

Evaluating a Single-Modality Ground-Based  
Activity Recognition Sensor for Human Inclusion  
into Digital Systems

ADAM N. A. RAMEKA

2018

A thesis submitted to Auckland University of Technology in partial  
fulfilment of the requirements for the degree of Master of Creative  
Technologies.

# TABLE OF CONTENTS

<i>ABSTRACT</i> .....	<i>IV</i>
<i>ATTESTATION OF AUTHORSHIP</i> .....	<i>V</i>
<i>LIST OF FIGURES</i> .....	<i>VI</i>
<i>LIST OF TABLES</i> .....	<i>VII</i>
<i>ACKNOWLEDGEMENTS</i> .....	<i>IX</i>
<b>1 INTRODUCTION</b> .....	<b>1</b>
<b>2 REVIEW OF LITERATURE</b> .....	<b>5</b>
<b>2.1 IoT Definitions</b> .....	<b>5</b>
2.1.1 Ubiquitous computing.....	7
<b>2.2 Sensory Systems</b> .....	<b>10</b>
2.2.1 Early IoT sensory systems .....	10
2.2.2 Smartphones – the ubiquitous future?.....	12
2.2.3 Smart environments .....	14
<b>2.3 Smart Objects</b> .....	<b>16</b>
2.3.1 Virtual representation .....	17
<b>2.4 Human-centred IoT</b> .....	<b>19</b>
2.4.1 Adoption .....	20
2.4.2 Autonomy .....	22
2.4.3 Creation and design .....	25
<b>2.5 Smart Floors</b> .....	<b>27</b>
2.5.1 Carpet integrated architectures .....	30
2.5.2 Physical activity recognition .....	34
<b>2.6 Big Data Analytics</b> .....	<b>37</b>
2.6.1 Machine learning .....	42
2.6.2 Machine learning in activity recognition .....	44
<b>2.7 Summary</b> .....	<b>48</b>
<b>3 METHODOLOGY</b> .....	<b>49</b>
<b>3.1 Research Objectives</b> .....	<b>49</b>

<b>3.2</b>	<b>Research Methodology .....</b>	<b>50</b>
<b>3.3</b>	<b>Research Design.....</b>	<b>51</b>
3.3.1	Artefact.....	52
3.3.2	Problem relevance .....	53
3.3.3	Design evaluation.....	53
3.3.4	Research contribution.....	54
3.3.5	Research rigor.....	55
3.3.6	Design as a search process .....	57
3.3.7	Communicate research .....	58
<b>4</b>	<b><i>ARTEFACT DESIGN AND IMPLEMENTATION.....</i></b>	<b>60</b>
<b>4.1</b>	<b>Use-case: Workplace Environment .....</b>	<b>60</b>
<b>4.2</b>	<b>Initial Exploration.....</b>	<b>63</b>
4.2.1	Semi-conductive polymer composite for piezoresistive pressure sensing .....	64
4.2.2	Electrode matrix and ground sensor configuration .....	64
4.2.3	Data acquisition and processing .....	67
4.2.4	Initial iteration summary.....	70
<b>4.3</b>	<b>Second Iteration.....</b>	<b>71</b>
4.3.1	Data processing .....	71
4.3.2	Semi-conductive polymer composite for piezoresistive pressure sensing .....	72
4.3.3	Electrode matrix and ground sensor configuration .....	74
4.3.4	Second iteration summary .....	76
<b>4.4</b>	<b>Third Iteration.....</b>	<b>76</b>
4.4.1	Electrode matrix and ground sensor configuration .....	76
4.4.2	Microcontroller and multiplexers .....	79
4.4.3	Third iteration summary.....	84
<b>4.5</b>	<b>Fourth Iteration .....</b>	<b>84</b>
4.5.1	Data acquisition .....	85
4.5.2	Data processing .....	86
4.5.3	Fourth iteration summary .....	88

4.6	<b>Final Iteration</b> .....	88
4.6.1	Data processing .....	89
4.6.2	Final iteration summary .....	89
5	<b>ACTIVITY RECOGNITION EVALUATION VIA MACHINE LEARNING</b>	
	<b>CLASSIFICATION</b> .....	90
5.1	<b>Data Analysis Pre-processing</b> .....	90
5.1.1	J48 decision trees .....	92
5.1.2	Random forest.....	93
5.1.3	Naïve bayes.....	94
5.1.4	Support vector machine .....	96
5.2	<b>Algorithm comparison and evaluation</b> .....	97
5.2.1	Accuracy .....	97
5.2.2	Efficiency .....	105
5.3	<b>Summary</b> .....	112
6	<b>LIMITATIONS, FUTURE RESEARCH AND CONCLUSION</b> .....	114
6.1	<b>Limitations</b> .....	115
6.1.1	Sample size .....	115
6.1.2	Limited activities .....	116
6.1.3	Dataset and algorithm training.....	116
6.1.4	IoT inclusion .....	117
6.1.5	Real-time vs post-processing of data for classification .....	117
6.2	<b>Future Research</b> .....	118
6.2.1	Physical artefact design.....	118
6.2.2	Sensor architecture .....	119
6.2.3	Machine learning for AR classification .....	120
6.2.4	Stakeholder engagement .....	121
6.2.5	IoT value .....	121
6.3	<b>Conclusion</b> .....	122
7	<b>REFERENCES</b> .....	123
8	<b>APPENDIX A: WEKA RESULTS</b> .....	140

## ABSTRACT

With the proliferation of relatively cheap Internet of Things (IoT) devices, Smart Environments have been highlighted as an example of how the IoT can make our lives easier. Each of these ‘things’ produces data which can work in unison with other devices to create an environment that can react to its users. Machine learning makes use of this data to make inferences about our habits and activities, such as our buying preferences or likely commute destinations. However, this level of human inclusion within the IoT relies on indirect inferences from the usage of these devices or services. Alternatively, Activity Recognition is already a widely researched domain and could provide a more direct way of including humans within this system. With intended application in the IoT, this research explores the feasibility of using a cost effective, unobtrusive, single modality ground-based sensor to track subtle direct, and indirect pressure changes. With the subsequent data, a number of machine learning classification approaches are utilised to assess the sensors performance in activity recognition. The results indicate that accuracy in Activity Recognition classification is generally high and provides a basis for further investigation as an interface to more complex digital systems, such as the IoT.

## ATTESTATION OF AUTHORSHIP

“I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.”

A handwritten signature in black ink, appearing to read 'A. N. A. Rameka', with a long horizontal flourish extending to the right.

