the purposes of this work, it is assumed that in a single-occupant home, over a given period the following definitions describe External Access node Anomalies:

An external access node anomaly is said to have occurred if two or more external access nodes (or doors which leas into and out of the SH) are open at the same time. In addition, irrespective of what type of home is being considered (i.e. single- or multi-occupant) an external access node anomaly is said to have occurred when one or more door is open for too long. These rules apply even if the nodes are proximal. For this scenario, a three-door home is considered with the relationship between the doors as shown in Table 4.6. The data excerpt presented in Table 5.19, taken from the overall Aruba SH dataset, is used as a basis for the discussion of this anomaly.

Date	Time	Sensor	State
17/06/2011	12:29:28	D004	closed
17/06/2011	12:29:58	D004	open
17/06/2011	12:30:03	D004	closed
17/06/2011	12:30:22	D004	open
17/06/2011	12:30:26	D004	closed
17/06/2011	14:05:28	D004	open
17/06/2011	14:05:33	D004	closed
17/06/2011	18:09:48	D002	open
17/06/2011	20:35:09	D004	open
17/06/2011	20:36:01	D004	closed
17/06/2011	20:36:13	D002	closed
17/06/2011	21:44:58	D004	open
17/06/2011	21:45:06	D004	closed
17/06/2011	21:45:36	D004	open
17/06/2011	21:45:41	D004	closed
18/06/2011	10:00:00	D002	open
18/06/2011	10:03:25	D001	open

Table 5.19: External Access Nodes Dataset (Excerpt)

From the highlighted section it is clear that `D002` is open for too long - approximately 1½hours which violates one of the external access node rules. In addition, during the same period `D002` and `D004` are both open at the same time. This also violates one of the external access nodes rule. In order to detect this anomaly, the system is trained on how long the doors are *typically* or *normally* open for and these values stored as normal baselines and comparisons are made every time the door is opened so that abnormal durations are detected.

The Truth table in Table 4.6 provides the rules that can be applied for AD in External Access Nodes. Observe that this applies to a single occupant home with three External Access Nodes.

5.5 Chapter Summary

In this Chapter the attributes identified in Chapter 4 were applied in the analysis and discussion of one real and several hypothetical but realistic SH scenarios in order to illustrate how these attributes can be applied to Anomaly Detection (AD).

In addition, the Forensics Decision Making Algorithm (FDMA) was used to analyse a section of the Aruba Smart Home (SH) dataset and - making use of the location and proximality attributes - it successfully detected more than one presence in the SH during the selected period. Even though in *this* SH more than one presence is a sign of a legitimate visitor, for the sake of simplicity, more than one presence is described as a sign of an intrusion in this work. Non-invasive presence detection as demonstrated in this Chapter is useful for supporting the security needs of SH occupants some of whom will be elderly who are enabled to continue to live in their own homes because of the support provided by SH tools and services but who may then become targets of physical intruder attacks.

The scenario analyses carried out in this chapter as well as the presence detection output of the FDMA demonstrate the applicability of the methodology to real-life SH anomaly detection situations.

The rules derived in this Chapter can be adjusted to fit into any smart environment or space which has zones and nodes. The steps to follow in setting up the smart space for AD will be that same one described in Chapter 4. The zones and the nodes they contain as well as the attributes of each of them should first be identified. When the space is being set up, the optimal security information about the attributes of the nodes and zones should be fed into the FEMS or any alternative security system as required by the SH occupant. These base values will form the first set of rules. Similar rules as introduced in this Chapter and Chapter 6 can then be harvested by the system as the home is inhabited and users perform their daily activities, with the home adjusting according to their changing patterns.

It is acknowledged that a limitation in this work is the use of scenarios which, though realistic, the test of the system can be more rigorously carried out using real people carrying out real activities and thereby generating real data which can then be analysed using the attributes introduced in Chapter 4. Another limitation is the fact that the Aruba dataset which was used in this research is from a single-occupant home and that, in reality, even though Smart Technologies in the Internet of Things (IoT) enable elderly and even invalid people to live independently, some of them will have visitors and some may also live in multiple-occupant homes. It will be essential therefore to carry out a similar analysis of the security situation in multi-occupant homes using the attributes identified in this work.

In Chapter 6 the FDMA and *Core activities* concept are combined and applied in the analysis of some additional realistic anomalous scenarios.

Chapter 6

Anomaly Detection Scenarios and Discussions (II)

6.1 Overview

In Chapter 5, several scenarios involving SH nodes and zones were analysed and these analyses were used to demonstrate how the FDMA methodology applies to Anomaly Detection. In this Chapter, several scenarios are used to illustrate how a combination of "Core Activities" [10], Core Events and the FDMA can be applied to Anomaly Detection (AD) tasks. A discussion section covers the merits of the approach. Just as in Chapter 5, the formalisms introduced in Chapter 4 are used in this Chapter. For example, *Events* and *Activities* are written in italicised *red coloured font*.

6.2 Core Activities

Every event is made up of activities (this definition is adapted from [10]). The FDMA can be trained with SH occupants' regular events such that every event (or or Activity of Daily Living (ADL)) is constrained by certain attributes. For example, for the purposes of anomaly detection, constraints can be placed on activities and events such that each one of them must be performed in particular locations using particular nodes (the use of particular nodes for particular activities is investigated and discussed in [43]).

A Smart Home (SH) occupant may carry out a particular event in several different ways. For the purposes of this work each of the different ways in which an event can be completed is known as an **Activity Sequence (ASeq)**. An Activity Sequence (ASeq) can also therefore be defined as one of the different sets of activity steps that can be taken to complete an event. Activity Sequences are described as making up an **Activity Class**. An **Activity Class** - made up of a series of ordered steps also known as Activity Sequences (ASeq) - is a potential way by which tasks such as Activities of Daily Living (ADL) can be accomplished or completed. Each of the Activity Sequences that make up an event has one or more **core activity(-ies)** which, if missing from the sequence, might be indicative of an anomaly.

The relationship between Events, Activity Classes and Activity Sequences is illustrated in Figure 6.1.

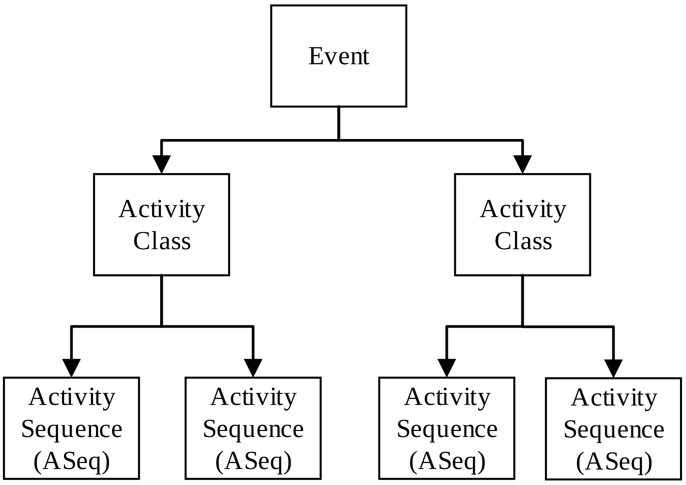

FIGURE 6.1: Relationship between Events, Activities Classes and Activity Sequences

By way of an example, take the event *OrderItemsUsingFridge*. This event has 3 potential Activity Classes:

- Local Physical Ordering (by Human)
- Remote Ordering (by Human)
- Self-replenishment (by Fridge)

In addition to core activities as a factor of AD, the sequence or *order* in which activities and events happen (and the number of times certain types of activities occur during certain events) play a key role in detecting anomalies in user activities. This gives rise to the "**happens after**" [10] and **happens before** relationships descriptions. For instance, consider the Activity Sequences list in Table 6.2: observe that activity a_9 always occurs after a_8. Activity a_9 therefore can be described as having a "happens after" relationship with a_8. The "happens" relationship order does not have to be in a tightly restrictive order i.e. activities must occur in a general pattern but they do not have to occur *immediately before* or *immediately after* each other for the approach described in this Chapter to apply.

i.e.

$$\bullet \text{ precedent activity} \longrightarrow \text{(core) activity} \longrightarrow \text{subsequent activity} \bullet$$

6.2.1 Core activities Rule

In order to use core activities for AD the following steps are followed: whenever an event is triggered, the FDMA checks for

- The Core activity as the selected Activity Sequence happens. At the end of the ASeq, if the core activity is missing, the decision is that the event was not completed successfully;
- The order of the activities for those with a *happens After* or *happens Before* relationship to ensure they occur in the right order.

Then, for every activity, the system checks for

- The nodes, zones and other attributes that should be involved so that if any of these attributes are missing or not in their normal state, an anomaly will be flagged;
- event and activity constraints.

The core activities rule can be summarised thus:

Box 6.1. Core activities Rule

Given event E, with activities space

$$\{a_1, a_2, a_3,...,a_n\}$$

and core activity (for example) a_k. For event E to be successful, Core Activity a_k

1. must occur AND
2. must occur in E AND,
3. must occur according to a *happens after* and/or *happens before* order with respect to the other activities AND,
4. in some instances, must occur the correct *number of times* (or frequency) in the Activity Sequence being followed.

A related concept - which is not discussed in detail in this work - is the concept of **Core events**. A Core event is one which must normally and typically occur, within a measurable time-frame. An example time-frame is a 24-hour period. Example daily core events include *breathing* and *brushing*. Consider a frail elderly dementia patient who lives alone in a Smart Home (SH) and takes daily medication. The Core events of *GetUp* during the morning period of every day and *TakePills* at stipulated times are important. Core events can be time-constrained so that they must occur at certain predefined times failing which an alert can be triggered. Consider too, a child suffering from teeth issues; her daily Core events may be *BrushTeethMorning* and *BrushTeethEvening*. Core activities and core events are useful for designing and providing tailored lifestyle services to individual users. This is because a SH security system can be set up to check for the occurrence of core event and if that event is missing or a core activity in that event is missing, the system can issue a reminder or even trigger an alarm depending on how the system is set up.

6.3 Application of Core Activities and FDMA in Anomaly Detection

In this section the *core activities*, *relationships* and *constraints* are combined to analyse several scenarios for anomalies. Take as an example the Core Activity *EntersToilet* as part of the Event *GoToToilet*. During this event, the user must *physically* be inside the BigBedroom (Figure 5.1 and Table 5.2). This means the *Presence* requirement for this Activity is Presence ≥ 1 and the location requirement is zone $=$ BigBedroom. Therefore, if activity *EntersToilet* happens whilst the BigBedroom is inactive (i.e. empty), it can be correctly concluded that the situation is anomalous.

Box 6.2. Core activities + FDMA Rule

If (Core Events, Yes) & (Core Activities, Yes) & (“Happy Home Network” Attributes, Normal) $\Rightarrow$ (Normal)

To demonstrate this combined approach four (4) ADLs were selected, each with 3 different ways of performing them: **normal-human**, **normal-machine** and **abnormal** ways. A suggested list of different activities that make up these events is presented in the Activities List in Table 6.1 in Section 6.3.1. The Activity Sequences and Core Activities for each case are as shown in the respective Scenario Sections. Lastly, they are combined with the constraints identified in this work to detect any anomalies in the “Home Happiness” status.

Selected ADLs (or Events) that are considered

ADL$_1$: Replenish Fridge Stock - *ReplenishFridgeStock*

ADL$_2$: Update Fridge Software - *Update Fridgeware*

ADL$_3$: Go to TOILET -*GoToToilet*

ADL$_4$: Drive around town -*DriveAroundTown*

The approach is validated by a detailed analysis of each scenario and the decisions reached.

As previously explained, each ADL (or event) is made up of a set of **Activity Sequences** (**ASeq**) and each ADL has one or more core activity which, if missing from the sequence, may be indicative of an anomaly. In addition to these, for them to be completed successfully and *normally*, each event and activity requires a set of nodes, zones, time (or period) of day and other factors. **Constraints** can also define the normality of an activity (or event). In some cases, constraints are relatively flexible (e.g. a home occupant may fall asleep in their LivingRoom instead of in the normal location of the BigBedroom - based on Figure 5.1 - without triggering an alarm), whilst in other cases they may not be flexible (e.g. a home occupant may not cook anywhere else but in the Kitchen thus implying that *Cooking* can be described as a Location-bound event).

6.3.1 Activities List

Following from the discussions in Chapter 3, during Stage 3 of the Home Preparation Phase of the Internet of Things Digital Forensics Framework, the SH system is (initially) explicitly trained on what is *normal* by the user after which the system subsequently learns and updates itself on what is normal - where *normal* represents the home status that is acceptable to the user. The user can also manually update the system and make adjustments to this ‘normal’ as necessary. Part of what the system learns is which activities make up events or ADLs. Example activities are as shown in Table 6.1. This table gives the list of the activities that were used to formulate the Activity Sequences used in the different scenarios being considered in this Chapter.

For each of the scenarios presented, the aim in the rest of this Chapter is to use attributes and relationships (described in Section 4.2.1), constraints (described Section 4.2.2) and Activity Sequences and core activities (described in Section 6.2) in different iterations - and not necessarily all at once - to analyse the decisions reached in the different scenarios as presented. Just as in Chapter 5, each scenario is represented by a row in the scenarios tables. The Decisions reached in the scenarios considered are evaluated and discussed.

Label	Activity	Label	Activity
$\mathbf{a}_{1}$	Go to Kitchen[1]	$\mathbf{a}_{20}$	Accept Terms and Conditions[2]
$\mathbf{a}_{2}$	Go to `Fridge`	$\mathbf{a}_{21}$	User enters Bedroom
$\mathbf{a}_{3}$	`M003` On	$\mathbf{a}_{22}$	User opens `ToiletDoor`
$\mathbf{a}_{4}$	Log in to `Fridge` interface	$\mathbf{a}_{23}$	User enters Bathroom
$\mathbf{a}_{5}$	Look up `Fridge` Contents	$\mathbf{a}_{24}$	User closes `ToiletDoor`
$\mathbf{a}_{6}$	Generate Shopping List	$\mathbf{a}_{25}$	User leaves Bathroom
$\mathbf{a}_{7}$	Accept pre-prepared Shopping List	$\mathbf{a}_{26}$	User leaves Bedroom
$\mathbf{a}_{8}$	Select Supplier[3]	$\mathbf{a}_{27}$	Exit Home
$\mathbf{a}_{9}$	Place Order	$\mathbf{a}_{28}$	Enter `Car`
$\mathbf{a}_{10}$	Accept Receipt	$\mathbf{a}_{29}$	Authenticate[4]
a_{11}	Log out	$\mathbf{a}_{30}$	Start `Car`
$\mathbf{a}_{12}$	Open `FridgeDoor`	$\mathbf{a}_{31}$	Accept suggested route(s)
$\mathbf{a}_{13}$	Place items in `Fridge`	$\mathbf{a}_{32}$	Decline suggested route(s)
$\mathbf{a}_{14}$	Update Stock List	$\mathbf{a}_{33}$	Leave Home
$\mathbf{a}_{15}$	Close `FridgeDoor`	$\mathbf{a}_{34}$	Drive Around Town
$\mathbf{a}_{16}$	Leave Kitchen	$\mathbf{a}_{35}$	Return Home
$\mathbf{a}_{17}$	Click "Update Fridgeware"	$\mathbf{a}_{36}$	De-authenticate
$\mathbf{a}_{18}$	Select Update Source: Manufacturer website		
$\mathbf{a}_{19}$	Select Update Source: USB		

Table 6.1: Hypothetical Activities List

According to [90], ADLs are either Basic ADLs (e.g. "bathing, dressing and eating") or Instrumental ADLs (e.g. "managing money or medication"). Basic ADLs, according to the authors, involve "less complex, implicitly learned activities" while Instrumental ADLs "involve more cognitively demanding tasks". This Chapter uses the generic expression ADL to cover both types because it is not necPage essary to highlight the difference between them in this work.