Adam N. A. Rameka

## LIST OF FIGURES

<b>Figure 1:</b> Life Cycle of a Data Mining project (Witten et al., 2016, p. 29)...	41
<b>Figure 2:</b> The DSR Process (Peppers et al., 2007, p. 54).....	51
<b>Figure 3:</b> Knowledge contribution assessment (Gregor & Hevner, 2013, p. 345) .....	55
<b>Figure 4:</b> Information system research framework (Marheineke, 2016, p. 14) .....	56
<b>Figure 5:</b> Initial implementation, assessing the sensitivity of the sensor using the rolling motion of the tape .....	65
<b>Figure 6:</b> A single conductive foam piece, one of many placed between the copper electrodes .....	65
<b>Figure 7:</b> Top layer and bottom layer with copper rows and columns exposed. Wires are soldered to each one and connected to the Arduino .....	66
<b>Figure 8:</b> Damaged exhibited on the top surface after pivoting of a chair castor .....	67
<b>Figure 9:</b> Visualisation of a fist pressed onto the surface of the sensor, with indications of a malfunctioning copper contact within the system .....	72
<b>Figure 10:</b> Comparison of material thickness, with Velostat in the foreground and conductive foam behind it.....	73
<b>Figure 11:</b> Rubber mat used in replacement of top and bottom layers .....	74
<b>Figure 12:</b> Comparison of the width of copper columns and rows between the initial sensor (bottom) and the second iteration (top) .....	75
<b>Figure 13:</b> Carpet and backing, replacing the top rubber layer.....	77
<b>Figure 14:</b> Sensor placed underneath chair in use-case scenario. Numbers indicate the relative position of electrodes .....	79
<b>Figure 15:</b> Arduino Uno replaced by the Teensy 3.6 .....	81
<b>Figure 16:</b> The set of multiplexers used for connection to the sensor rows and columns, with multiplexers 1 – 4 used for the top layer, and multiplexers 5 – 8 used for the bottom layer. The microcontroller is removed. ....	82
<b>Figure 17:</b> Example of pressure map produced by the sensor, with further options to initiate different modes in the GUI .....	86

# LIST OF TABLES

<b>Table 1:</b> Current technological frameworks: existing and missing IoT features (Atzori et al., 2017, p. 136) .....	8
<b>Table 2:</b> High-Level Smart Object Metamodel (Fortino, Guerrieri, Russo, & Savaglio, 2015, p. 1298) .....	19
<b>Table 3:</b> Benefits of data analytics for IoT applications (Ahmed et al., 2017, p. 466) .....	40
<b>Table 4:</b> Design-Science Research Guidelines(Hevner et al., 2004, p. 83) ..	52
<b>Table 5:</b> Overview of the seven datasets used in the machine learning evaluation of AR .....	91
<b>Table 6:</b> Percentage correctly classified with a significance level of 0.001 (two-tailed confidence level of 99.9%) .....	98
<b>Table 7:</b> Staged dataset with 2800 instances and 60/40 split show identical tree structure irrespective of pruning. ....	98
<b>Table 8:</b> Free dataset with 17096 instances and 60/40 split show the effect of pruning on more complex data .....	99
<b>Table 9:</b> Staged dataset with 4000 (2400 training and 1600 test) instances and using a holdout method of 60/40 to show the effect of pruning different data .....	99
<b>Table 10:</b> Free dataset with 5516 (randomized exclusive 3309 training and 1985 test) instances and using a holdout method of 60/40 to show the effect of different values in conjunction with pruning .....	100
<b>Table 11:</b> Free dataset with 5516 (randomized exclusive 3309 training and 1985 test) instances and using a holdout method of 60/40 to show comparing RandomForest and SVM in WEKA’s default configuration .....	101
<b>Table 12:</b> Kappa statistic for each algorithm across all datasets when using a two-tailed significance of 0.001 and 10-fold cross-validation.....	101
<b>Table 13:</b> Comparison of Gaussian and Supervised Discretization Naïve Bayes with dataset 5 .....	102
<b>Table 14:</b> Comparison of Gaussian and Supervised Discretization Naïve Bayes with dataset 7 .....	102
<b>Table 15:</b> Confusion matrix for Supervised Discretization Naïve Bayes using dataset 7 .....	103

<b>Table 16:</b> False negative rates across all datasets and algorithms using a significance factor of 0.001(two tailed) 10-fold cross validation .....	103
<b>Table 17:</b> False positive rates across all datasets and algorithms using a significance factor of 0.001(two tailed) 10-fold cross validation .....	104
<b>Table 18:</b> Percentage correctly classified for all algorithms after optimization across all datasets with a significance level of 0.05 (two-tailed), using 60/40 holdout train and test sets .....	104
<b>Table 19:</b> Percentage correctly classified for all algorithms after optimization across all datasets with a significance level of 0.05 (two-tailed), using 10-fold cross validation.....	105
<b>Table 20:</b> Comparison of CPU time (seconds) taken for model training ...	107
<b>Table 21:</b> Time taken training an SVM model using normalization, standardization, and an alternative kernel, when used with the 60/40 holdout test with dataset 5.....	108
<b>Table 22:</b> Time taken evaluating the SVM models from Table 21 .....	109
<b>Table 23:</b> Time taken (seconds) to classify the test set of each dataset.....	110
<b>Table 24:</b> Model size (bytes) of each algorithm and dataset (Java Serialized Objects).....	111

## ACKNOWLEDGEMENTS

I would firstly like to thank my supervisors Andy Connor and Jan Kruse for their guidance and contribution during this research process. Their support and wisdom has proved invaluable in navigating me through this journey, and I am deeply grateful for this. Ngā mihi nui.

I would also like to acknowledge AUT and specifically the Colab faculty for my acceptance into the programme and the opportunity to pursue this research. From the support of the academic staff, to my peers who shared this journey with me. Thank you.

My thanks also goes to telecommunication company Spark NZ whom, in collaboration with AUT, provided the funding for this research. Only having this joint investment into my research from both academic and commercial entities has made this experience viable for me, and I am greatly appreciative of this opportunity. Thank you again.

I would also like to thank my parents, Sonny and Sharon, for instilling in me the value of education at a young age. He tina ki runga, he tāmōre ki raro. Your guidance, from school and through my degrees, has led me to this accomplishment.

I would also like to thank my whānau for the constant support. To Byron and Elizabeth, thank you for being the extra hand, helping with my whānau and being there when I can't. To Kayley and Dana, your support in caring for the little ones is deeply appreciated too. I cannot thank you all enough. Ehara taku toa, he takitahi, he toa takitini.

To my children, Ayvah and Adrian, thank you for providing the drive to complete this research. I hope I inspire you to achieve in your future pursuits. Whāia te iti kahurangi ki te tūohu koe, me he maunga teitei.

Lastly, to my fiancé Shannan. Words cannot express my gratitude to you during our journey through this. Your support in allowing me to complete this is immeasurable. E kore e mimiti te aroha mōu.

Tēnā rawa atu koutou.