6.3.2 Activity of Daily Living 1 (ADL 1): Replenish Fridge Stock

Several Activity Sequences (ASeq) that exemplify how the groceries in a `Fridge` in a SH can be replenished are shown in Table 6.2 and are discussed in the scenarios in this section and its subsections. In these scenarios it is assumed that the `Fridge` is located in the Kitchen. As with all the scenarios in this Chapter, the *ReplenishFridgeStock*

[1] When motion is detected in a zone, that zone is described as being *active*.

[2] Accepting the Terms and Conditions implicitly begins the Fridgeware Update sequence.

[3] In a_8 the user identifies the cheapest supplier from a list of websites.

[4] Authentication includes identification, authentication, authorisation, and accounting.

scenarios occur in the SH layout[5] in Figure 5.1 and (hypothetical) KITCHEN layout in Figure 6.2.

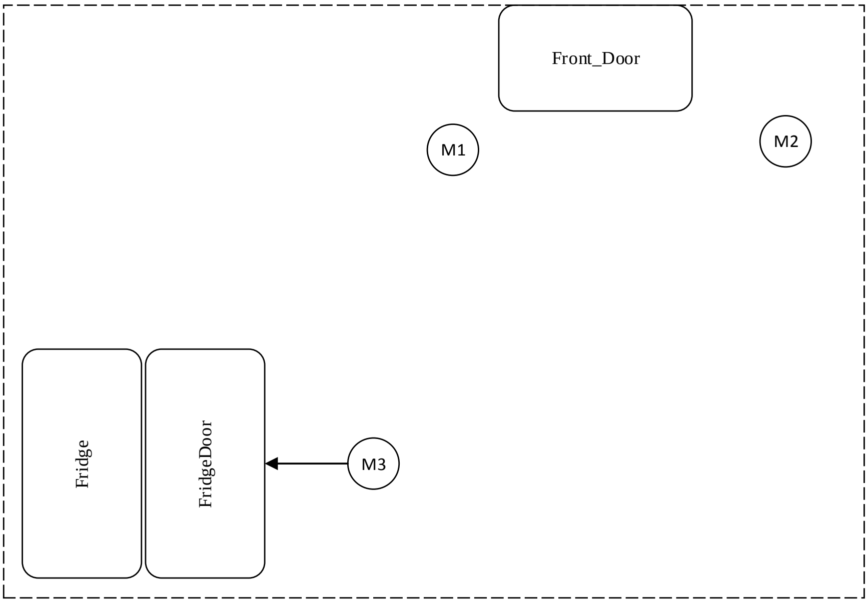

FIGURE 6.2: Hypothetical Kitchen Nodes

The *ReplenishFridgeStock* ADL can be considered as being made up of two separate ADLs - the first one being the activities involved in **placing the order** and the second one being the activities which occur after the order has been delivered: i.e. **placing the delivered items in the fridge**. It is important to know which part of the ADL is being conducted because they may not have the same Constraints - for instance, ordering items does not require a physical presence because the user can log in to the fridge interface without being physically present in the Kitchen. However placing items in the fridge will require a physical *presence*.

6.3.2.1 ADL 1, Activity Class I: Locally by Human

ReplenishFridgeStock requires that an activity of *OpenFridgeDoor* (i.e. a_{12}) is performed and completed by an activity of *PlaceItemsInFridge* (i.e. a_{13}). In addition, to ascertain that the *OpenFridgeDoor* activity actually happened, the `FridgeDoor` sensor must fire to an *active* state. Also, *OpenFridgeDoor* must be followed by a *CloseFridgeDoor* i.e. the `FridgeDoor` must go to the *close* state thus signifying a complete event.

[5] The nodes `FrontDoor`, `M1` and `M3` were added to Figure 6.2 in order to give an idea of where the nodes `Fridge`, `FridgeDoor` and `M3` are, hypothetically, in the KITCHEN in Figure 5.1.

	$\mathbf{a_{1}}$	$\mathbf{a_{2}}$	$\mathbf{a_{3}}$	$\mathbf{a_{4}}$	$\mathbf{a_{5}}$	$\mathbf{a_{6}}$	$\mathbf{a_{7}}$	$\mathbf{a_{8}}$	$\mathbf{a_{9}}$	$\mathbf{a_{10}}$	$\mathbf{a_{11}}$	$\mathbf{a_{12}}$	$\mathbf{a_{13}}$	$\mathbf{a_{14}}$	$\mathbf{a_{15}}$	$\mathbf{a_{16}}$	**Core Activities**
ASeq 1	x	x	x	x	x	x	o	x	x	x	x	x	x	x	x	x	a_{12}, a_{13}
ASeq 2	x	x	x	x	x	o	x	x	x	x	x	x	x	x	x	x	a_{12}, a_{13}
ASeq 3	x	x	x	x	o	x	o	x	x	x	x	x	x	x	x	x	a_{12}, a_{13}
ASeq 4	x	x	x	x	o	o	x	x	x	x	x	x	x	x	x	x	a_{12}, a_{13}
ASeq 5	x	x	x	x	x	x	o	x	x	x	x	x	x	o	x	x	a_{12}, a_{13}

Table 6.2: Activity Class I for *Activity of Daily Living* (ADL) 1

In all the Activity Class Tables in this Chapter, a '**x**' means the activity is part of the Activity Sequence (ASeq) in question and a '**o**' means it is not. The core activities are identified following the approach in [10]. The constraints on the different ways of achieving the *ReplenishFridgeStock* (*physically* by human) event are given in Table 6.3. In summary, the constraints are that the normal location and current location must be the Kitchen and there must be a *presence*.

Core activities	a_{12} and a_{13}
Current location ($\acute{Z}$)	Kitchen[6]
Normal location (Z)	Kitchen
presence (Pr)	≥ 1

Table 6.3: Constraints on *Activity of Daily Living* (ADL) 1, Activity Class I

Considering Presence (Pr), Peripheral (Per) and Proximality (ρ) attributes, a table of scenarios such as shown in Table 6.4 can be drawn up to represent some of the possibilities for this event. In this table, each row is a scenario. The decision reached for each scenario is shown in the **Decision** column and each of the decisions is analysed in Section 6.3.2.1 which is the section immediately following the Table.

[6] *presence* is therefore in the Kitchen.

Scenario	Presence (Pr)	`FridgeDoor`	`M003`	**Decision**
1	0	Open	Off	**Anomaly**
2	0	Closed	On	**Anomaly**
3	≥ 1	Closed	Off	**Normal**[7]
4	≥ 1	Closed	On	**Normal**
5	0	Closed	Off	**Normal**
6	0	Open	On	**Anomaly**
7	≥ 1	Open	Off	**Anomaly**
8	≥ 1	Open	On	**Normal**

TABLE 6.4: ADL 1 (Activity Class I) Scenarios table

Analyses[8] of Table 6.4 Decisions

Each scenario is now analysed and the decision reached in each case explained. It should be remembered that for every *presence*-bound activity (e.g. a_{17} in Table 6.1), there must be at least one *presence* (where *presence* is determined by a motion sensor in a zone being active).

Consider Scenario 1 in which there is no one in the KITCHEN ($Pr = 0$) and the `FridgeDoor` and motion sensor `M003` are both active. This is an **anomaly** because `M003` being in state *on* indicates at least one *presence* and the `FridgeDoor` cannot be opened unless there is someone there.

Consider Scenario 2: there is no one in the KITCHEN and the motion sensor `M003` is active. The `FridgeDoor` in this scenario is in state *closed.* This scenario is considered **anomalous** because the `M003` is detecting a *presence* (motion) that is not there.

Consider Scenario 3 in which there is a *presence* in the KITCHEN ($Pr \geq 1$) and the `FridgeDoor` and motion sensor `M003` are both inactive. This is **normal** because the other nodes may be active or inactive as long as there is a *presence* in the KITCHEN. A similar explanation applies in Scenario 4.

Consider Scenario 5, where there is a *presence* in the KITCHEN and the `FridgeDoor` and motion sensor `M003` are both inactive. This is **normal** because all the nodes are inactive and there is a*presence* in the KITCHEN.

[7] A **Normal** Decision can be described as a *True Negative* or *benign* situation.

[8] Note that the *next* location the user went to is not considered in these analyses. Considering this value would result in different decisions. For instance, if the user moves from the KITCHEN to a proximal zone, say the DINING and the motion sensor `M003` is still active even though Pr in the KITCHEN is now **0**. This combination of factors (which in fact describe Scenario 2 exactly) may not be the sign of an anomaly because motion sensors are observed to stay on for some time after the motion that triggers them.

Consider Scenario 6. During this scenario there is no one in the Kitchen, the `FridgeDoor` is open and the motion sensor `M003` is on. This is **anomalous** because the `FridgeDoor` and `M003` cannot be normally active without a *presence* in the Kitchen.

Consider Scenario 7, where there is a *presence* in the Kitchen ($Pr \geq 1$) and the `FridgeDoor` is open whilst the motion sensor `M003` is inactive. This decision reached is that this is an **anomaly** because the user cannot reach and access the `FridgeDoor` without triggering motion sensor `M003`.

Consider Scenario 8, where there is a *presence* in the Kitchen and the `FridgeDoor` and motion sensor `M003` are both active. This decision reached is that this is **not an anomaly** because even though both nodes are active, as long as there is a presence, then everything is normal.

Still considering the Activity Sequences and core activities in Table 6.2 and the scenarios in Table 6.4. Irrespective of the particular ASeq (or, indeed the Activity Class) which is followed to place the order of the food stock, the `FridgeDoor` must be in state *open* (i.e. a_{12}) before items can be added to the `Fridge` (since fridges, currently, do not physically fill themselves). After this, the door may physically be returned to state *closed* (a_{13}) - or even left to close automatically. Therefore, for event *ReplenishFridgeStock*, with an established set of potential activities (a_{1} - a_{16}), when *ReplenishFridgeStock* is started (making use of any of the Activity Sequences presented in Table 6.2) the FDMA checks for core activities a_{12} and a_{13} and, if either one or both are missing from the ASeq, the situation is flagged as anomalous.

A discussion on Rule Induction using the Machine Learning tool Weka and the Rule Induction Algorithm PRISM is done in Section 6.5. Prior to this discussion, however, some rules based on domain knowledge are drawn up in several sections. An example rule for the event physically *OrderItems* in this particular SH can be expressed as follows:

Box 6.3. Pr, Z, ρ, Nx, ASeq, Core Activities

If (Pr, ≥ 1) & (Z, Kitchen) & (ρ, On) & (Nx, na) & (ASeq, correct) & (Core Activities, a_{12}; a_{13}) $\Rightarrow$ (No Anomaly)

This rule, as with all the rules in this work are applicable to certain scenarios.

According to this rule, if there is no one in the Kitchen, and the node's proximal is on (the network reading is not applicable), the ongoing activity is following a known Activity Sequence and the core activities are as they should be then no anomaly will be flagged. To test this rule, consider the event *ReplenishFridgeStock* during which the `FridgeDoor` must be *open* (one of the core activities). Applying the rule to a scenario where there is at least one user in the Kitchen, the `FridgeDoor` is inactive, the correct

Activity Sequence is being followed and the core activity is happens leads to the decision that the scenario is **anomalous**.

Box 6.4. Example: Event, core activity, *Per*

If (Pr, ≥ 1) & (Z, Kitchen) & (Per, inactive) & (Activity, a_{13}) & (ASeq, correct) ⇒ (Anomaly)

This scenario correctly resolves to an anomaly because even though there is a *presence* in the correct zone (Kitchen), the activity a_{13} cannot be completed if the peripheral `FridgeDoor` is inactive.

6.3.2.2 ADL 1, Activity Class II: Remotely by Human

In this Activity Class the user places the order remotely. Example Activity Sequences for this scenario are shown in Table 6.5.

	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	**Core Activities**
ASeq 1	x	x	x	o	x	x	x	x	a_4, a_9, a_{10}, a_{11}
ASeq 2	x	o	o	x	x	x	x	x	a_4, a_9, a_{10}, a_{11}
ASeq 3	x	x	x	o	x	x	x	x	a_4, a_9, a_{10}, a_{11}

Table 6.5: Activity Class II for ADL 1

For this ADL, activities a_1 to a_3 and a_{16} are not relevant and are therefore not shown in the table. These activities are not applicable because the user does not physically *go to* the Kitchen or the `Fridge` in order to complete the ADL. As was done with the previous Activity Class, each of the scenarios and decisions is analysed individually.

The selected example scenarios and their corresponding decisions for this event are presented in Table 6.6. The premise is that if there is no *presence* (i.e. $Pr = 0$) and the Network (Nx) state is Up-Normal, then the situation is **Normal**. The list of allowed websites can be fed to the FDMA during the SH set-up stage as described in Section 3.8.1 so that during this event, if the websites being accessed are on the list of allowed websites, the FDMA will take the decision that the situation is normal, otherwise the decision reached would be that the on-going event is an anomalous. In addition, the occurrence of *presence*-bound activities where no presence is expected will be identified by the FDMA as being anomalous. These factors are taken into account to populate the decision column of Table 6.6.

Scenario	Pr	Location (Z)	Activity	Network (Nx)	**Decision**
1	≥ 1	KITCHEN	a_4	Up-Normal	**Anomaly**
2	≥ 1	TOILET	a_{15}	Up-Normal	**Anomaly**
3	≥ 1	BIGBEDROOM	a_9	Up-Normal	**Normal**
4	0	n/a	a_{17}	Up-High	**Anomaly**
5	0	n/a	a_{18}	Up-Normal	**Normal**
6	0	n/a	a_3	Up-Normal	**Anomaly**
7	≥ 1	KITCHEN	a_{17}	Up-Normal	**Anomaly**

TABLE 6.6: ADL 1 (Activity Class II) Scenarios table

Analyses of Table 6.6 Decisions

Consider Scenario 1, where there is a *presence* in the KITCHEN ($Pr \geq 1$), and the on-going activity is a_4. If the Network state is Up-Normal, this is an **anomaly** because there should not be a *presence* in the single-user home for a remote order.

Consider Scenario 2, where there is *presence* in the TOILET while a_{15} is ongoing. Note that a_{15} requires a *presence* and can only occur in the KITCHEN. In addition, the order is being placed remotely and can be placed from the TOILET. This scenario is **anomalous** because even though there is a *presence* as required, it is in the wrong location of the house (and the location TOILET in question is not proximal to the KITCHEN where the *presence* should be). it is further anomalous because a_{15} should not be ongoing during order items-only.