# 1 INTRODUCTION

Computerised Technology has become an integral part of a modern society. Proof can be seen in the need for people to have such technology on their person at all times, with an estimated 4.77 billion smartphone users worldwide (“Number of mobile phone users,” n.d.). Furthermore, for the first time since mobile computer technology has existed, mobile platforms have overtaken the traditional personal computer to become the device of choice for those using the internet (“Mobile and tablet internet usage,” 2016). Computerised technology is in fact so prevalent within everyday life of a modern society that mental disorders and even deaths are attributed to the overuse and misuse of some of these technologies (Dossey, 2014; Rothberg et al., 2010).

Despite the number of devices with internet connectivity already in use, the Internet of Things (IoT) promises to further envelop the entire globe with even more bits of technological wizardry than ever before. Industry experts estimate the IoT will consist of more than 8.4 billion connected things in 2017, and 20.4 billion by 2020 (“Gartner Says 8.4 Billion Connected,” n.d.).

Currently, most of the devices consisting of the IoT are used in industrial applications; highly automated systems, smart cities and buildings, as well as manufacturing and supply chain management (Da Xu, He, & Li, 2014). In this regard, the data produced by the IoT and its accompanying devices help gain further insight into the applicable domain. For example, farmers are using the IoT and devices to obtain data that effects the productivity of their crops (Yan-e, 2011).

In 2005, the United Nations recognized the importance of this emerging technology by issuing their first report related to the IoT by the International Telecommunication Union (ITU). In this report, they described the IoT as being “*anytime, any place* connectivity for *anyone*,” further adding that “we will now have connectivity for *anything*.”

Connections will multiply and create an entirely new dynamic network of networks – an Internet of Things” (Strategy, ITU & Unit Policy, 2005, p. 2).

If this IoT definition given by the United Nations is a true description of what the IoT is perceived to be, the IoT cannot be limited to the domain of industrial applications only, but must exist in more mundane and common aspects of society as well. For this to occur, it means the inclusion of an entity that is consistently overlooked in IoT discussion; people. As Shin (2014) notes, “the discussion of the IoT so far has been predominantly focused on the technical aspect of design, such as network development” (p. 520). Even IoT solutions focused on affecting people, such as “the design of products, services, and applications often are driven by technological opportunities rather than the underlying needs” (Salim & Haque, 2015, p. 32).

Incorporating people into the IoT in order to create a system true to the United Nations definition requires a modality shift in human and machine interfaces. For example, many current interfaces between humans and the IoT exist in the confines of object interaction, or through an objects based user-interface on a smartphone app. These neglect the implied ubiquity of the IoT and limit accessibility. Some argue that these types of interfaces are “particularly problematic for the mainstreaming of IoT products and services” creating terms such as “the “app trap”: the tendency for each connected thing to develop and require its own smartphone application” (Cerf & Senges, 2016, p. 37).

Furthermore, these interfaces do not address the invisibility and subsequent era of automation and autonomy that the IoT will facilitate. In fact, it is suggested that interfaces which demand our attention and time are simply “digital chores” which have to be attended to (Krishna, 2015). Rather, what is required to achieve an automated and autonomous future is an interface that alleviates the need for conscious interaction. Therefore, it seems necessary that the interface between humans and the IoT should be created with the same attributes that define the IoT; invisibility, ubiquity, autonomy, and communication.

A domain in academia that investigates automated digitisation of human actions into computer systems is that of activity recognition. A sub-domain

of the long-studied field of computer vision, activity recognition researchers typically use image and video data to create machine algorithms that attempt to automatically classify a given activity (Vrigkas, Nikou, & Kakadiaris, 2015). However, a system that comprises of vast amounts of cameras may be seen as too visible and intrusive, especially if a requirement of the system is to be pervasive.

Therefore, the purpose of this research is to investigate a suitable interface for the IoT by designing and implementing a single-modality ground-based activity recognition sensor that considers the findings of relevant literature in its iterative development. As acknowledged in the following discussion, research has been conducted in ground-based activity recognition already. However, in a similar vein to IoT discussion, the main objectives are technical with less consideration to domain application, and on lack of evidence identified in the literature through this study, research into an IoT interface using ground-based activity recognition has not previously been undertaken prior to this research. This leaves further scope for knowledge discovery in this regard beyond the extent of the research presented in this thesis. Evaluation of the sensory system will inform the suitability for such an implementation within the IoT.

This thesis consists of six chapters. Chapter one provides the motivation and background of the research. Chapter two examines literature that covers relevant topics on the IoT, from early iterations to its current form in smart objects and smart environments. Discussion on human inclusion in digital systems and perceived barriers to this are also reviewed, before analysing literature discussing a particular subset of interfaces and the associated machine learning techniques used in activity recognition. This literature review helps inform the discussion of chapter three, which outlines the research objective and highlights the research methodology used in this study. Chapter four discusses the construction of an artefact in investigation of the research questions, from the physical components to the software components. Chapter five contains the results and discussion of evaluating the artefact. Finally, chapter six presents the conclusion,

highlights the contribution to knowledge in this domain, acknowledges the limitations of the research, and provides further areas of investigation going forward.

## 2 REVIEW OF LITERATURE

Given that the aim of this research is to understand how activity recognition could be used in a smart environment of IoT devices, this literature review will touch upon relevant areas that include sensors and technology in pursuit of ubiquitous computing in the IoT, the human factors involved, introducing humans in systems through unobtrusive activity recognition (AR), existing implementations of AR, and machine learning efforts in the field of AR.

### 2.1 IOT DEFINITIONS

An all-encompassing definition of the IoT is given by Gubbi, Buyya, Marusic & Palaniswami (2013) who extensively describe their definition of the Internet of Things as it being the:

Interconnection of sensing and actuating devices providing the ability to share information across platforms through a unified framework, developing a common operating picture for enabling innovative applications. This is achieved by seamless ubiquitous sensing, data analytics and information representation with Cloud computing as the unifying framework. (p. 1647)

This rather comprehensive definition of a ubiquitous utopian computing architecture is more akin to the modern interpretations of the IoT, which is reflected on the redefining of the ITU's IoT designation in 2012 as "A global infrastructure for the information society enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies", further adding that "From a broader perspective, the IoT can be perceived as a vision with technological and societal implications" ("Y.2060 : Overview of the Internet of things," 2012., p. 1).

This repositioning in definition by authors is to be expected and is considered by Atzori, Iera, & Morabito to be attributable to two foremost biases concerning defining the IoT. The first bias is "The historical period, with all the relevant evolutionary history of ICT technologies adopted by

IoT, in which the definition is conceived” (Atzori, Iera, & Morabito, 2017, p. 135). As time passes and technology evolves, the concept and therefore the definition of the IoT will change. Furthermore, societal pressures and consumer trends will also affect what the IoT may become in the future.

The second bias in defining the IoT is dependent on the paradigm or domain of expertise in which the author or stakeholders’ interests lie (Atzori et al., 2017). Further to this, in earlier work they designated three paradigms concerning IoT discussion as internet and network-oriented (middleware), things or devices oriented (sensors) and semantic-oriented (knowledge) (Atzori, Iera, & Morabito, 2010). Thus, bias in defining the IoT can be seen in the first ITU definition above, which is predominantly concerned with the connectivity issues within the IoT (middleware) and is to be expected given the focus of the ITU in telecommunications.

Considering these two points, this research will adopt definitions provided by more recent literature, while focusing on the development of sensors and knowledge in application of activity recognition while simultaneously recognising the effects of middleware on both of these.

The delineation of the IoT into three domains expressed by Atzori et al. is also identified by Gubbi et al. who say that “this type of delineation is required due to the interdisciplinary nature of the subject,” yet warned against as unhelpful in the realisation and direction of the IoT, and that “the usefulness of IoT can be unleashed only in an application domain where the three paradigms intersect (Gubbi et al., 2013, p. 1646).

Thus, in an ever evolving attempt to define the IoT, the definition given by Atzori et al. (2017) in its current iteration is that it is:

a conceptual framework that leverages on the availability of heterogeneous devices and interconnection solutions, as well as augmented physical objects providing a shared information base on global scale, to support the design of applications involving at the same virtual level both people and representations of objects. (p. 137)

Another way in which to define the IoT is to discuss comparative features in literature pertaining to the IoT. Atzori et. al discuss six paramount concepts present in the majority of literature in some form; “a global network infrastructure or network connectivity”; “Everyday objects, not only ICT devices, are the main players of the IoT”, further adding the importance of “virtual representations” of these objects “within a digital overlay information system that is built over the physical world”; “Autonomy and autonomicity” allowed by smart objects and smart systems; “the design of effective (better if “intelligent”) interfaces both between humans and things and between things”; “Heterogeneity of the technologies” and the enabling of collaboration between them; “Services need to be associated to the objects” and provide value (Atzori et al., 2017, pp. 135-136).

As such, the IoT must be recognized and considered in its entirety; a pervasive multi-part system and framework of sensors, things, devices and environments, with connectivity technology allowing communication between them all, and the services and frameworks that make use of the data and information provided by them. Each of these performing a necessary function in-order to offer a utopian and ubiquitous computing reality that will undoubtedly evolve over time. As this will be the terminology used within the research, a brief investigation of one of the most prevalent aspects of this definition will be discussed next.

### *2.1.1 UBIQUITOUS COMPUTING*

The IoT therefore shares concepts in existing technological frameworks, and thus adds to the difficulty in understanding the IoT paradigm, a view expressed by Atzori et al. (2017) who recently argued that:

if one brings into the Internet of Things many concepts derived from different architectures and technologies, such as ubiquitous/pervasive computing, Internet Protocol (IP), Machine-to-Machine and embedded devices, Internet of People, then eventually this makes IoT synonymous with everything and, therefore, denies to IoT the specific connotation it deserves. (p. 132)

Importantly though, Atzori et al. in their critique are not rebuking the underlying existing technology used, but rather the use of the term IoT as a rebranding or substitute of these existing technological frameworks that are deficient in features compared to the IoT. Table 1 highlights these discrepancies as considered from their perspective.

*Table 1: Current technological frameworks: existing and missing IoT features (Atzori et al., 2017, p. 136)*

<b>Technology</b>	<b>IoT features</b>	<b>Missing features</b>
<b>RFID platforms</b>	Pervasiveness; often integrated with sensors/actuators	Effective object virtualization; autonomy and autonomicity; interaction between objects
<b>Pervasive computing platforms</b>	Pervasiveness, autonomy and autonomicity; heterogeneity of technologies; association of services with objects	Global network infrastructure; interfaces for thing to thing interactions
<b>Cyber-physical systems</b>	Pervasiveness; autonomy and autonomicity; interfaces between humans and things as well as between things; heterogeneity of technologies; association of services with objects	Global network infrastructure
<b>Sensor networks</b>	Autonomy and autonomicity; association between services and physical resources	Global network infrastructure; pervasiveness; heterogeneity of the technologies;
<b>M2M systems</b>	Connectivity and global network infrastructure; Interfaces between humans and things as well as between things; heterogeneity of the technologies	Pervasiveness; autonomy and autonomicity

One of the most obvious features that converge amongst the different paradigms is that of pervasiveness. In this terminology, no longer are the computerized technologies of tomorrow be limited to the banal screens, beige

boxes, or even pocket-sized computing fashion statements familiar to computing. In 1966 Karl Steinbuch, a German Computer Science pioneer, believed that computers would be inter-woven with almost every industrial product, and nearly three decades ago, Mark Weiser in his paper “The Computer for the 21<sup>st</sup> Century” anticipated the ubiquity of modern technology when he opened his discourse with the notion that “The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it” (Weiser, 1991, p. 94).

Unfortunately for both Steinbuch and later Weiser and his team at Xerox PARC, “this was a vision too far ahead of its time — the hardware technology needed to achieve it simply did not exist” (Satyanarayanan, 2001, p. 10). Now, well into the 21<sup>st</sup> century at the epoch of technological ubiquity, with advances in both the computing technology and supporting systems, it could be argued that technology is indeed in the fabric of everyday life as indicated by the statistics of IoT and smartphone devices mentioned at the outset.

However computerised technology has not largely “disappeared” or become “indistinguishable” from the mundane objects people rely heavily on or immediately associate with computerised technology, such as smartphones, tablets or personal computers. Or, maybe it is, and has indeed become so indistinguishable that most are naïve to its existence or dependence on it. Weiser (1991) acknowledged this when he wrote that:

Such a disappearance is a fundamental consequence not of technology but of human psychology. Whenever people learn something sufficiently well, they cease to be aware of it. When you look at a street sign, for example, you absorb its formation without consciously performing the act of reading. (p. 94)

Therefore, truly ubiquitous computing means more than being physically invisible or unseen, but more importantly can be interacted with unconsciously (Barbosa, 2015). Satyanarayanan provides further compromise to complete, literal invisibility by writing that a “reasonable

approximation to this ideal is *minimal user distraction*. If a pervasive computing environment continuously meets user expectations and rarely presents him with surprises, it allows him to interact almost at a subconscious level” (Satyanarayanan, 2001, p. 11). This concept of invisibility and minimal user distraction is very pertinent to the IoT and discussion within this research.

## 2.2 SENSORY SYSTEMS

The precursors to the IoT largely focused on data produced from sensors. Even today, it is assumed that the majority of objects within the IoT will be simple data acquiring modules rather than whole appliances or machines. Furthermore, many of the implementations of sensory based AR, rather than vision based AR, rely on two sensory instruments. Both have relevance in the history and realisation of ubiquitous computing and will be discussed in the next section.