Consider Scenario 3, where there is a *presence* in the BIGBEDROOM ($Pr \geq 1$) and activity a_9 is ongoing with the Nx at state Up-Normal. This is **anomalous** because there is a *presence* in the KITCHEN.

Scenario 4 is **anomalous** because a_{17} is not part of the normal ASeq in Table 6.5. It is also anomalous because the state of the network is Up-High and the network state is supposed to be Normal for a normal situation.

Consider Scenario 5. This situation is **normal** because the event does not require a presence ($Pr = 0$), there is therefore no active zone, the ongoing activity can be carried out remotely and the network is at state Up-Normal.

Consider Scenario 6. In this Scenario, there is no *presence* in the Kitchen, `M003` is on and the state of the network is Up-Normal. This is an anomalous situation because there is no one in the KITCHEN as required by this ADL however, the motion sensor which detects presence is active.

Consider Scenario 7. There is a *presence* in the the KITCHEN, the activity ongoing is a_{17} which is not part of any of the ASeq for this event. Therefore even if the Nx state is Up-Normal i.e. an acceptable state, the situation is overall **anomalous**.

The following Box gives one (possible) **discriminant rule** that governs all the cases in this scenario (Order Items Human, Local)

Box 6.5. Pr, η, Per

If (Pr, ≥ 1) & (η, active) & (Per, active) $\Rightarrow$ (Normal)

This rule adequately captures the scenarios in Table 6.6.

From the perspective of Network-based anomalies, an anomaly can also be said to be ongoing if there are an abnormally high number of occurrences of certain activities that should only normally occur **once or a few times** in every given Activity Sequence mainly because of the *types of activities* that they are. Taking as an example Scenario 4 in Table 6.6. Consider activities a_4 and a_{17}: too many occurrences of either one of these activities - i.e. repeated, perhaps, rapidly-occurring (failed) login attempts - may be indicative of a brute force attack. In order to counter this threat a threshold of a maximum number of login attempts can be set during the set-up and training stage of the SH. An example constraint can be *Total Number of Login attempts* = ***5*** *per minute.* This example demonstrates the combination of the network and activities attributes.

6.3.2.3 ADL 1, Activity Class III: Machine

For the *ReplenishFridgeStock* to be considered **complete** - even in a machine-driven scenario - activity a_{12} must occur. However, consider that this event is made up of two sub-events: *placing* the order online by the fridge and adding the items to the fridge and that the first sub-event is the one being considered. The Activity Sequences in Table 6.7 give several examples of how this can be accomplished. This event is network-bound and, unlike with events that are carried out by humans, does not require a *presence*.

	a_6	a_7	a_8	a_9	a_{10}	a_{11}	**Core Activities**
ASeq 1	x	o	x	x	x	x	a_9
ASeq 2	x	o	x	x	x	o	a_9
ASeq 3	o	x	x	x	x	x	a_9

TABLE 6.7: Activity Class III for ADL 1

Scenario	$\omega \subset W$	Node (η)	Activity	Nx	**Decision**
1	yes	`Fridge`	a$_6$	Up-Normal	**Normal**
2	yes	`Fridge`	a$_8$, a$_{14}$	Up-Normal	**Anomaly**
3	no	n/a	a$_9$	Up-Normal	**Anomaly**
4	yes	`Fridge`	a$_7$	Up-Normal	**Normal**

Table 6.8: ADL 1 (Activity Class III) Scenarios table

For this Activity Class, activity a$_{11}$ may potentially be redundant depending on how the overall Fridge security system is set up. The fridge can automatically log out for instance. The scenarios in Table 6.8 are used to analyse this Activity Class.

Analyses of Table 6.8 Decisions

Consider Scenario 1. The network state is Up-Normal and the website being accessed is on the trusted list of websites ($\omega \subset W$ is *yes*). The correct node is active in this Scenario. The ongoing activity (a$_6$) is part of a number of the typical Activity Sequences (see Table 6.7). These factors together make the overall situation **normal**.

Observe in Scenario 2 that even if the website which was accessed is on the list of acceptable websites, the node involved in the scenario is normal and the network state is Up-Normal, but because a$_{14}$ is part of the activities happening, the decision reached is that the scenario represents an **abnormal** situation.

In Scenario 3 the machine places the order from a website that is not on the trusted list of websites. Therefore, irrespective of the fact that all the other attributes appear to indicate that everything is normal, the overall scenario is **abnormal**.

Scenario 4 is **normal** because the website is recognised as a trusted website, the correct node (`Fridge`) is involved in the scenario, the activity is part of the sequence of steps that should be taken to complete the ADL and the network state is Up-Normal.

Therefore, taking the preceding analyses into consideration, the following example scenario correctly resolves to an anomaly:

Box 6.6. ω

If (Pr, n/a) & (Z, n/a) & (ρ, n/a) & (Nx, Up-Normal) & ($\omega \not\subset W$) & (Core Activities, a$_9$) & (ASeq, correct) $\Rightarrow$ (No Anomaly)

Activity Sequences can be expressed in a way that shows the actual *sequence* of activities in the specific ASeq. Thus, instead of using a table of *several* suggested Activity Sequences such as, for example, in Table 6.7, an ASeq can be expressed as follows:

$$\bullet\ \mathrm{a}_1 \longrightarrow \mathrm{a}_2 \longrightarrow \mathrm{a}_3 \longrightarrow \ldots \longrightarrow \mathrm{a}_n\ \bullet$$

where a_1, a_2, a_3 and so on are activities in the Activity Sequence and activity a_1 happens before activity a_2, activity a_3 happens after a_2 and so on.

Drawing up such an ASeq involves looking up the activities on an activities list like Table 6.1 and linking the activities with a ⟶. The bullet (•) at the start and end of the Activity Sequence indicates the beginning and end of the sequence of activities. This representation can be used to express the **order** of activities as well as show clearly if any activities are **repeated**.

Using this method to represent the *ReplenishFridgeStock* event gives:

$$\bullet\ \mathrm{a}_6 \longrightarrow \mathrm{a}_8 \longrightarrow \mathrm{a}_9 \longrightarrow \mathrm{a}_{10} \longrightarrow \mathrm{a}_{11} \longrightarrow \mathrm{a}_{13} \longrightarrow \mathrm{a}_{14} \longrightarrow \mathrm{a}_{15} \longrightarrow \mathrm{a}_{16}\bullet$$

Assume for this scenario that $Pr \geq 1$. An analysis of this ASeq shows that even if a *presence* is detected, because of the fact that activity a_{12} (Open `FridgeDoor`) is not part of the ASeq - and this is a required activity for items to be added to the fridge - this ASeq is anomalous. Essentially, a complete and successful *ReplenishFridgeStock* ADL depends on `FridgeDoor` being in state *open* i.e. activity a_{12} must be carried out.

If an activity is repeated in a given ASeq, it can be shown by inserting it into this sequence more than once. In the following example, a_2 is a repeated activity:

$$\bullet\ \mathrm{a}_1 \longrightarrow \mathrm{a}_2 \longrightarrow \mathrm{a}_3 \longrightarrow \mathrm{a}_2\ \bullet$$

This particular notation does not show options (such as alternative activities) or time however it can be further enhanced. However, optional Activity Sequences for completing the same event are shown in Tables 6.2, 6.5, 6.7, 6.10, and 6.14.

6.3.2.4 Anomaly Detection using the *Order* of Activities

As explained in Section 6.2, for AD purposes, if a core activity is missing from an Activity Sequence then there is an anomaly; this is irrespective of the order in which the ASeq is completed. However, there is potentially a further anomaly if the *order of activities* is wrong. Thus, considering the *happens before* relationships, from an AD standpoint, for every *ReplenishFridgeStock* event, the `FridgeDoor` must change to state *open* (activity a_{12}) before activity a_{13} can occur. In addition, before the `FridgeDoor` and the `Fridge` are accessed, node `M003` must be accessed first (i.e. activities a_{12} and a_{13}).

the motion sensor (e.g. `M003`) must be accessed too (*before* the `Fridge` is reached). Also, activity a_5 must occur before a_8. Otherwise, an anomalous event is flagged and the ASeq is anomalous. Therefore, the following ASeq is anomalous. Thus consider the following sequence of activities:

• $a_5 \longrightarrow a_4 \longrightarrow a_6 \longrightarrow a_7 \longrightarrow a_8 \longrightarrow a_9 \longrightarrow a_{10} \longrightarrow a_{11} \longrightarrow a_{13} \longrightarrow a_{14} \longrightarrow a_{12} \longrightarrow a_{15}$ •

Some peculiarities with respect to these particular activity sequence are that:

Analyses of Happens Before/Happens After Relationships Scenarios:

1. a_5 can happen before a_4 if the user looks up the Fridge contents from a remote location;
2. having a_6 and a_7 on the same ASeq may be a potential source of conflict;
3. if a_{13} happens and it is not preceded directly or otherwise by a_{12}, then the scenario being described is anomalous.

This entire Activity Sequence is therefore redundant. To avoid these anomalies, the FDMA can be trained explicitly with activities that have these happens before and happens after relationships. In general, Smart Home occupants' regular ADLs (i.e. events), activities and Activity Sequences can be programmed into the FEMS (see Section 3.6) so that as they carry out their ADLs, the relationships between these attributes can be monitored for anomalies.

6.3.3 Activity of Daily Living 2 (ADL 2): Update Fridgeware

The fridge software (referred to in this Chapter as fridgeware) update can be carried out locally or remotely, by a user, by a remote third-party, or as part of a self-update event by the fridge. In addition, pre-scheduled auto-updates are possible. This event, therefore, is not overall a *presence*-bound event however some of the ways of achieving it require a human i.e. if the update is being carried out by a person locally and physically at the Fridge screen. The state of the Network Nx during the update is however important in determining if this ADL is anomalous or not. In addition, the average number of bytes per minute and packets per byte values can be used along with the number of network connections, to determine the level of network activity, whether Up-High, Up-Normal or Down. The total number of login attempts during this event is also vital for anomaly detection. This section analyses several scenarios that describe how the fridgeware can be updated locally, remotely and by a self-update by the fridge itself.

6.3.3.1 ADL 2, Activity Class I: Locally by Human

This subsection analyses several scenarios that describe how the fridgeware can be updated locally. We assume the maximum allowed number of login attempts is 5 (after which the user is locked out and a third-party security company is alerted). One example ASeq that can be followed to complete this event is

$$\bullet\ a_1 \longrightarrow a_2 \longrightarrow a_3 \longrightarrow a_4 \longrightarrow a_{17} \longrightarrow a_{18} \longrightarrow a_{20} \longrightarrow a_{11} \longrightarrow a_{16}\ \bullet$$

Let the scenarios in Table 6.9 represent several scenarios that occur over a period in the SH.

Scenario	Login attempts	Activities	Network (Nx)	**Decision**
1	2	a_1-a_4, a_{11}, a_{16}-a_{20}	Up-Normal	**Normal**
2	1	a_8, a_{14}	Up-Normal	**Anomaly**
3	15	a_{17}	Up-Normal	**Anomaly**
4	2	a_{18}	Up-Normal	**Normal**

Table 6.9: ADL 2 (Activity Class I) Scenarios table

The analysis of the decisions reached in each of the scenarios are given next.

Analyses of Table 6.9 Scenarios:

In Scenario 1 the number of login attempts is below the threshold, the activities are correct and the network state is normal. These factors together make this a **normal** situation.

In Scenario 2 the total number of login attempts is less than the maximum of five (5), the network is at state Up-Normal however the activities being carried out do not fit within the framework of the ASeq that can be used to accomplish this task. These activities belong to the ASeq for a different Activity of Daily Living, ADL 1. This scenario therefore describes an **abnormal** situation.

Scenario 3 shows a scenario in which the number of login attempts exceeds the threshold of 5. The event as presented is **anomalous**. This decision is reached irrespective of the fact that the activity that was carried out was a normal, expected activity based on the selected ASeq or that the network was at a normal state.

Scenario 4 is **normal** because the maximum login threshold is not exceeded, the activity (a_{18}) being carried out is part of the chosen ASeq and the network state is Up-Normal.

Taking these attributes into account - along with the attributes *website*, *proximal*, *zone* and *core activities*, the following example query for this particular Activity Class correctly resolves to normal:

Box 6.7. Example: Local Physical Update by Human

If $(Pr, \geq 1)$ & $(\acute{Z}$, Kitchen) & (`Fridge`, On) & (`M003`, On) & (Nx, Up-Normal) & ($\omega \subset W$) & (Core Activities, a_{17}, a_{19}, a_{20}) & (ASeq, correct) $\Rightarrow$ (Normal)

This decision is correct because there is a *presence* (this is required for this Activity Class since the update is being carried out via the fridge's physical interface by a human), the zone is the correct one, the other node that is active besides the `Fridge` is proximal to the fridge, the Nx state is Up-Normal, the website is on a trusted list of websites, the core activities are correct and the ASeq being followed is actually for this activity.

6.3.3.2 ADL 2, Activity Class II: Locally, USB dongle

In this subsection, the Activity Class table itself is first used to discuss how activities themselves can be used to discover anomalies. Given the three (3) Activity Sequences in Table 6.10. The aim is to analyse each of them and reach a decision about which one of them is anomalous? An 'x' indicates that the activity in question occurred while an 'o' indicates it did not occur.

	a_1	a_2	a_3	a_4	a_{17}	a_{19}	a_{20}	a_{11}	a_{16}	**Core Activities**
ASeq 1	x	x	x	x	x	x	x	x	x	a_{17}, a_{19}, a_{20}
ASeq 2	x	x	x	x	x	x	o	x	x	a_{17}, a_{19}, a_{20}
ASeq 3	x	x	x	x	x	x	x	o	x	a_{17}, a_{19}, a_{20}

Table 6.10: Activity Classes II and III for ADL 2

Decisions for Table 6.10

ASeq 1 is **normal** because all the activities on the list are necessary to complete ADL 2.

ASeq 2 and ASeq 3 represent **anomalous** situations, the former because activity a_{20} is missing and the latter because the login session is allowed to expire which may provide a potential attack opportunity for a malicious party to gain access to the fridge whilst it is still logged on. An example Activity Sequence (ASeq) of steps in this case can be

$$\bullet\ a_1 \longrightarrow a_2 \longrightarrow a_3 \longrightarrow a_4 \longrightarrow a_{17} \longrightarrow a_{19} \longrightarrow a_{20} \longrightarrow a_{11} \longrightarrow a_{16}\ \bullet$$

This Activity Class has some *presence*-bound (Pr), node-bound (η) and location-bound (Z) activities. For example, the activities a_1, a_2 and a_3 are location-bound and must happen in the KITCHEN. Considering these specific constraints, Table 6.11 can be drawn up and used to illustrate how these constraints on this Activity Class can be used for Anomaly Detection. Taking Pr, η, Z and ρ factors into account, the analysis of the scenarios in Table 6.11 leads to the decisions shown. In this part of this subsection, the decisions reached are analysed. The proximality relationships are obtained by looking up the relationships in Table 5.14, the Aruba SH proximals table.