### 2.2.1 EARLY IOT SENSORY SYSTEMS

Although Weiser is largely thought of as the principal champion of ubiquitous computing, Kevin Ashton, co-founder of the Auto-ID Centre at the Massachusetts Institute of Technology, is credited as having coined the term “Internet of Things” in a presentation to Proctor & Gamble (P&G) in 1999 (Barbosa, 2015; Mattern & Floerkemeier, 2010). In this presentation, Ashton was discussing the implementation of Radio-Frequency Identification (RFID) for use within the supply chain at P&G. RFID tags or transponders, as they are also known as, are the physical part of a system that is used for identification purposes similar to typical printed barcodes, but “support a larger set of unique IDs than bar codes and can incorporate additional data such as manufacturer, product type, and even measure environmental factors such as temperature” and without the need for line-of-sight like traditional bar codes (Want, 2006, p. 25). Atzori et.al acknowledged the importance of RFID in IoT development saying that “Radio-Frequency IDentification (RFID) technology played the role of founding technology for the Internet of Things” (Atzori et al., 2017, p. 126).

Developments were made in what Atzori et al. refer to as the “The first generation of the IoT: the tagged things” where the IoT was primarily considered to be networked RFID (Atzori et al., 2017, p. 126). In this period, work was conducted by various organisations and researches to integrate RFID as the physical component of a larger Wireless Sensor Networks (WSN) in order to achieve ubiquitous computing and overcome the limitations of RFID only platforms, along with Machine-to-Machine (M2M) communication to allow automated processing (Liu, Bolic, Nayak, & Stojmenovic, 2008; Sung, Lopez, & Kim, 2007; Welbourne et al., 2009). Developments are still being made in this area “With the interest of enabling globally interoperable data sharing, a collection of standards and specification have been released by GS1 EPCglobal to support supply chain management” (Tolcha & Kim, 2016, p. 29).

Research was also conducted to develop more capability in the RFID tags themselves by extending the functionality of these simplistic devices to offer sensing capabilities, more memory storage, and powered RFID tags that rely on batteries or other technologies to power these sensors embedded in RFID tags and, in some instances, enhance communication abilities (Paing et al., 2007; Sample, Yeager, Smith, Powledge, & Mamishev, 2006). Through this research, integrating the functionality of RFID tags into a more encompassing system extends the domains in which RFID may be useful within an IoT system, especially as a physical or hardware component. Embedding these RFID tags with multiple layers of information offers far more flexibility and complexity than simple bar codes.

It is interesting to note that despite all these developments and continuing exploration of, RFID, WSN, and their related technologies, the applicable domains are still largely limited to the those same ones mentioned earlier, namely manufacturing, processing, and logistical applications (Da Xu et al., 2014). Having a technology which is largely “within isolated, vertically integrated, systems, used only for identification and/or tracking of objects”(Miorandi, Sicari, De Pellegrini, & Chlamtac, 2012, p. 1500) doesn't bode well for the realization of a ubiquitous computing paradigm.

Interestingly, Miorandi et.al also offer an explanation for the limited domains in the application of RFID by arguing that they need to be “part of a larger system, where identification of an object is only a step of the workflow to be executed to provide a final service” (Miorandi et al., 2012, p. 1500). A comparatively similar explanation is offered by Kortuem, Kawsar, Fitton, & Sundramoorthy (2010) discussing the RFID system as a whole:

RFID system architecture is marked by a sharp dichotomy of simple RFID tags and an extensive infrastructure of networked RFID readers. This approach optimally supports tracking physical objects within well-defined confines (such as warehouses) but limits the sensing capabilities and deployment flexibility that more challenging application scenarios require. (p. 44)

This aligns with some of the limitations in RFID platforms and architecture that was alluded to by Atzori et al. in their list of misappropriated references to the IoT, namely “Effective object virtualization; autonomy and autonomicity; interaction between objects” (Atzori et al., 2017, p. 125). Therefore, the physical taxonomy of the IoT is not built on RFID alone, unless limited domains are to be the expected outcome, voiding the argument of ubiquitous computing.

### *2.2.2 SMARTPHONES – THE UBIQUITOUS FUTURE?*

Smartphones are everywhere in a modern society (“Number of mobile phone users,” n.d.). However, studies have shown that smartphones demand more attention and distract users from such simple tasks as walking, causing injuries and even deaths. (Nasar & Troyer, 2013; Hatfield & Murphy, 2007). Therefore, if the IoT is considered to be the epitome or at the very least shares principles of ubiquitous computing, then recounting views discussed earlier such as Gubbi et al. (2013) who realise the importance ubiquity plays in the IoT, and Satyanarayanan’s (2001) reasoning on minimal user distraction, devices such as the smartphone or personal computers should not be acknowledged as part of the IoT paradigm. To describe these technologies as the ubiquitous computing future that Weiser envisioned would then seem to be erroneous.

Many industry stakeholders agree with this viewpoint, such as MacGillivray, IDC program vice president for mobile services, IoT and infrastructure; “The key words in our definition is that it’s communicated without human interaction. So, at the simplest level, we are not including smartphones, tablets, PCs, etc.”, and Dennis Ward, Internet of Things analyst at ACG agrees with this sentiment, saying that IoT devices are “not designed for direct human interaction, connectivity or control” (Duffy, 2014). However, others disagree with the exclusion of smartphones from the IoT ecosystem; “Using your smartphone’s range of sensors (accelerometer, gyro, video, proximity, compass, GPS, etc.) and connectivity options (cell, Wi-Fi, Bluetooth, NFC, etc.) you have a well- equipped Internet of Things device in your pocket that can automatically monitor your movements, location, and workouts throughout the day” (Weber, 2016, p. 44). While a smartphone’s primary function can be arguably seen as a human to human telecommunication device, its various sensors and network communication technology changes functionality so that it can produce data with or without conscious human interaction. This delineation between the devices functionality and the device itself is fundamental in consideration of the smartphone as part of ubiquitous computing paradigm and as physical component of the IoT.

Acknowledging this separation between designed intentions and actual capabilities, consider the description of ubiquitous computing in relation to the IoT given by Gubbi et al. (2013):

There are three IoT components which enables seamless ubicomp:  
(a) Hardware—made up of sensors, actuators and embedded communication hardware (b) Middleware—on demand storage and computing tools for data analytics and (c) Presentation—novel easy to understand visualization and interpretation tools which can be widely accessed on different platforms and which can be designed for different applications. (p. 1647)

Therefore, according to Gubbi et al., a smartphone has to be an IoT device with its sensors and communication hardware, with the ability to interact

and facilitate physical interaction through its interface. Furthermore, it could be argued that a smartphone is all three components of this paradigm in one, not only being able to provide the sensory components, but also storage and computing as well as provide the necessary tools to aid with presentation and visualization. However, it is paramount to acknowledge smartphones as part of a system, which becomes evident in environments which leverage aspects of the IoT.

### *2.2.3 SMART ENVIRONMENTS*

A Smart Environment (SE) is simply a “physical environment enriched with sensing, actuation, communication and computation capabilities aiming at acquiring and exploiting knowledge about the environment so as to adapt itself to its inhabitants’ preferences and requirements” (Franco Cicirelli, Fortino, Guerrieri, Spezzano, & Vinci, 2017, p. 274). SE’s do so by using the sensors such as RFID and Smartphones as well as others to enable this adaptability not possible in conventional static environments. They are arguably an extension of the paradigm of Weiser’s ubiquitous computing vision by being “richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network” (M. Weiser, Gold, & Brown, 1999).

This concept of SE shares aspects with Ambient Intelligent (AmI) systems with “sensing/computing capabilities embedded in the environment”, but differing in that “AmI applications have been mainly developed for “closed” environments (e.g., a room, a building), whereby a number of specific functions (known at design time) can be accommodated and supported” (Miorandi et al., 2012, p. 1500). However, it is the ability to communicate data away from the closed environment and send “information to the cloud so that it can access remotely” services, combined with the inoperability of the larger system that warrants the discussion of SE in relation to IoT (Raun, 2016, p. 4).

Smart Homes are a very popular topic for researchers in the field of SE’s with many published papers appearing since 2010 (Alaa, Zaidan, Zaidan,

Talal, & Kiah, 2017). It is apparent that research in this area is primarily “made up of technical and prospective studies that focus on security and control, with a secondary emphasis on activity”, ignoring the fact that a home is “ a place of security and control; activity; relations and continuity; identity and values” (Gram-Hanssen & Darby, 2018, p. 98). Taking a more human-centred approach in discussing Smart Home Technology (SHT), a study found that “Prospective users of SHTs more strongly perceive *potential risks* in the increasing dependence of domestic life on systems of technology provision (77% agree or strongly agree) and electricity networks (63%),” but also found that these perceived risks could be overcome by creating a system that is “easy to use, controllable, and easy to over-ride” adding that they “should guarantee privacy, confidentiality, and secure data storage. SHTs should also be provided by credible companies with resources to provide performance warranties” (Wilson, Hargreaves, & Hauxwell-Baldwin, 2017, p. 44). Indeed, human perception is an important facet of emerging and disruptive technology acceptance, which will be discussed later.

Commenting on Large Smart Environments (LSE), F. Cicirelli et al. (2017) discuss how these can be “longtime running systems where new devices can be dynamically added and removed, and new functionalities can be added, removed or replaced by exploiting the available devices and/or composing the existing services” (p. 738). Therefore, the notion of adaptability and flexibility of a SE is not only a comment on the physical aspects of the environment itself but on the changeability of the system as well, as potential participants interact and transit, their presence and accompanying technology (smartphones, fitness bracelets etc.) changing the environment and system parameters accordingly. A system of fixed sensors, such as RFID, as well as mobile devices, such as smart-phones, need to be accounted for in such a system.

This is a complicated dilemma for LSE to remedy in regard to the ubiquitous invisibility mentioned earlier, “Since motion is an integral part of everyday life, such a technology must support mobility; otherwise, a user

will be acutely aware of the technology by its absence when he moves” (Satyanarayanan, 2001, p. 2). The success of LSE or even smaller SE will rely heavily on the intelligence of an AmI and as such relies heavily on autonomy, a field discussed later. Further to this point of mobility of such objects or Things is the fact that they may need to exist and operate beyond the bounds of an encapsulating SE, particularly in the present time where such SE implementations are limited, a view is supported by Satyanarayanan (2001) who acknowledges that “Smartness may also extend to individual objects, whether located in a smart space or not” (p. 11). This notion of smartness in objects will be discussed next.

### 2.3 SMART OBJECTS

Refrigerators that automatically order milk when low or thermostats that control the heating and cooling appliances in your home are usually the examples given as intrinsically being the Internet of Things (IoT). However as already explored, this layman notion of the IoT is obviously flawed; given the estimates of 20.4 billion IoT devices by 2020, there simply isn’t such demand for that many refrigerators or thermostats. Apart from the definitions already provided, a description that includes physical aspects to the definition of the IoT gives more understanding to the heterogeneity of these ‘Things’ and the need for interoperability, as described by Atzori, Iera, & Morabito, (2010):

The basic idea of this concept is the pervasive presence around us of a variety of things or objects – such as Radio-Frequency Identification (RFID) tags, sensors, actuators, smartphones, etc. – which, through unique addressing schemes, are able to interact with each other and cooperate with their neighbours to reach common goals. (p. 2787)

Indeed, while stocked refrigerators or temperate homes may be desirable and perceived by some as the epitome of the IoT, the physical taxonomy of the IoT seems just as ambiguous and undefined as the use of the word ‘Things’, making it hard to describe what a physical IoT device should be. In fact, some use the term Internet of Everything (IoE) coined by Information

and Communications Technology (ICT) company Cisco as a synonym for the IoT (Schatten, Ševa, & Tomičić, 2016) adding even more confusion to this debate. In spite of this, some authors hold views that the direction of the IoT is driven primarily by these Things or Smart Objects (SO), describing a model for IoT architecture “as a loosely coupled, decentralized system of *smart objects* – that is, autonomous physical/digital objects augmented with sensing, processing, and network capabilities” (Kortuem et al., 2010, p. 44).

However, with Aztori et al. (2017) insisting that “Everyday objects, not only ICT devices, are the main players of the IoT” (p. 135), than the IoT must be ubiquitous in not only the proliferation of simple standalone sensory devices, but through the inconspicuousness of technology laden appliances and objects that are commonplace to everyone but aren’t currently part of this IoT paradigm. However, this introduces an issue related to the reliable integration of these everyday objects or things into the IoT framework.

### 2.3.1 VIRTUAL REPRESENTATION

The abstraction of necessary attributes and functionality of the physical object into its virtual representation is a crucial procedure if an everyday object intends to move from an analogue to digital presence and participate practically in the IoT. Van Kranenburg (2008) describes this necessity for digital representation; “physical and virtual ‘things’ have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network” (p. 10). Atzori et al. (2017) further extend the importance of a virtual intermediary:

sensors and actuators shall be embedded into physical objects to enable them to operate through their virtual representations within a digital overlay information system that is built over the physical world. (p. 135)

The benefit of this layer is that it provides an interface between the physical world of SO with possible actions or services, both local and remotely through a global network. This enhancement occurs by making

“heterogeneous objects interoperable through the use of semantic descriptions; enable them to acquire, analyse and interpret information about their context in order to take relevant decision and act upon the virtual objects” (Nitti, Pilloni, Colistra, & Atzori, 2016, p. 1228).

There is much literature on the abstraction of key elements of a physical object into its virtual representation, with the majority of focus on the integration between the physical object and the web, as well as attempts to standardize related architectures for the realisation of interoperability of heterogeneous SO's, with organisations such as European Telecommunications Standards Institute (ETSI) M2M Common Service Layer and projects such as Collaborative Open Market to Place Objects at Your Service (COMPOSE) and iCORE, among others (Bergesio, Bernardos, & Casar, 2017; Gračanin, Matković, & Wheeler, 2015; Han & Crespi, 2017; Stecca, Moiso, Fornasa, Baglietto, & Maresca, 2015).