Scenario	Activity	Pr	`M003`	$\acute{Z}$	$-Z$	$\acute{Z}\triangle -Z$	Network	**Decision**
1	a_{19}	≥ 1	on	KITCHEN	KITCHEN	Yes	up-Normal	**Normal**
2	a_5	≥ 1	on	TOILET	KITCHEN	No	up-High	**Anomaly**
3	a_{17}	≥ 1	on	BIGBEDROOM	LIVING ROOM	Yes	up-Normal	**Normal**
4	a_{17}	≥ 1	off	BIGBEDROOM	LIVING ROOM	Yes	up-Normal	**Anomaly**
5	a_2	≥ 1	off	KITCHEN	KITCHEN	Yes	up-Normal	**Anomaly**

TABLE 6.11: ADL 2 (Activity Class III) Scenarios table

As introduced in Chapter4, Z represents the current location and $-Z$ represents the previous location.

Taking Scenario 1, a **normal** decision is correctly reached because a_{19} is part of the normal ASeq for this event, there is at least one person in the KITCHEN (while this activity is going on), the only other active node (i.e. `M003`) is proximal to the `Fridge`, the previous zone the user was in is the kitchen and the network is at state Up-Normal.

Scenario 2 is **anomalous** for several reasons. The first reason is because the activity shown is not part of the normal Activity Sequence for this event. Secondly, even if the user was in the correct zone ($-Z$ = KITCHEN), because their new (current) location $\acute{Z}$ is the TOILET and the two zones are not proximal (i.e. the user could not have feasibly moved between these two zones whilst `M003` was still in state on), this situation is also

anomalous. Thirdly, the network state is Up-High and this is indicative of too much network activity.

Considering Scenario 3, activity a_{17} is a core activity for this event, a presence is required and, `M003` is at the correct state. However, the user moved from the Living Room to the BigBedroom. Looking up proximality in Table 5.14, the two zones are indeed proximal. Since the Network state is at Up-Normal, the scenario is **normal**.

Scenario 4 describes a situation where during this event (Fridgware update), the home occupant performs activity "Update Fridgeware" (a_{17}) on the Fridge interface. However, the current zone ($\acute{Z}$) is the BigBedroom, the zone the user left is the LivingRoom and the network is at state Up-Normal. This situation is **anomalous** because node `M003` is what indicates that there is indeed a *presence* in the correct location in the house and in this case `M003` is *off*.

In Scenario 5, even if there is a report of a *presence*, because `M003` is *off*, the situation is **anomalous**. All the other conditions make the scenario appear normal but this is not the case.

6.3.3.3 ADL 2, Activity Class III: Remotely by Human

Given the following example normal Activity Sequence for this Activity Class:

$\bullet\ a_4 \longrightarrow a_{17} \longrightarrow a_{18} \longrightarrow a_{20} \longrightarrow a_{11}\ \bullet$

Given this ASeq and the constraint that the event is *Network*-bound (the user carries out the update over the Internet). Assume that the scenarios in Table 6.12 are as shown.

Scenario	Activity(ies)	Pr	`M003`	$\acute{Z}$	$-Z$	$\acute{Z}\triangle -Z$	Network	**Decision**
1	a_{18}	0	off	n/a	n/a	n/a	up-Normal	**Normal**
2	a_{20}	0	on	n/a	n/a	n/a	up-High	**Anomaly**
3	a_1, a_{17}, a_{20}	≥ 1	on	Dining	Kitchen	Yes	up-Normal	**Normal**

Table 6.12: ADL 2 (Activity Class IV) Scenarios Table

The analysis of each of the Decisions reached in Table 6.12 is presented next.

Analyses of Table 6.12 Decisions

Scenario 1 is **normal** because during this scenario, the user carries out an activity that is part of the normal set of activities for this Activity of Daily Living (ADL).

Scenario 2 is **anomalous** because even though `M003` is *on*, no *presence* is detected.

Scenario 3 is **normal** because all the activities are correct, a *presence* is detected (`M003` is *on*) and the two zones that the user moved between whilst this activity was ongoing are proximal. Observe that this scenario describes a situation where the user updates the fridgeware over the Internet - i.e. remotely - whilst being inside the house.

6.3.3.4 ADL 2, Activity Class IV: Self-Update

During this Activity Class, the fridge self-updates its own fridge software. Therefore, the event is not a *presence*-bound event. This means that a human presence is not required for this event to be successfully completed. See the Definition of *presence*-bound anomalies in Subsection 5.4.2 for a discussion on *presence*-bound events and activities.

One example normal ASeq by which this ADL is completed is

$$\bullet\ \mathrm{a}_{17} \longrightarrow \mathrm{a}_{18} \longrightarrow \mathrm{a}_{20}\ \bullet$$

Activity a_{11} is not explicitly included in the ASeq because the `Fridge` can log the user out automatically after a maximum allowed *log in period.* The two core activities are a_{17} and a_{20}. Table 6.13 gives some examples of normal and anomalous scenarios for this event. The presence of a person in the (correct) zone in this case may not necessarily imply an anomaly with respect to the particular Activity Class being considered.

Scenario	Pr	$\acute{Z}$	Activity	Nx	**Decision**
1	≥ 1	KITCHEN	a_{18}	up-Normal	**Normal**
2	0	n/a	a_{18}	up-Normal	**Normal**
3	≥ 1	KITCHEN	a_{18}	Down	**Anomalous**
4	0	n/a	a_{18}, a_{20}	up-Normal	**Normal**

TABLE 6.13: ADL 2 (Activity Class V) Scenarios Table

Analyses of Table 6.13 Decisions

Consider Scenario 1. The activity is part of the Activity Sequence for this ADL and the Nx is at state Up-Normal as required; the fact that there is a *presence* in the Kitchen does not indicate an anomaly. The overall scenario is therefore **normal**.

Scenario 2 is also **normal** because the activity and network state are as expected.

Scenario 3 is anomalous because the network state is down implying that this ADL cannot be completed successfully.

Scenario 4 is **normal** since activities a_{18} and a_{20} are essential for this ADL to be completed successfully.

6.3.4 Activity of Daily Living 3 (ADL 3): Going to Toilet

6.3.4.1 ADL 3, Activity Class I: Going to Toilet

	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	**Core Activities**
ASeq 1	x	x	x	x	x	x	a_{23}, a_{24}

Table 6.14: ADL 3 (Activity Class)

Before this ADL is analysed one premise has to be established: there is generally only one way in which a user can physically *go to* the zone Toilet. Notwithstanding this limited set of scenarios, from the standpoint of privacy it would expected that during this ADL, activity a_{24} (*ClosesToiletDoor*) will always *happen after* the activity a_{23} (*EntersToilet*) in the ASeq. It would also be expected that activity a_{23} (*EntersToilet*) is possible only after activity a_{22} (*OpensToiletDoor*).

The only possible normal Activity Sequence for this ADL therefore is :

$\bullet\ a_{21} \longrightarrow a_{22} \longrightarrow a_{23} \longrightarrow a_{24} \longrightarrow a_{22} \longrightarrow a_{25} \longrightarrow a_{26}\ \bullet$

For a single-occupant SH, taking the preceding conditions into account, therefore, the following two Activity Sequences are anomalous:

$\bullet\ a_{21} \longrightarrow a_{25} \longrightarrow a_{24}\ \bullet$

or

$\bullet\ a_{24} \longrightarrow a_{23} \longrightarrow a_{22}\ \bullet$

This first ASeq represents an **anomalous** scenario because the user cannot go from the BigBedroom (see Figure 5.1) to the Toilet without passing through the `ToiletDoor` and completing activity a_{22} in the process. Since this *core* activity is not in the ASeq, the situation is anomalous. The scenario is also anomalous because the `ToiletDoor` cannot go to state *closed* if its starting state was not *open*.

The second ASeq is anomalous because the user cannot enter the Bathroom if the `ToiletDoor` is in state *closed*.

An example rule that governs this activity can be drawn up as follows:

Box 6.8. GoToToilet ADL Rule

If $(Pr, \geq 1)$ & $(Z$, Toilet) & (ASeq, correct) $\Rightarrow$ (Normal)

There is more on Rules and Rule Induction in Section 6.5.

6.3.5 Activity of Daily Living 4 (ADL 4): Drive Around Town

This event can be completed by a human driving around Town or by a car self-driving its way around Town.

6.3.5.1 ADL 4, Activity Class I: Drive Around Town - Human

One example normal ASeq which shows how a Human-driven car ADL may be completed is given thus:

$$\bullet\ \mathrm{a}_{27} \longrightarrow \mathrm{a}_{28} \longrightarrow \mathrm{a}_{29} \longrightarrow \mathrm{a}_{30} \longrightarrow \mathrm{a}_{31} \longrightarrow \mathrm{a}_{33} \longrightarrow \mathrm{a}_{34} \longrightarrow \mathrm{a}_{35} \longrightarrow \mathrm{a}_{36}\ \bullet$$

This particular ADL requires a presence (Pr), information on the change in location (i.e. current location $(\acute{Z})$ versus normal location (Z)), proximals (ρ), and the use of a specific node (η) which is a `Car` in this case. In addition, the correct core activity must also be part of the ASeq and the *order* of activities in the ASeq must be correct. The proximals involved must also be correct. For instance, for driving sround Town, the `Car` may be proximal to the `GarageDoor` at the start and end of the event. The formation of such a relationship will not be flagged as anomalous by the FDMA. If, however, the `Car` becomes proximal to the `BathTub` in the Bathroom in Zone G or node `M031` is in the Bathroom in Zone D (see Figures 5.1 and 5.2), while updating the relationships, the FDMA will flag this as an anomaly.

Scenario	Pr	$\acute{Z}$	η	Activity	$\acute{\rho}$	**Decision**
1	≥ 1	Garage	`Car`	a_{33}	`GarageDoor/D004`	**Normal**
2	≥ 1	Garage	`Car`	a_{30}	`TrafficLight`	**Anomaly**
3	≥ 1	Kitchen	`Car`	a_{33}	`Fridge`	**Anomaly**
4	≥ 1	Global	`Car`	a_{33}	`TrafficLight`	**Normal**
5	0	n/a	`Car`	a_{33}	n/a	**Anomaly**
6	0	n/a	None	a_{33}	n/a	**Anomaly**
7	≥ 1	Garage	`Car`	a_{31}	`GarageDoor/D004`	**Normal**

Table 6.15: ADL 4 (Activity Class I) Decision Table

The decisions reached in each of the scenarios presented in Table 6.15 for the Drive Around Town (Human) ADL discussed.

Analyses of Table 6.15 Decisions

Scenario 1 indicates a **normal** scenario. This is because there is a presence in the Garage, the `Car` is the active node and the activity (a_{33}) is part of a a normal ASeq for this ADL. The proximals for this node in this scenario are shown to be the `GarageDoor` and `D004`. Based on the SH being considered (5.1), this is normal.

One of the reasons why Scenario 2 is **anomalous** is because the proximal of `TrafficLight` cannot be accurate if the `Car` is still in the Garage and the user is still in the process of authenticating (a_{30}), an activity which precedes any activity that will lead the user to a point where the car's proximal can be a `TrafficLight`.

Scenario 3 is **anomalous** since the user leaves home (a_{33}) but somehow the Kitchen is active and the `Fridge` is proximal to the `Car`.

Scenario 4 is **normal** since the user is anywhere outside the home ($\acute{Z}$ = Global) in the `car`, leaving home (a_{33}) and the list of proximals includes a `TrafficLight`.

Scenario 5 and Scenario 6 are **anomalous** because the user is not present in either - and one of the constraints for a human-driven scenario is that a *presence* is required (see Subsubsection 4.2.2.2). Another reason the decision is anomalous is because no node is active in Scenario 6 whereas according to Subsubsection 4.2.2.2, a strict requirement for a successful *DriveAroundTown* is that the `Car` is active.

Scenario 7 is **normal** since all the required elements for a successful event are present: there is a presence in the correct zone using the correct, required node and the activity being performed as part of this ADL is part of what should be on any normal

ASeq for this ADL. Lastly, the proximals to the `Car` are the `GarageDoor` and `D004` and, according to Table 5.14 and Figure 5.1, this is as it should be.

6.3.5.2 ADL 4, Activity Class II: Drive Around Town - Machine

An example normal ASeq for a machine-driven scenario is given below.

$\bullet\ a_{29} \longrightarrow a_{30} \longrightarrow a_{31} \longrightarrow a_{33} \longrightarrow a_{34} \longrightarrow a_{35} \longrightarrow a_{36}\ \bullet$

If activity a_{30} is missing, this could be indicative of an unauthorised party trying to control the car. If activity a_{31} is missing, it is the sign of an incomplete event because without a_{31} the remaining activities cannot happen. Therefore, for security and Anomaly Detection purposes, the event can be constrained by one of these core activities a_{30} and a_{31} depending on the degree of security that the user is interested in.

6.4 Anomaly Rankings

Drawing from the rule rankings as shown in the TextBox 5.1: the Anomaly Rankings TextBox and the discussion on constraints in Chapter 4, the following conclusion can be drawn:

Box 6.9. n-bound activities or events, Constraints

The Anomaly Rank is highest when the constraint n is missing from, or inaccurate during, an n-bound activity or event.

The analysis in this section is used to illustrate how the anomaly rankings as shown in Anomaly Rankings TextBox apply depending on the outcome of an FDMA analysis of a scenario. Assume that for each *Sleep*, Table 6.16 applies

Event	Sleep
Normal Location (Z)	BigBedroom[9]
Presence	$= 1$
GoToBed time[10]	$\geq$ 12 midnight
Constraints[11]	Presence (Pr), Time

Table 6.16: Sleep Event Attributes

Analysis to illustrate Anomaly Ranking:

At 12 midnight (or a few minutes after), the system checks for *presence* in the normal zone that the user should be in; in this case this zone is the BigBedroom. If the BigBedroom is inactive (i.e. user is not present in the BigBedroom) the system will raise the alarm level to **True Positive**. The alarm level is highest because the person should be asleep but appears to be missing.

If the total number of *presences* in Home is exactly 1 then the anomaly ranking can be reduced to a **False Negative** level. The system assumes that at least the legitimate home owner is in the house even if she is not where she should be. Therefore physical intrusion is ruled out and the home occupant is not in imminent danger.

The FDMA then checks for the location of the *Presence* by determining which zone(s) inside or outside the house is active?

If the $\acute{Z}$ is the Toilet, it checks the previous zone $(-Z)$ and the last event.

Having a *presence* in the Toilet during the period when *Sleep* should be happening or having a *presence* in the Toilet for too long will change the alarm ranking to **True Negative**. This is because the Toilet is a **highly-unlikely** location for a person to sleep in safely

If the *presence* is in the LivingRoom, the FDMA checks the value of $-Z$ and the last activity in the last Event Activity Sequence.