Fortino, Guerrieri, Russo, & Savaglio (2015) suggest a “*High-Level Smart Object Metamodel*” in construction of a SO, and identify six key characteristics that should be implemented in a virtual layer for a successful SO shown in Table 2.

**Table 2:** High-Level Smart Object Metamodel (Fortino, Guerrieri, Russo, & Savaglio, 2015, p. 1298)

Characteristic	Function
<b>Status</b>	represents the SO status
<b>FingerPrint</b>	contains the information of the SO such as its identifier, its creator, its type (e.g. smart pen, smart office, etc.) and its associated quality of service parameters (e.g. trustness, reliability, availability, etc.)
<b>PhysicalProperty</b>	describes the physical properties of the original object without any hardware augmentation and embedded smartness
<b>Service</b>	describes a provided digital SO Service, by specifying its identifier, description, input/output parameters
<b>Device</b>	defines the hardware/software characteristics of a device (e.g. sensors, actuators, etc.) that allows to augment the physical object and makes it smart
<b>Location</b>	expresses the SO geophysical position

## 2.4 HUMAN-CENTRED IOT

While important use of models to enable virtualisation of physical objects enable easier instantiation of these objects with “smart” technology, it doesn’t particularly address the ‘meatspace’ discussed by Atzori et al; “The interesting definition of IoT as an “intersection of people (meatspace), systems (cyberspace) and physical world (atomspace)” (Atzori et al., 2017, p. 135). This intersection that Atzori et al. describes is a complicated amalgamation of spaces, as each space must interact with each other. This means the virtual representation of a device must coherently represent the objects own functionality for both human and machine agents, important in the concepts of autonomy and autonomicity (discussed later), as well as translate between the two without inhibiting the communication or use of the object between the other two spaces, or the ‘meatspace’ and ‘atomspace’, as traditional analogue activity between object and human occurs.

Creating this harmonious intermediary then requires thought and consideration to include humans within the system, and while not all

definitions of the IoT include humans as part of the system, it seems only logical that a system aimed at creating a utopian ubiquitous computing environment at the very least includes humans as a factor of that environment, and must therefore account for the addition of biological agents within this digital system along with all this entails. Therefore, research regarding human inclusion into the IoT from adoption, creation and design, and the removal of human elements will be discussed.

#### *2.4.1 ADOPTION*

Salim & Haque, (2015) in their discussion of ubiquitous urban computing noted that there has to be “a strong compelling reason or clear benefit” for users to engage in this technological future” (p. 35), which they reason is possible as long as “the needs are greater than the dilemmas” (p. 43).

One such dilemma to consumers is cost to benefit ratio. Even at the dawn of technological ubiquity, Want (2006) discusses that one of the major inhibitors of mass adoption of early RFID technology was cost, because “Although RFID tags are now potentially available at prices as low as 13 cents each, this is still much more expensive than printed labels” (p. 32). Therefore, the perceived usefulness and benefit of RFID tags had to overcome the cost barrier of traditional barcodes, and while prices of technology tend to reduce over time, Salim & Haque’s argument remains applicable in that the perceived usefulness of SO’s and sensors in conjunction with the IoT must outweigh the cost in the mind of the consumer (Meyers, Williams, & Matthews, 2010) .

Perceived usefulness correlates with many aspects of Human and Computer Interaction (HCI) and not just SO’s within the IoT but should be more prevalent in SO discussions simply because of the necessary ubiquity and pervasiveness given the IoT paradigm. This importance of user perception is supported by Marangunic & Granic’s (2015) review of literature on the Technology Acceptance Model (TAM), a model proposed by Fred Davis in 1986 for investigating user behaviour in relation to new technology. They wrote that the “The TAM presumes a mediating role of two variables called perceived ease of use and perceived usefulness in a complex relationship

between system characteristics (external variables) and potential system usage” (p. 15). Aside from perceived usefulness, TAM also introduces the perceived ease of use as another aspect that needs to be taken into consideration in creating SO’s for the masses. If an object or system is intrinsically difficult to use or benefit from, this negatively impacts acceptance, and thus will reduce the uptake of SO by those with this view. Furthermore, Parasuraman (2000) introduced the idea of Technology Readiness (TR), describing TR as “people’s propensity to embrace and use new technologies for accomplishing goals in home life and at work” (p. 308). In this day and age of ubiquitous computing, where computing devices have become part of the compulsory school stationary list, and smart phone usage has reached record numbers, people’s readiness to use technology is already apparent. However, Parasuraman highlights two major negative emotions affecting TR. These are discomfort, stemming from a user’s lack of control and confidence, discussed in the previous section, and insecurity in the technologies ability to be reliable and accurate (Parasuraman, 2000).

In relation to the latter point, there are real issues regarding Smart Objects and the IoT, such as privacy and security (Hu, 2016; Miorandi et al., 2012; Williams, 2016). A study conducted by HP showed that on average, 25 vulnerabilities existed on each of the most common devices found within the IoT, with 80% of devices not requiring sufficiently complex passwords and 70% not using encryption for communication, with a further 60% of these devices containing vulnerabilities within its own firmware or user interface (“HP News”, n.d.).

It is no surprise then that hackers have already reportedly used SO’s and IoT devices to attack large corporations “thanks to everyday gadgets, like baby monitors and webcams. The hackers gained access to these innocent-looking objects and turned them into an army to take down the internet” (Batchelor, 2017, p. 6). A recent report also showed that this concern is held in the business and industry environment as well, where “The findings of the underlying study revealed that even though over 69 percent of organizations have adopted, or plan to adopt, IoT solutions within the next

year, 40 percent of companies have serious concerns around cybersecurity” (Cradlepoint, 2017). It is out of the scope of this thesis to discuss the security and privacy factors and possible solutions in-depth, sufficed to say these issues of SO’s, sensors, and data privacy and security within the IoT are known, and impact negatively on public and industry perceptions, thus affecting TR by validating these insecurities.

Further to the discussion of SO acceptance in application of a Smart Home, Balta-Ozkan, Davidson, Bicket, & Whitmarsh, (2013) summarises literature that covers the points raised above, namely “fit to current lifestyles; technological complexity; interoperability and standards; reliability; privacy and security”, but found that human perceptions were also negatively affected by “loss of control and apathy; reliability; viewing smart home technology as divisive, exclusive or irrelevant” further adding to the latter point that “deeper, moral concerns about human nature, inequality, and trust were a stronger feature of public discussions” (p. 371). These succinctly describe the issues that will need to be addressed for people to change their perception and willingly adopt these emerging technologies.

#### *2.4.2 AUTONOMY*

Autonomy is not a recent goal of computing, and frameworks for implementing computing autonomy have been introduced by major industry stakeholders, including IBM who in the early 2000’s designed such a framework (Kephart & Chess, 2003).