If the last event was a benign one e.g *WatchTV* in LivingRoom or *MakePopcorn* in $-Z =$ Kitchen, the system will take the decision to do nothing. This decision will be taken because the home occupant probably fell asleep in the LivingRoom. This anomaly ranking will therefore be changed to a **False Positive** ranking.

The decision reached by the FDMA can be corroborated using CCTV video evidence if available.

6.5 Rule Induction

According to [91], "**Rule Induction** is one of the most important techniques of Machine Learning". It involves identifying relationships in datasets that may not be

[9] *Sleep* is not location-bound; the home occupant may sleep anywhere.

[10] This time value can be obtained, for instance, from an average of the *Sleep* times over several months.

[11] The use of Constraints to aid AD and reduce False Positives is discussed in Chapter 6 but is applied briefly here.

obvious from human observation. The data that is used for Rule Induction can be presented in the form of **tables**; the rows in the tables represent individual *cases* (described in this thesis as *scenarios*), the variables are known as *attributes* and a *decision* column completes each table [91]. Also according to [91] rules can be expressed in the following form:

Box 6.10. Generic Format of a Rule

If (attribute - 1, value - 1) and (attribute - 2, value - 2) and ... and (attribute - n, value - n) then (decision,value)

In this Chapter, the following slightly modified format of the rule is used:

Box 6.11. Modified Generic Format of a Rule

If (attribute - 1, value - 1) & (attribute - 2, value - 2) & ... & (attribute - n, value - n) ⇒ (Decision)

The rule induction algorithm selected for this work is the *PRISM* classification algorithm [92]. This algorithm was selected because it captures and presents the induced rules (representing the relationships "within" data) in a non-cumbersome, straightforward style. PRISM is available in the Machine Learning tool Weka. The goal in this section is to generate rules which can govern and constrain the relationships and interactions between the different components in Smart Homes. The PRISM algorithm takes the following steps for induction of classification rules:

Box 6.12. Prism Algorithm

"(assume there are n (>1) possible classes):
For each class i from 1 to n inclusive:

1. Calculate the probability that class = i for each attribute-value pair
2. Select the attribute-value pair with the maximum probability and create a subset of the training set comprising all instances with the selected combination (for all classes)
3. Repeat 1 and 2 for this subset until it contains only instances of class i. The induced rule is then the conjunction of all the attribute-value pairs selected in creating this subset
4. Remove all instances covered by this rule from the training set

Repeat 1-4 until all instances of class i have been removed"

Using the **PRISM** algorithm to evaluate the dataset in Table 6.4, the following rules are induced for the event *ReplenishFridgeStock*:

```
Scheme:weka.classifiers.rules.Prism
Relation:     Replenish Fridge Stock Local Human
Instances:    8
Attributes:   4
              Presence
              FridgeDoor
              M003
              Decision
Test mode:5-fold cross-validation

=== Classifier model (full training set) ===
Prism rules
If Presence = no
   and FridgeDoor = open then anomaly
If Presence = no
   and M003 = on then anomaly
If FridgeDoor = open
   and M003 = off then anomaly
If Presence = yes
   and FridgeDoor = closed then normal
If Presence = yes
   and M003 = on then normal
If FridgeDoor = closed
   and M003 = off then normal
```

PRISM Rules for Table 6.4

Consider the following which is the first PRISM rule from Listing 6.5.

Box 6.13. Example PRISM Rule (1)

If $(Pr, 0)$ & $(Per, \text{open}) \Rightarrow (\text{Decision}, \text{Anomaly})$

This rule - and all the other rules induced from Table 6.4 - are quite limiting and do not represent the reality of the situation in the SH in question. These rules will potentially capture too many situations and lead to too many False Positive outcomes if they are applied in digital forensics AD processes.

In order to induce rules which better capture the scenarios, more attributes were introduced to Table 6.4. The resulting table is Table 6.17.

Scenario	Presence (Pr)	`FridgeDoor`	`M003`	Normal Location (Z)	Current Location ($\acute{Z}$)	**Decision**
1	0	Closed	Off	n/a	n/a	**Normal**
2	0	Open	On	`Kitchen`	`Kitchen`	**Anomaly**
3	0	Closed	Off	`Kitchen`	`Kitchen`	**Anomaly**
4	0	Open	On	`Kitchen`	`BigBedroom`	**Anomaly**
5	0	Closed	Off	`Kitchen`	`BigBedroom`	**Normal**
6	0	Open	Off	`Kitchen`	`Kitchen`	**Anomaly**
7	0	Open	Off	`Kitchen`	`BigBedroom`	**Anomaly**
8	≥ 1	Closed	Off	`Kitchen`	`Kitchen`	**Normal**
9	≥ 1	Closed	On	`Kitchen`	`Kitchen`	**Anomaly**
10	≥ 1	Closed	Off	`Kitchen`	`BigBedroom`	**Anomaly**
11	≥ 1	Closed	On	`Kitchen`	`BigBedroom`	**Anomaly**
12	≥ 1	Open	Off	`Kitchen`	`Kitchen`	**Anomaly**
13	≥ 1	Open	On	`Kitchen`	`Kitchen`	**Normal**
14	≥ 1	Open	Off	`Kitchen`	`BigBedroom`	**Anomaly**
15	≥ 1	Closed	On	`Kitchen`	`BigBedroom`	**Anomaly**
16	≥ 1	Closed	Off	`Kitchen`	`BigBedroom`	**Anomaly**

TABLE 6.17: ADL 1 (Activity Class I) Scenarios (Table 2)

Using the PRISM rule induction algorithm to induce rules from this dataset (Table 6.17) yields several rules for the *presence*-bound events and activities:

```
Scheme:weka.classifiers.rules.Prism
Relation:TableforPRISM-weka.filters.unsupervised.attribute.Remove-R1
Attributes:    12
              constraintA  = presence in Kitchen
              constraintB  = location is Kitchen
              constraintC  = peripheral is open
              Presence
              Peripheral
              Proximal
              Normal Location
              proximal location
              current location
              previous location
              are active and previous proximal
              Decision

=== Classifier model ===
Prism rules
If Proximal  = on
   and Peripheral = open
   and Presence = one then normal
If Proximal  = off then anomaly
If Peripheral = Closed then anomaly
If Presence = none then anomaly
```

PRISM Rules for Table 6.17

Consider the first rule from this listing:

Box 6.14. Example PRISM Rule (2)

If (ρ, on) & (Per, open) & (Pr, one) $\Rightarrow$ (Decision, Normal)

This rule more adequately captures the scenarios presented in the table and will lead to fewer False Positives when applied.

The Decision Tree in Figure 6.3 (which was derived from an analysis of the Table 6.17 dataset using the Weka RandomTree algorithm) illustrates how the decisions are reached. In this Tree the *presence* values are either *one* or *none* unlike in all the preceding datasets in this work where presence has been either 0 or ≥ 1.

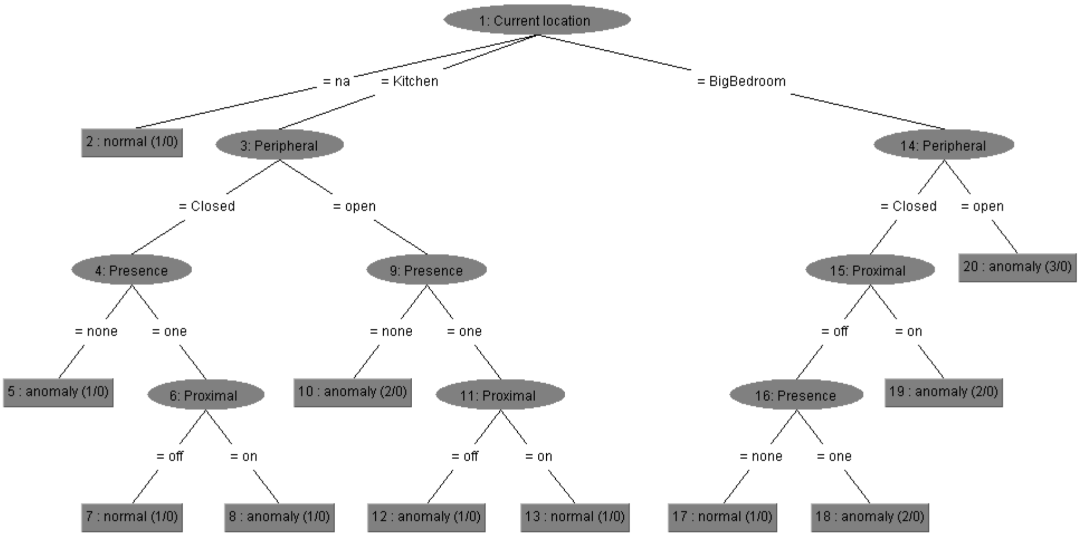

Figure 6.3: Decision Tree for ADL 1 (Activity Class I) Scenarios (Table 2)

Weka was selected because it is a straightforward tool that has the required algorithm and which presents the output in an understandable format. The Induction of **Rule sets** is described as an important Machine Learning technique by [28]. There are several Rule Induction Algorithms that can be used to derive rules or rule sets from datasets [28, 91]. A SH Rule set (or rule) may be used in SH expert-systems to support a more secure planning and design process of smart homes and training of SH AD systems in order to aid Anomaly Detection (AD), maximum evidence acquisition during investigations, and reduce false alarms.

6.6 Chapter Summary

Smart Homes are poised to become more commonplace in the near future. There is a requirement for security assurance solutions in SH. However, in the event that implemented security assurance solutions fail and cyber-physical attacks against SH are successful, the work in this Chapter discusses how the Forensics Decision-Making Algorithm (FDMA) can be combined with the event, activities, core activities and other constraints to investigate anomalous occurrences within SH environments. The PRISM Weka Machine Learning tool was used to induce some rules from two different datasets - the first dataset had fewer attributes and fewer instances than the second one and led to less widely-applicable rules which, if applied, would capture False Positive anomalies. Increasing the number of attributes and the size of the dataset led to a rule that more fully captured the instances and reasoning behind the scenarios in question.

Currently, no method has been designed to automate the selection and application of the appropriate rules for any given scenario. This can form the subject of further research.

It is, just as in Chapter 5 acknowledged that a limitation in this work is the use of scenarios which, though realistic, the test of the system can be more rigorously carried out using real people carrying out real activities and thereby generating real data which can then be analysed using the attributes introduced in Chapter 4.

Chapter 7

Conclusion and Future Directions

7.1 Overview

This thesis introduces an Internet of Things Digital Forensics Framework (IDFF) and Incident Response Plan which can be applied by Forensics Investigators and Incident Response Teams during investigations of Cyber-physical attacks in IoT environments such as the Smart Home (SH). The IoT Framework (IDFF) and IR Plan are introduced in Chapter 3.

This work also investigated anomaly detection in SH by identifying features that can be used to describe and build a baseline security model of a SH. These attributes - which are described in Chapter 4 - include Node and Zone ID, the Modes of nodes, Proximality, Peripherals, Presence, Location, Type, State, and so on. This model described a SH at its most secure state, the state at which all these features are at their most optimal or normal. A SH in this optimal state of security is referred to in this work as being in the Happy Home state. An anomaly is said to have occurred when there is a deviation from the Happy state. This deviation can be caused by an error or an attack. Detecting such an anomaly can be done through the use of a SH monitoring system.

In the Conclusions section of this Chapter a summary of the work done in this thesis is presented. After this, the Challenge faced during the research is discussed and potential Future Work or research directions which can be exploited are presented.

7.2 Conclusions

Smart Home (SH) environments are set to become the most prevalent of smart spaces of the future. Any solutions and methods developed to address security and improve Anomaly Detection (AD) in SH will contribute to the overall feeling of well-being and confidence that SH occupants have in their SH systems.

The methods developed in this thesis can be used to support existing Anomaly Detection (AD) methodologies and tools whilst avoiding their shortcomings of forming training sets and normal baselines which are heavily focused on human actions and behaviours. This work instead applies the **attributes** and (the more stable) **relationships** formed from the actions and interactions between non-human nodes, zones and human players in Smart Home environments. A combination of these relationships and learned normal user behaviours leads to improved approaches to AD.

Existing multivariate Anomaly Detection methods and approaches were reviewed in Chapter 2.

The Internet of Things Digital Forensics Framework (IDFF) that was developed from a review of existing Digital Forensics (DF) frameworks and methodologies as well as from interviews with DF academics was discussed in Chapter 3. An IoT Incident Response (IR) Plan was also introduced in Chapter 3. An investigation into a realistic SH Incident Response scenario was used to evaluate the IoT IR plan.

The attributes identified in Chapter 4 can be used to aid existing AD methodologies and tools in detecting anomalies by creating relationships that are used to build a normal SH setting such that any deviations from this setting would indicate an anomaly. The anomaly detected can be further investigated to determine what type of anomaly it is i.e. an attack or an error, or one of the anomaly types as identified in Section 5.2. Correctly identifying an anomaly type is useful because it can help reduce call-outs for non-threatening events thus ensuring time and money savings. This research did not go further to define a method for differentiating between attacks and errors. This can form the subject of further research.

In Chapter 5, the Forensics Decision-Making Algorithm (FDMA) was demonstrated to be useful for anomaly detection in the real-life Aruba Smart Home dataset (Dataset 17 at the WSU CASAS website (`http://casas.wsu.edu/datasets/`), available as at the time of writing). The FDMA algorithm (see Appendix A), through the use of the *presence* and *proximals* attributes detected the presence of more than one person in the home (without prior training data or information). If the assumption is made that any other *presence* in this single-occupant SH is that of an intruder, the FDMA was thus demonstrated to be useful for physical intrusion detection in SH especially for elderly people living alone.

The approach in this work, which makes use of the attributes of nodes and zones, human activities and the relationships that are formed between these elements, supports existing AD solutions by eliminating the need for training classifiers with significant, labelled, training datasets. In addition, in real-life, these datasets may not exist until *after* an (suspicious) incident occurs thereby reducing the potential for the attack to be captured in real-time whereas, using the FDMA, once the SH security baseline has been created the FDMA continually checks for any deviations from the security baseline. The outputs of this research are as listed:

- The identification of useful attributes and relationships between nodes and zones in Smart Homes; these attributes can be used to define similar relationships in different SHs and, possibly, even beyond the SH environment.
- The identification of a ranking method of anomalies from False Positive (non-event) to True Negatives thus representing fully the spectrum of anomalies that may occur.
- The combination of the FDMA, Relationships activities, Core Activities and constraints to aid AD
- Example rules that demonstrate how the attributes and relationships can be applied to rules and used to constrain and govern these identified relationships thus aiding anomaly detection tasks and contributing towards ensuring peace of mind to the users of SH..

7.3 Limitations and Challenge faced during Research: Relevant Data

Obtaining Smart Home (SH) datasets that closely mimic what is applicable in real-life anomalous situations is one of the challenges that was faced during this research. The requirement was for datasets that did not include only generic sensor data such as motion, temperature, but one that had wired and wireless network data, electricity readings, all combined. Such a data set was not found to be available during the early stages of this research.

In order to overcome this challenge, an approach was taken that involved labelling certain normal, but *unusual*, instances in the Aruba SH dataset as anomalies. By doing this, the normal Aruba SH dataset became the *presence*-detection anomaly dataset.

Irrespective of this workaround - and even though this data deficiency led to the in-depth visual analysis of the existing Aruba dataset which led to the discovery of the proximality attribute - a dataset which has the required IP addresses, port numbers, and other such networked node outputs would have been useful.

A limitation in this work involved using realistic scenarios. It is however a recommendation for any future researchers that testing this system can be more rigorously carried out using real people carrying out real activities and thereby generating real data which can then be analysed using the attributes introduced in Chapter 4.

7.4 Future Work

As part of raising greater awareness of the attributes identified in Chapter 4 (Proximality, Type, Mode, etc.) and the Forensics Decision-Making Algorithm (FDMA) and how these can be applied to anomaly detection in Smart Homes, a peer-reviewed Conference paper (Presence Detection from Smart Home Motion Sensor Datasets: A Model [6]) has been published. More publications are also being written up for two Journals. In addition, the IDFF will be made publicly-available online on ResearchGate which is a forum through which researchers can invite discussion and obtain immediate feedback on their work.

The following are areas that were identified during the course of this research as potential subjects for future research:

7.4.1 Detecting presence in Multi-occupant Smart Homes

The capability to detect a *second* "presence" in single-occupant smart homes is useful from the perspective of security for the home occupants e.g. elderly patients who live alone. An additionally useful capability will be the ability of a SH security system to accurately detect a *third* or *fourth presence* in a home where the normal number of persons in the home is already more than one.

7.4.2 Pet Detection

It would be useful to be able to identify a set of attributes which can be used to accurately detect the presence of pets within smart homes. This will help reduce False Positive firings by AD systems which may erroneously detect the presence of a pet as that of an intruder. One way to achieve this may be to check the height of the pet in combination with the nodes most likely used by pets alone e.g. a `CatFlap`.

7.4.3 Relationship between Internal Doors

As with external nodes, the relationship between internal access nodes can also be explored and added to the constraints so that even more anomalies may be detected and/or False Positives reduced.

7.4.4 Relevant Evidence

A method of determining before an investigation which sources (physical and logical) and evidence (the information from the sources) are relevant will be useful for DF investigations. This can be in the form of a framework which helps Incident Response Teams identify and rank evidence sources that they encounter during investigations in cyber-physical environments. The ranking can be based on a *degree of importance* or relevance. Such a ranking can help investigators save time during investigations because they will approach their tasks knowing which nodes to give highest priority based on its ranking in this framework. A ranking method will help them avoid the loss of potential evidence. The most important sources of evidence can be given the "highest" rank. It also helps with the application of appropriate security measures to protect assets.

7.4.5 Smart Home Relationships Constraints Ontology

A sample SH Constraints Ontology is presented in Appendix C. This ontology illustrates the relationships and attributes that exist between nodes and zones in Smart Homes as discussed in Chapter 4. Intelligent systems such as the Protege framework (`http://www.protege.stanford.edu`) or similar can be used to create and test these relationships and constraints as well as demonstrate their impact on SH operations.

7.4.6 Proximality Index (PI)

The Proximality Index, or PI, is the total number of proximality relationships that can exist between sensors. For example, the PI of one (1) sensor (with itself) is 1. For two (2) proximal sensors, the PI[1] is as illustrated in Figure 7.2. For three sensors, the PI is as shown in Figure 7.3. Figure 7.4 illustrates the four-sensor proximality relationship. The PI for each is calculated.

[1] Note that, apart from in Figure 7.1, the **double** arrow heads are used in place of two separate arrows where one of them will be originating *from* and the other *leading* to the node. This does not apply to the double-headed arrow used to indicate the one-to-one proximality relationship a sensor has with itself. This arrow is counted as a single arrow *only*.

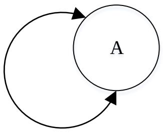

Figure 7.1: One-to-One single sensor Proximality

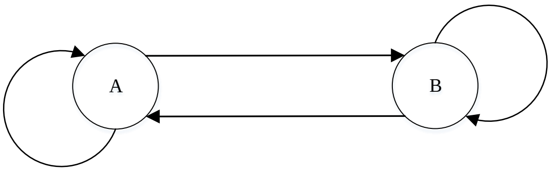

Figure 7.2: Two (2)-sensors Proximality Illustration

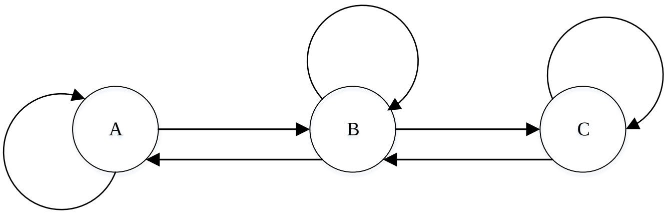

Figure 7.3: Three (3)-sensors Proximality Illustration

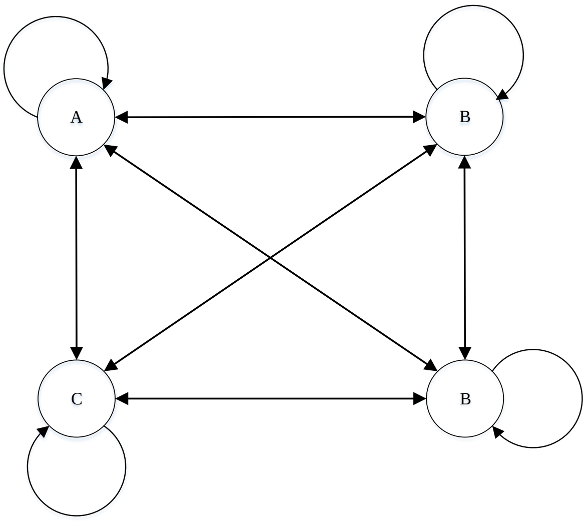

Figure 7.4: Four (4)-sensors Proximality Illustration

From Figures 7.1 to 7.4, it can be deduced that the rule guiding the **Proximality Indices** is calculated thus:

$$PI_i = \mathrm{i}^2$$

where
i = number of nodes and
PI = the Proximality Index (PI)

Applying the equation $PI_i = \text{i}^2$ to calculate the PI for 2-, 3- and 4-sensor proximals gives
$PI_2 = 2^2$, which gives 4
$PI_3 = 3^2$, which gives 9
$PI_4 = 4^2$, which gives 4

A PI value is a theoretic measure that gives an indication of the *density* of the sensors in a location. With respect to "Motion Detection" anomaly detection tasks (from motion sensor data), the higher the PI value (more motion sensors in a location), the greater the chances of detecting motion-related activity. Thus, if the distance between two nodes is reduced e.g. by adding more nodes to a zone, the proximality is increased. This, however increases the risk of anomalous mis-firing of sensors due to node *over-crowding*. A method of calculating how many sensors to deploy within a given space so as to support motion detection needs and maximize the potential for an anomaly to be detected through the use of proximality without over-crowing the space will be a useful future work. Planning the deployment of sensors in this way will help maintain the balance between the provision of security, user comfort and usability.

7.4.7 Thesis Conclusion

This work has provided a holistic view at Smart Home (SH) anomaly investigations and proposed a Digital Forensics Framework (IDFF) and an Incident Response Plan for investigating SH anomalies (Chapter 3). In Chapter 4 a method of describing a secure state of a SH - where all its constituent zones (rooms) and nodes (items in the SH) - are at their secure state. This state is referred to as the "Happy Home" state. See Section 4.2 for more on the "Happy Home". This baseline model was derived through the identification and use of key attributes including the Node ID, Type, Mode, Peripherals, Proximals, Time of Day and so on. When these attributes are at their most secure state - and all activities being carried out in the home follow a recognised Activity Sequence (see 6.2 for more on Activity Sequences) which have all the relevant constituent core activity/activities and fall within the Constraints as identified in Chapter 4 - then the home is said to be in a "Happy Home" state. The secure state of the attributes can be decided upon by the home occupant or a trusted third party such as a family member or carer. One example of a secure state can be for all the external access nodes to be locked by 10 p.m. every night.

An anomaly is aid to have occurred if there is a deviation from this secure baseline. Such a deviation can be detected through the use of the FEMS monitoring device introduced in Chapter 3. The FEMS system runs the FDMA algorithm as part of its decision-making module. Determining if the detected anomaly is the result of a fault or human error or the result of a targeted cyber-physical attack can form the subject of future research.

This work was developed within several limitations. The first limitation is that the data that was analysed focussed on SH motion sensor data and for future work, Internet Protocol-based data should be incorporated to observe how the model will scale when this type of data is introduced. In addition, it would be a useful test to demonstrate the use of this model in an a real-life SH rather than through the analysis of carefully designed scenarios which represent realistic anomalous situations that IR Teams will face in real life. The Aruba SH dataset which was used in this research was obtained from a single-occupant home. Due to the fact that elderly SH occupants may receive visitors and even live in multi-occupant homes, applying the attributes identified by this work to analysing data from multi-occupant homes will be very useful going forward towards testing its broader applicability.

The Future Work section in this chapter outlines some areas that require further investigation. These areas are: extending the work to demonstrate its use in detecting the presence of pets in addition to the Human presence detection that it can be used for; building a baseline security model to represent the relationship between the internal access nodes (doors) in a SH; developing a method of identifying what kind of evidence is relevant and should therefore be captured and analysed during a Forensic Incident Response exercise in SH environments because this will prevent time wastage on collecting irrelevant information which will also take up storage space; the development of a SH Ontology which can be used to analyse the effects of the Constraints on a SH; and the development of a Proximality index as a tool for calculating the optimal number of sensors to deploy in a space so as to avoid overcrowding the space.

Appendix A

The Forensics Edge Management System (FEMS) Decision-Making Algorithm (FDMA) Core

```
activeLocation = '';
activeSens = '';
year = 2015;
day = 25;
month = 02;
lastActiveTime = 0;
threshold = seconds(10);
T=table(currSens,currLoc);
doorList = [4001,4002,4004]; %List of doors

% sensor(1).type = 'door'
% sensor(1).state = 1
% sensor(1).location = zone(1)
% sensor(1).peripherals = []
% sensor(1).mode = 1      %1 = FF, 2= MF, 3 = FM, 4 = MM
%
% sensor(2).type = 'motion'
% sensor(2).state = 0
% sensor(2).location = zone(2)
% sensor(2).peripherals = []
% sensor(2).mode = 2      %1 = FF, 2= MF, 3 = FM, 4 = MM
FullHouseSensorStructure;

%for i = 1:314
%for i = 1:66
for i = 1:900 %At row 458 the algorithm stops recognising the correct last active
     sensor.
    for j= X(:,2)
%   First, check if sensor is OFF
    fprintf(num2str(i))
    currLoc = int8(X(i,2));
    currSens = int8(X(i,2));
    currTime = datetime(year,month,day,X(i,3),X(i,4),X(i,5));
        if (X(i, 1)==0)
            fprintf(’ -Inactive: ’);
            fprintf(zone(currLoc));
            %fprintf ( ’devID=’)
            %fprintf(num2str(X(i,2)));
            fprintf(’ current sensor  = ’);
            fprintf(num2str(currLoc));
            sensor(currSens).state = 0; %sets the sensor state to be off
%             fprintf(’ sensor set to ’);
%             fprintf(num2str(sensor(currSens).state));
            if(strcmp(sensor(currSens).type,’door’))
                if(testSensorDoorCheck(sensor) == 1)%test for door anomoly
                    fprintf(’ Door Anomoly ’);
                end
            end

            %need to deactivate
        elseif X(i, 1)==1
            fprintf(’ -Active: ’);
            fprintf(zone(currLoc));
            %fprintf ( ’devID=’)
            %fprintf(num2str(X(i,2)));
            fprintf(’ current sensor  = ’);
            fprintf(num2str(currLoc));
            sensor(currSens).state = 1; %sets the sensor state to be on
            if (strcmp(activeLocation,’’))  %no previousn active location
                activeSens = currLoc;
                activeLocation = zone(currLoc);
%                 if (ismember((currLoc),(num2str(KitchProx(X(i,2))))))
%                     lastActiveTime= currTime;
            else
                if(˜strcmp(activeLocation,zone(currLoc)))
%                     if (˜strcmp((currLoc) , KitchProx(1,i)))
%                         timeDiff = currTime - lastActiveTime;
%                         timeDiff.Format = ’s’;
                        if(sensor_proximity(currLoc, activeSens) == 0)

                            if(timeDiff < threshold)

                                fprintf(’ : Warning! ’);
                                fprintf(’ current sensor  = ’);
                                fprintf(num2str(currLoc));
                                fprintf(’ last active sensor = ’);
                                fprintf(num2str(activeSens));
                                fprintf(’ Previous Active Location was ’);
                                fprintf(activeLocation);
                                fprintf(’ And current active Location is ’);
                                fprintf(zone(currLoc));
                                fprintf(’ ’);
                                fprintf(num2str(seconds(timeDiff)));
                                fprintf(’ seconds travel time’);
                                % AND nodes between the two Locations were
                                % NOT ACTIVATED, a highly abnormal situation.
                                if(sensor_proximity(currLoc, activeSens) == 1)
                                    fprintf(' : Proximal');
                                else
                                    fprintf(' : Non-Proximal!!!!!!!');
                                end
                            end
                        end
%                       end

%                  end
                end
            end



        end %end of 0 or 1 activation if

            activeSens = currLoc;
            lastActiveTime= currTime;
            currLoc = currSens;
            fprintf(' sensor data size = ');
            fprintf(num2str(size(sensor(currSens).data)));
            if(size(sensor(currSens).data) ~= [0,0])%checks to see if it has data
               fprintf(' has data ');
            end
            activeLocation = zone(currLoc);
            if(activeLocation == 0)
                fprintf(' Unregistered node detected ');
            end
        display('')
    end
end
```

Core of the Forensics Edge Management System (FEMS) Decision-Making Algorithm (FDMA) code

Appendix B

Interview Questions

Interviews were conducted as part of the *qualitative* validation process for the Internet of Things Digital Forensics Framework (IDFF). The following are the questions which were put to the interviewees.

1. **Do you know *of* the Internet of Things (IoT)?**
   This question was asked to find out about their broad awareness and /or knowledge of the IoT.

2. **What challenges, if any, does the IoT pose to forensics as far as existing frameworks are concerned?**
   This question was asked to find out if the challenges already identified by the author of this thesis were pertinent and relevant (or otherwise) and to find out if there were other challenges that the IoT presented to DF that the author had not considered.

3. **What is your opinion on Automated Forensics?**
   The method proposed by this work involves the application of Automated Forensics. This question was asked to find out the opinions of the interviewees in this regard. In summarised form, the feedback in this case was that Automated Forensics has its uses and applications but that it must not be relied upon as the singular approach to Digital Forensics (DF) investigations; that the roles that human investigators play in DF investigations is still necessary and significant.

4. **Do you think the Internet of Things Digital Forensics Framework (IDFF) Framework as presented has any merits in terms of its speed, timeliness, comprehensiveness, approach and direction?**
   This question was asked to gather specific feedback on the IDFF Framework from the interviewees.

5. **Do you consider a new, more tailored DF Framework necessary or might existing ones suffice?**
   There already exist a number of frameworks within the field of DF, some which even claim to be applicable to future environments (an example future environment is the SH as part of the IoT). This question was asked to find out if the interviewees thought that there was any need for a different, tailored Framework for the IoT or if existing ones currently meet all the requirements of DF in IoT-type environments such as Smart Homes.

6. **Any suggestions for improving the IDFF Framework as presented?**
   Each of the interviewees had been given a copy of the (draft) IDFF Framework to peruse briefly; some of the interviewees were sent the draft IDFF Framework some days before their interview - others were sent the file during the interview. The feedback received, as applicable, was used to adjust the Framework.

7. **Any other recommendations for Digital Forensics going forward in the IoT?**
   This question was to glean any further input that the interviewees might have with regard to DF as a general field and how it might be influenced by the emergence of the IoT.

Appendix C

Proximal Sensors Table

An ‘**X**’ indicates that a proximality relationship cannot be established based on the sensor readings available or sensor firings; a ‘**1**’ indicates that the nodes are proximal and a ‘**0**’ indicates non-proximality.

Sensors	**M001**	**M002**	**M003**	**M004**	**M005**	**M006**	**M007**	**M008**	**M009**	**M010**
M001	**1**	**1**	0	1	0	0	0	0	0	0
M002	**1**	**1**	0	0	0	0	0	0	0	0
M003	0	0	**1**	0	**X**	**X**	**X**	0	0	0
M004	0	0	0	**1**	**X**	0	**X**	0	0	0
M005	0	0	**X**	**X**	**1**	**X**	**X**	0	0	0
M006	0	0	**X**	0	**X**	**1**	**X**	1	0	0
M007	0	0	**X**	**X**	**X**	0	**1**	0	0	0
M008	0	0	0	0	0	**X**	0	**1**	**X**	0
M009	0	0	**X**	0	0	0	0	**X**	**1**	0
M010	0	0	0	0	0	0	0	**X**	**X**	**1**
M011	0	0	**X**	0	0	0	0	**X**	**X**	**X**
M012	0	0	0	0	0	0	0	0	**X**	**X**
M013	0	0	0	0	0	0	0	**X**	**X**	0
M014	0	0	0	0	0	0	0	0	0	0
M015	0	0	0	0	0	0	0	0	0	0
M016	0	0	0	0	0	0	0	0	0	0
M017	0	0	0	0	0	0	0	0	0	0
M018	0	0	0	0	0	0	0	0	0	0
M019	0	0	0	0	0	0	0	0	0	0
M020	0	0	0	0	0	0	0	**X**	0	0
M021	0	0	0	0	0	0	0	**X**	0	0
M022	0	0	0	0	0	0	0	**X**	0	0
M023	0	0	0	0	0	0	0	0	0	0
M024	0	0	0	0	0	0	0	0	0	0
M025	0	0	0	0	0	0	0	0	0	0
M026	0	0	0	0	0	0	0	0	0	0
M027	0	0	0	0	0	0	0	0	0	0
M028	0	0	0	0	0	0	0	0	0	0
M029	0	0	0	0	0	0	0	0	0	0
M030	0	0	0	0	0	0	0	0	0	0
M031	0	0	0	0	0	0	0	0	0	0

Table C.1: The Complete Sensor Proximals table - *Part One*

Sensors	**M011**	**M012**	**M013**	**M014**	**M015**	**M016**	**M017**	**M018**	**M019**	**M020**
M001	0	0	0	0	0	0	0	0	0	0
M002	0	0	0	0	0	0	0	0	0	0
M003	0	0	0	0	0	0	0	0	0	0
M004	0	0	0	0	0	0	0	0	0	0
M005	0	0	0	0	0	0	0	0	0	0
M006	0	0	0	0	0	0	0	0	0	X
M007	0	0	0	0	0	0	0	0	0	0
M008	**X**	0	0	0	0	0	0	0	0	X
M009	**X**	0	0	0	0	0	0	0	0	X
M010	**X**	0	0	0	0	0	0	0	0	X
M011	1	0	0	0	0	0	0	0	0	X
M012	0	1	0	0	0	0	0	0	0	X
M013	0	0	1	**X**	0	0	0	0	0	X
M014	0	0	1	1	0	0	0	**X**	**X**	X
M015	0	0	0	0	1	0	0	0	0	0
M016	0	0	0	0	0	1	0	0	0	0
M017	0	0	0	1	0	**X**	1	0	**X**	0
M018	0	0	0	0	0	0	**X**	1	**X**	0
M019	0	0	0	0	**X**	0	0	0	1	0
M020	0	0	0	**X**	0	0	0	0	0	1
M021	0	0	0	0	0	0	0	**X**	0	0
M022	0	0	0	0	0	0	0	0	0	0
M023	0	0	0	0	0	0	0	0	0	0
M024	0	0	0	0	0	0	0	0	0	0
M025	0	0	0	0	0	0	0	0	0	0
M026	0	0	0	0	0	0	0	0	0	0
M027	0	0	0	0	0	0	0	0	0	0
M028	0	0	0	0	0	0	0	0	0	0
M029	0	0	0	0	0	0	0	0	0	0
M030	0	0	0	0	0	0	0	0	0	0
M031	0	0	0	0	0	0	0	0	0	0

TABLE C.2: The Complete Sensor Proximals table - *Part Two*

Sensors	**M021**	**M022**	**M023**	**M024**	**M025**	**M026**	**M027**	**M028**	**M029**	**M030**	**M031**
M001	0	0	0	0	0	0	0	0	0	0	0
M002	0	0	0	0	0	0	0	0	0	0	0
M003	0	0	0	0	0	0	0	0	0	0	0
M004	0	0	0	0	0	0	0	0	0	0	0
M005	0	0	0	0	0	0	0	0	0	0	0
M006	0	0	0	0	0	0	0	0	0	0	0
M007	0	0	0	0	0	0	0	0	0	0	0
M008	0	0	0	0	0	0	0	0	0	0	0
M009	0	0	0	0	0	0	0	0	0	0	0
M010	0	0	0	0	0	0	0	0	0	0	0
M011	0	0	0	0	0	0	0	0	0	0	0
M012	0	0	0	0	0	0	0	0	0	0	0
M013	0	0	0	0	0	0	0	0	0	0	0
M014	**X**	0	0	0	0	0	0	0	0	0	0
M015	0	0	0	0	0	0	0	0	0	0	0
M016	0	0	0	0	0	0	0	0	0	0	0
M017	0	0	0	0	0	0	0	0	0	0	0
M018	**X**	0	0	0	0	0	0	0	0	0	0
M019	0	0	0	0	0	0	0	0	0	0	0
M020	0	0	0	0	0	0	0	0	0	0	0
M021	1	**X**	0	0	0	0	0	0		0	0
M022	0	1	**X**	0	0	0	0	0	**X**	0	0
M023	0	**X**	1	1	0	0	0	0	0	0	0
M024	0	0	0	1	0	0	0	0	0	0	0
M025	0	0	0	0	1	**X**	**X**	0	0	0	0
M026	0	0	0	0	0	1	**X**	0	0	0	0
M027	0	0	0	0	1	**X**	1	**X**	0	0	0
M028	0	**X**	**X**	0	0	**X**	**X**	1	X	0	0
M029	0	0	0	0	0	0	0	**X**	1	0	0
M030	0	0	0	0	0	0	0	**X**	**X**	1	0
M031	0	0	0	0	0	0	0	0	0	0	1

TABLE C.3: The Complete Sensor Proximals table - *Part Three*

Appendix D

Smart Home Relationship Constraints Ontology

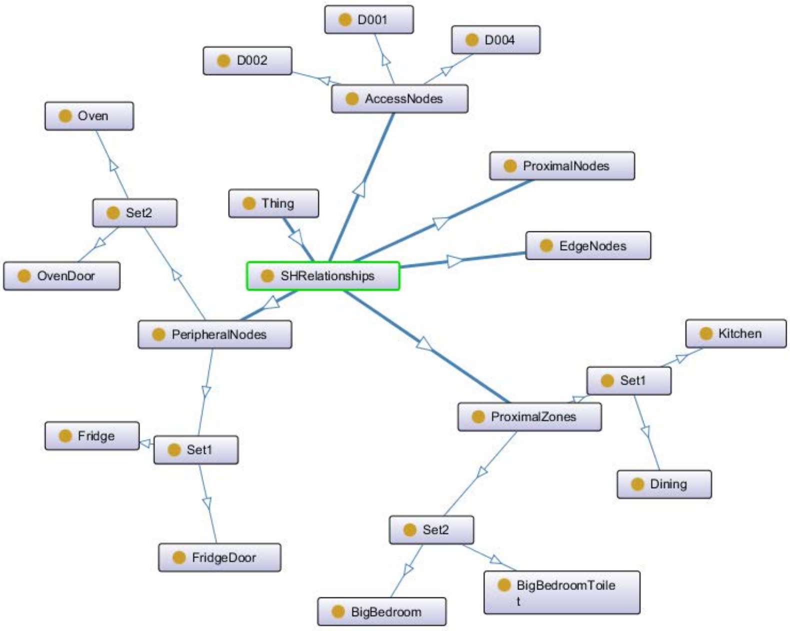

FIGURE D.1: Smart Home Relationship Constraints Ontology

References

[1] D. J. Cook. Learning Setting-Generalized Activity Models for Smart Spaces. *Intelligent Systems, IEEE*, 27(1):32–38, 2012. ID: 1.

[2] C. Sik-Lányi, E.-J. Hoogerwerf, K. Miesenberger, and P. Cudd. Assistive Technology - Building Bridges. In *Studies in Health Technology and Informatics*, volume 217, 2015. ISBN 978-1-61499-565-4. URL http://www.iospress.nl/book/assistive-technology/.

[3] J.L. Wiles, A. Leibing, N. Guberman, J. Reeve and R.ES Allen. The Meaning of "Ageing in Place" to Older People. *The gerontologist*, page gnr098, 2011.

[4] C. Wilson, T. Hargreaves, and R. Hauxwell-Baldwin. Smart Homes and their Users: a Systematic Analysis and Key Challenges. *Personal and Ubiquitous Computing*, 19(2):463–476, 2015.

[5] T. Koskela and K. Väänänen-Vainio-Mattila. Evolution Towards Smart Home environments: Empirical Evaluation of Three User Interfaces. *Personal and Ubiquitous Computing*, 8(3-4):234–240, 2004.

[6] E. Oriwoh and M. Conrad. *Presence Detection from Smart Home Motion Sensor Datasets: A Model*, pages 1249–1255. Springer International Publishing, 2016. ISBN 978-3-319-32703-7. doi: 10.1007/978-3-319-32703-7_240. URL http://dx.doi.org/10.1007/978-3-319-32703-7_240.

[7] S. Kumar and E.H. Spafford. An Application of Pattern Matching in Intrusion Detection. 1994.

[8] A. Patcha and J. Park. An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends. *Computer networks*, 51(12):3448–3470, 2007.

[9] A. Lotfi, C. Langensiepen, S.M. Mahmoud and M.J. Akhlaghinia. Smart Homes for the Elderly Dementia Sufferers: Identification and Prediction of Abnormal Behaviour. *Journal of Ambient Intelligence and Humanized Computing*, 3(3):205–218, 2012.

[10] H. K. Ali and I. G. Amalarethinam. Detecting abnormality in activities performed by people with dementia in a smart environment. *International Journal of Computer Science and information Technologies, IJCSIT*, 5(2), 2014-05-02 2014. URL http://www.ijcsit.com/docs/Volume%205/vol5issue02/ijcsit20140502337.pdf.

[11] V.R. Jakkula and D.J Cook. Detecting anomalous sensor events in smart home data for enhancing the living experience. *Artificial intelligence and smarter living*, 11:07, 2011.

[12] K. Ashton. That 'Internet of Things' Thing. *RFiD Journal*, 22:97–114, 2009.

[13] R.H. Weber. Accountability in the internet of things. *Computer Law Security Review*, 27(2):133–138, 4 2011.

[14] J. Solomon and E. Lattimore. Computer forensics, 2006. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.9392&rep=rep1&type=pdf.

[15] P.C. Jain, A. Noor and V.K. Sharma. Internet of things - an introduction. *Fourth Annual Seminar of CDAC Noida Technologies (ASCNT-2011)*, 2011. URL http://www.cdacnoida.in/ascnt2011/Presentations%5CUC%5C1_IoT%20ASCNT-11-4%20Slides.pdf.

[16] M.H. José, B.V. Jesús, M. Luis, A.G. José, P. Mirko, A.H.G. Luis and P. Jan. *Smart Cities at the Forefront of the Future Internet*, pages 447–462. The future internet. Springer, 2011.

[17] J. Bradley, J. Barbier and D. Handler. Embracing the Internet of Everything To Capture Your Share of $14.4 Trillion: More Relevant, Valuable Connections will improve Innovation, Productivity, Efficiency & Customer Experience. Technical report, 2013. URL http://www.cisco.com/web/about/ac79/docs/innov/IoE_Economy.pdf.

[18] Sukanya, P. and Gayathri, K.S. An Unsupervised Pattern Clustering Approach for Identifying Abnormal User Behaviors in Smart Homes 1, 2013.

[19] Z. A. Baig. On the use of Pattern Matching for rapid Anomaly Detection in Smart Grid Infrastructures. In *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pages 214–219, 2011. ID: 1.

[20] M.T. Dlamini, M.M. Eloff and J.H.P. Eloff. Internet of Things: Emerging and Future Scenarios from an Information Security Perspective. Southern Africa Telecommunication Networks and Applications Conference, 2009.

[21] A. Laurie. Practical Attacks against RFID. *Network Security*, 2007(9):4–7, 9 2007.

[22] H.J. Wang C. Guo and W. Zhu. Smart-phone Attacks and Defenses. In *HotNets III*, 2004.

[23] M. R. Rieback, B. Crispo, and A. S. Tanenbaum. The Evolution of RFID Security. *IEEE Pervasive Computing*, 5(1):62–69, 2006.

[24] V. Srinivasan, J. Stankovic and K. Whitehouse. Protecting your daily in-home activity information from a wireless snooping attack. In *Proceedings of the 10th international conference on Ubiquitous computing*, UbiComp '08, pages 202–211, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-136-1. URL http://doi.acm.org/10.1145/1409635.1409663.

[25] T. Roosta, S. Shieh and S. Sastry. Taxonomy of security attacks in sensor networks and countermeasures. In *The First IEEE International Conference on System Integration and Reliability Improvements*, volume 25, page 94, 2006.

[26] D. Kozlov, J. Veijalainen and Y. Ali. Security and privacy threats in iot architectures. In *Proceedings of the 7th International Conference on Body Area Networks*, BodyNets '12, pages 256–262, Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering; ICST, Brussels, Belgium, Belgium, 2012. ICST. ISBN 978-1-936968-60-2. URL http://dl.acm.org/citation.cfm?id=2442691.2442750.

[27] E.K. Wang, Y. Ye, X. Xiaofei, S.M. Yiu, L.C.K. Hui and K. P. Chow. Security Issues and Challenges for Cyber Physical System. In *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, pages 733–738, 2010. ID: 1.

[28] F. Johannes, G. Dragan et al. *Foundations of Rule Learning*. Springer Science & Business Media, 2012. ISBN 978-3-540-75196-0.

[29] C. Chen, B. Das and D.J. Cook. A Data Mining Framework for Activity Recognition in Smart Environments. In *Proceedings of the 2010 Sixth International Conference on Intelligent Environments*, IE '10, pages 80–83, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4149-5. doi: 10.1109/IE.2010.22. URL http://dx.doi.org/10.1109/IE.2010.22.

[30] Bourobou, S.T.M. and Yoo, Y. User Activity Recognition in Smart Homes Using Pattern Clustering Applied to Temporal ANN Algorithm. *Sensors*, 15(5): 11953–11971, 2015. ISSN 1424-8220. doi: 10.3390/s150511953. URL http://www.mdpi.com/1424-8220/15/5/11953.

[31] J.F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[32] L. Chen, C.D. Nugent and Hui Wang. A Knowledge-driven Approach to Activity Recognition in Smart Homes. *Knowledge and Data Engineering, IEEE Transactions on*, 24(6):961–974, 2012.

[33] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur and Y. Katz. Pellet: A Practical OWL-DL Reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2):51–53, 2007.

[34] T. Magherini and A. Fantechi and C. D. Nugent and E. Vicario. Using Temporal Logic and Model Checking in Automated Recognition of Human Activities for Ambient-Assisted Living. *IEEE Transactions on Human-Machine Systems*, 43(6): 509–521, Nov 2013. ISSN 2168-2291. doi: 10.1109/TSMC.2013.2283661.

[35] Y. Chen, C. Chen, W. Peng and W. Lee. Mining Correlation Patterns among Appliances in Smart Home Environment. In *Advances in Knowledge Discovery and Data Mining*, volume 8444 of *Lecture Notes in Computer Science*, pages 222–233. Springer International Publishing, 2014. ISBN 978-3-319-06604-2. doi: 10.1007/978-3-319-06605-9_19. URL http://dx.doi.org/10.1007/978-3-319-06605-9_19.

[36] Y. Chen, W. Peng, J. Huang, and W. Lee. Significant Correlation Pattern Mining in Smart Homes. *ACM Trans. Intell. Syst. Technol.*, 6(3):35:1–35:23, April 2015. ISSN 2157-6904. doi: 10.1145/2700484. URL http://doi.acm.org/10.1145/2700484.

[37] Y. Chen, J. Jiang, W. Peng and S. Lee. An Efficient Algorithm for Mining Time Interval-based Patterns in Large Database. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 49–58, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0099-5. doi: 10.1145/1871437.1871448. URL http://doi.acm.org/10.1145/1871437.1871448.

[38] S. Wu and Y. Chen. Mining Nonambiguous Temporal Patterns for Interval-Based Events. volume 19, pages 742–758, June 2007. doi: 10.1109/TKDE.2007.190613.

[39] D. Patel, W. Hsu, and M.L. Lee. Mining Relationships Among Interval-based Events for Classification. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 393–404, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-102-6. doi: 10.1145/1376616.1376658. URL http://doi.acm.org/10.1145/1376616.1376658.

[40] I. Paredes-Oliva, I. Castell-Uroz, P. Barlet-Ros, X. Dimitropoulos and J. Sole-Pareta. Practical anomaly detection based on classifying frequent traffic patterns. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 49–54, March 2012. doi: 10.1109/INFCOMW.2012.6193518. URL http://dx.doi.org/10.1109/INFCOMW.2012.6193518.

[41] E.A. Leicht and M.E.J. Newman. Community Structure in Directed Networks. *Physical review letters*, 100(11):118703, 2008.

[42] L. Yao, Q.Z. Sheng, B.J. Gao, A.H.H. Ngu, and X. Li. A Model for Discovering Correlations of Ubiquitous Things. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 1253–1258. IEEE, 2013.

[43] G. Okeyo, L. Chen, H. Wang, and R. Sterritt. Dynamic Sensor Data Segmentation for Real-time Knowledge-driven Activity Recognition. *Pervasive and Mobile Computing*, 10, Part B(0):155–172, 2 2014.

[44] M. Novak, M., Binas and F. Jakab. Unobtrusive Anomaly Detection in Presence of Elderly in a Smart-Home Environment. In *ELEKTRO, 2012*, pages 341–344, May 2012. doi: 10.1109/ELEKTRO.2012.6225617.

[45] M. Novák, F. Jakab, L. Lain, K. Salman, Y. Yi, Y. Zhou, X. Fu, F. Shen, R.S. Higa, R.F.L Chavez, et al. Anomaly Detection in User Daily Patterns in Smart-Home Environment.

[46] D. Siman M. Andrášová D. Šimšík, A. Galajdová and R. Balog. Information Technology Supporting Daily Activities of Seniors. *Gerontechnology*, 11(2):306, 2012.

[47] K. Sequeira and M. Zaki. ADMIT: Anomaly-based Data Mining for Intrusions. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 386–395, New York, NY, USA, 2002. ACM. ISBN 1-58113-567-X. doi: 10.1145/775047.775103. URL http://doi.acm.org/10.1145/775047.775103.

[48] L. Chen and C. Nugent. Ontology-based Activity Recognitionin Intelligent Pervasive Environments. *International Journal of Web Information Systems, IJWIS*, 5 (4):410–430, 2009. URL http://scm.ulster.ac.uk/~lchen/ijwis_ontoAR.pdf.

[49] E. Oriwoh and M. Conrad. 'Things' in the Internet of Things: Towards a Definition. volume 4(1), pages 1–5, 2015. doi: 10.5923/j.ijit.20150401.01. URL http://article.sapub.org/10.5923.j.ijit.20150401.01.html.

[50] T. Espiner. MPs: UK 'needs to prepare' for Driverless Cars. *BBC Business*, 6 March 2015 2015. URL http://www.bbc.co.uk/news/business-31759071.

[51] Z. Ji and Q. Anwen. The Application of Internet of Things (IoT) in Emergency Management System in China. In *Technologies for Homeland Security (HST), 2010 IEEE International Conference on*, pages 139–142. IEEE, 2010.

[52] M.C. Domingo. An Overview of the Internet of Things for People with Disabilities. *Journal of Network and Computer Applications*, 35(2):584–596, 3 2012.

[53] A. J. Jara, A. F. Alcolea, M. A. Zamora, A. F. G. Skarmeta, and M. Alsaedy. Drugs Interaction Checker based on IoT. In *Internet of Things (IOT), 2010*, pages 1–8, 2010. ID: 1.

[54] E. Oriwoh, P. Sant and G. Epiphaniou. Guidelines for Internet of Things Deployment Approaches: The Thing Commandments. *Procedia Computer Science*, 21(0): 122–131, 2013.

[55] P. Najera R. Roman and J. Lopez. Securing the internet of things. *Computer*, 44 (9):51–58, 2011. ID: 1.

[56] M.L. Rustad. Private Enforcement of Cybercrime on the Electronic Frontier. *S.Cal.Interdisc.LJ*, 11:63, 2001.

[57] R.H. Weber. Internet of things - need for a new legal environment? *Computer Law Security Review*, 25(6):522–527, 11 2009.

[58] H. Suo, J. Wan, C. Zou and J. Liu. Security in the Internet of Things: A Review. In *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, volume 3, pages 648–651, 2012. ID: 1.

[59] Computer Misuse Act 1990 (1990 c 18), .

[60] Theft Act 1968 (1968 c 60), .

[61] D. Evans. The internet of things how the next evolution of the internet is changing everything. *CISCO white paper*, 2011.

[62] Data Privacy Integrity Advisory Committee. The use of rfid for human identity verification.

[63] United States Department of State Bureau of Consular Affairs. The U.S. Electronic Passport. URL http://travel.state.gov/passport/passport_2498.html.

[64] P. Peris-Lopez, J. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda. RFID Systems: A Survey on Security Threats and Proposed Solutions. In *Personal Wireless Communications*, pages 159–170. Springer, 2006.

[65] L. Atzori, A. Iera and G. Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 10/28 2010.

[66] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. Jubert, M. Mazura, M. Harrison, and M. Eisenhauer 10. Internet of things strategic research roadmap. *Internet of Things: Global Technological and Societal Trends*, page 9, 2009.

[67] M. Meingast, J. King, and D. K. Mulligan. Embedded rfid and everyday things: A case study of the security and privacy risks of the u.s. e-passport. In *RFID, 2007. IEEE International Conference on*, pages 7–14, 2007. ID: 1.

[68] D. Engels. Securing the next wave of technology. In *National Institute of Standards and Technology (NIST) Workshop on Cryptography for Emerging Technologies and Applications (CETA)*, 2011. URL http://csrc.nist.gov/groups/ST/CETA_2011/presentations/engels.pdf.

[69] European Commission. Internet of things in 2020-a roadmap for the future. *European Platform for Smart Systems Integration*, 2008.

[70] The Web of Things. 2009. URL http://www.caba.org/resources/Documents/IS-2009-121.pdf.

[71] S. L. Garfinkel, A. Juels, and R. Pappu. Rfid privacy: an overview of problems and proposed solutions. *Security Privacy, IEEE*, 3(3):34–43, 2005. ID: 1.

[72] K. R. Foster and J. Jaeger. Ethical implications of implantable radiofrequency identification (rfid) tags in humans. *American Journal of Bioethics*, 8(8):44–48, 08 2008. URL http://0-search.ebscohost.com.brum.beds.ac.uk/login.aspx?direct=true&db=rzh&AN=2010045925&site=ehost-live&scope=site. ID: 2010045925.

[73] B. Benhamou. Organizing internet architecture. *Fostering Public Sector Performance in Europe*, page 40, 2007.

[74] Yingtao Jiang, Lei Zhang, and Ling Wang. Wireless Sensor Networks and the Internet of Things, 2013. URL http://dx.doi.org/10.1155/2013/589750.

[75] I. Sommerville. *Models for responsibility assignment*, pages 165–186. Responsibility and dependable systems. Springer, 2007.

[76] An Act. Health Insurance Portability and Accountability Act of 1996. *Public Law*, 104:191, 1996.

[77] R. Lock, T. Storer, I. Sommerville and Gordon Baxter. Responsibility modelling for risk analysis. 2010.

[78] E. Oriwoh and P. Sant. The Forensics Edge Management System: A Concept and Design. In *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*, pages 544–550, 2013. ID: 1.

[79] E. Oriwoh, D. Jazani, G. Epiphaniou and P. Sant. Internet of Things Forensics: Challenges and Approaches. In *Collaborative Computing: Networking, Applications*

and Worksharing (Collaboratecom), 2013 9th International Conference Conference on, pages 608–615, 2013. ID: 1.

[80] The WritePass journal. Sample Dissertation Methodology. In *How to Write a Dissertation: Methodology*, Apr 2013. URL `http://writepass.com/journal/2013/04/sample-dissertation-methodology/`.

[81] E. Oriwoh and G. Williams. *Internet of Things: The Argument for Smart Forensics.* Handbook of Research on Digital Crime, Cyberspace Security, and Information Assurance. IGI-Global Publishing, 2014.

[82] Felix C. Freiling and Bastian Schwittay. A common process model for incident response and computer forensics. *IMF*, 7:19–40, 2007.

[83] M. I. Cohen, D. Bilby, and G. Caronni. Distributed Forensics and Incident response in the Enterprise. *Digital Investigation*, 8, Supplement(0):S101–S110, 8 2011.

[84] C. Yeoh, H. Tan, C. Kok, H. Lee and H. Lim. e2Home: A Lightweight Smart Home Management System. In *Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on*, volume 1, pages 82–87, Nov 2008. doi: 10.1109/ICCIT.2008.191.

[85] M. Ciampa. Security+ Guide to Network Security Fundamentals, Fifth Edition. 2014.

[86] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, and R. Steinmetz. On the accuracy of Appliance Identification based on Distributed load metering Data. In *Sustainable Internet and ICT for Sustainability (SustainIT), 2012*, pages 1–9, Oct 2012.

[87] I. Chrisment, A. Couch, R. Badonnel and M. Waldburger. *Managing the Dynamics of Networks and Services: 5th International Conference on Autonomous Infrastructure, Management, and Security, AIMS 2011, Nancy, France, June 13-17, 2011, Proceedings*, volume 6734. Springer, 2011.

[88] D. Niyato, L. Xiao and P. Wang. Machine-to-machine communications for home energy management system in smart grid. *Communications Magazine, IEEE*, 49 (4):53–59, 2011. ID: 1.

[89] D. Han and J. Lim. Smart home energy management system using ieee 802.15.4 and zigbee. *Consumer Electronics, IEEE Transactions on*, 56(3):1403–1410, 2010. ID: 1.

[90] L.S. Miller, C.L. Brown, M.B. Mitchell, and G.M. Williamson. Activities of Daily Living Are Associated With Older Adult Cognitive Status Caregiver Versus Self-Reports. *Journal of Applied Gerontology*, 32(1):3–30, 2013.

[91] J.W. Grzymala-Busse. Rule induction. pages 249–265, 2010. doi: 10.1007/978-0-387-09823-4_13. URL http://dx.doi.org/10.1007/978-0-387-09823-4_13.

[92] Max Bramer. Automatic induction of classification rules from examples using N-Prism. In *Research and development in intelligent systems XVI*, pages 99–121. Springer, 2000.

Acronyms and Glossary

A

AAL **A**mbient **A**ssisted **L**iving, a system that implements sensing of the environment and decision-making based on the technologies and systems available in Smart Homes. 1

AD **A**nomaly **D**etection, the use of tools and methodologies to investigate anomalous occurrences in Smart Home networks, either in real-tie or from Smart Home datasets. 13

ADL **A**ctivities **o**f **D**aily **L**iving are all activities carried out by people daily. 1, 7, 8

AT **A**ssistive **T**echnology, Technology-based solutions and services provided to the disabled and otherwise challenged. 1

C

CPE **C**yber-**P**hysical **E**nvironments, Environments that are composed of both Cyber and Physical components, co-operating to provide services for human and societal needs. 7

CPS **C**yber-**P**hysical **S**ystems, systems which are both Cyber and Physical in nature. 22

F

FDMA **F**orensics **D**ecision-**M**aking **A**lgorithm, an anomaly-detection algorithm that males a decision about whether an occurrence or scenario is anomalous or otherwise. 8, 58

FEMS **F**orensics **E**dge **M**anagement System, A system that autonomously performs security and digital forensics functions within Smart Home environments. These functions include network management, data mining, logging and alert management. 8, 49

I

IoT **I**nternet **o**f **T**hings, describes the potential for the interconnection of every feasible relevant thing to every other feasible and necessary things, living or not, animate or otherwise, depending on requirements. 8

S

SH **S**mart **H**omes, these are homes that are equipped with tools and technologies to sense and actuate in such ways as to provide assistance to their occupants with their Activities of Daily Living. 1, 13