As Atzori et al. (2017) discuss in one of the features apparent in literature regarding the IoT:

Autonomy and autonomicity are two recurrent features which are claimed to characterize the objects that populate the IoT. This has not to surprise, as it clearly emerges from the literature that system complexity can be controlled through the achievement of self-governance (autonomy) and self-management (autonomicity). (p. 135)

Adding automation to this statement, automation, autonomy, and autonomy are seen as being the obligatory and unavoidable component of a ubiquitous computing future. Regardless of which is being discussed, at the forefront of these terminologies is the removal of human intervention in various parts of a system. For example, many studies focus on the application of automation within different domains, removing humans from menial tasks. (Bello & Zeadally, 2017; Delsing, Eliasson, Van Deventer, Derhamy, & Varga, 2017; Giri, Dutta, & Neogy, 2017; C. Wang, Bi, & Xu, 2014). Other investigate autonomy and autonomy in the IoT, further removing human intervention by negating the need to manually manage systems (Iannacci, Sordo, Serra, & Schmid, 2015; Le, Liao, & Yang, 2017; Lee et al., 2016).

Gubbi et al. (2013) comment on the insignificance time has had on this ambition of automation and autonomy, saying “Although the definition of ‘Things’ has changed as technology evolved, the main goal of making a computer sense information without the aid of human intervention remains the same” (Gubbi et al., 2013, p. 1646). To have devices gather information without human interference is relatively simple, however, the goal of making a computer not only sense but *make sense* of information aligns more toward the goal of autonomy. This has become a real possibility with the advent of machine learning techniques, which will need to make inferences about human objectives if the completely autonomous IoT future is to be realised.

Khan et al. (2012), in establishment of a three part IoT workflow, acknowledge that automation and autonomy of some level is required saying that “object information is processed by a smart device/system that then determines an automated action to be invoked”, however also commenting that “The smart device/system provide rich services and includes a mechanism to provide feedback to the administrator about the current system status and the results of actions invoked” (p. 258). Thus, they argue that autonomy shouldn’t necessarily remove humans altogether, but have some feedback mechanism showing vital system information. This is a view

which would seem to support a semi-autonomous rather than a fully autonomous architecture. This is because feedback of the current system status seems unnecessary if the mechanisms for a system are completely autonomous and manual intervention in this system by a human is unavailable. It may ease some of the psychological issues such as loss of control by allowing a human user to know a systems current status in an autonomous environment, but it would be akin to having a normal dashboard of a current car (speedometer, revolutions per minute, fuel gauge) visible in a completely autonomous car where a human user is unable to influence the system; interesting for information and psychological sake but useless and ultimately unnecessary from an operational perspective.

While the importance of information presentation or feedback to humans in a fully autonomous system are debatable, it is argued that autonomy in the IoT is a necessity “due to the large number of devices involved. Specifically, there exists an inefficiency that can be resolved by minimizing user intervention” (Ashraf & Habaebi, 2015, p. 215). This is valid argument and would ultimately create a more beneficial IoT by optimizing efficiency.

Importance of a feedback mechanism to a governing machine system in a Machine to Machine operation, such as a smart-device to a cloud autonomy service, are less debatable as this communication is required for effective IoT operation between heterogeneous devices. Some however debate the efficiency of cloud based versus edge based autonomous services, due to latency and bandwidth issues (Hernández & Reiff-Marganiec, 2016).

While these are only some of the issues regarding automation and autonomy in the IoT, these alone demonstrate the complexity of normalizing machine control within the general human population, a view expressed by Verame, Costanza, & Ramchurn (2016), who describe “the design of interaction mechanisms that enable users to understand the operation of autonomous systems and flexibly delegate or regain control is currently an open challenge for HCI” (p. 1). Thus, creating effective automation, autonomy, and autonomicity may remove some of the human inhibitions related to

technological complexity, but exacerbate those notions related to loss of control, apathy and trust, affecting TR as discussed earlier.

### *2.4.3 CREATION AND DESIGN*

There have been numerous studies highlighting the need and recognition of the human importance within the IoT (Fauquex, Goyal, Evequoz, & Bocchi, 2015; Nazari Shirehjini & Semsar, 2017; Nuamah & Seong, 2017). Atzori et al. (2017) noted the importance saying “Particular attention has to be paid to the design of effective (better if “intelligent”) interfaces both between humans and things and between things” (p. 135). Design of Smart Objects (SO) or Things should therefore include consultation with all human stakeholders, not just the creators of the technology. It would be more advantageous to identify and create SO’s through the objectives they are designed to achieve by these stakeholders, instead of simply bestowing objects with some technological capabilities or finding ways to implement objectives with existing solutions “driven by technological opportunities rather than the underlying needs” (Salim & Haque, 2015, p. 32). Related to this view, Kranenburg (2008) mentions that “‘things’ are expected to become active participants in business, information and social processes” (p. 10). To become active participants in multiple domains therefore, these SO’s and the encompassing systems will need to be designed by those with expertise in these domains who may require a specific service, or with an objective that could be filled by these objects, not by the technologists who, at this point in time, seem to be the proponents of SO’s and the integration into the IoT. Collaboration is key.

This human-centric approach, with domain experts at the fore, is not new to ubiquitous computing, and is reflected in Rogers (2006) view who writes:

In terms of who should benefit, it is useful to think of how ubicomp technologies can be developed not for the Sal’s of the world, but for particular domains that can be set up and customized by an individual firm or organization, such as for agricultural production, environmental restoration or retailing. (p. 412)

Following this ethos would create a ubiquitous computing environment that was more accessible to those in specific domains, but also for the general public and their everyday activity by following the same principles of user-led design.

Another recent study investigated the need for alternative views of thought in the construction and design of these SO's. By using children in their experiments, Uğur Yavuz, Bonetti, & Cohen (2017) showed that “children as “design partners” can bring novel ideas, expanding the notions of smart objects through the use of storytelling technique” because of “their unbiased and free-spirited imagination towards the creation of ‘the new’, of new future scenarios” (p. 3800). From this, some of the ideas generated were smart food plates that could identify the food being served and change its form to suit the food accordingly, or smart refrigerators that not only keep stock of its contents but are able to share unused or unneeded contents with other remote smart refrigerators, reducing waste. While there are some obvious logistic implications with some of the ideas, it shows that “Design for emerging technologies can benefit from game based fictioning methods in which constrains of real world has less influence on idea generation and thereby can lead to originality” (Uğur Yavuz, Bonetti, & Cohen, 2017, p. 3800).

It may be possible that the design of these SO's may not be realised by industry at all, but by individuals. For example, the “Maker Culture has been considered as a possible new industrial revolution, as more people than expected demonstrated their interest in creating *things* with electronics toolkits” (Mazzei, Fantoni, Montelisciani, & Baldi, 2014, p. 294). However, to date there is no standardised platform or model for individuals in building a SO, especially one that could integrate with the IoT. There is research regarding possible implementations of models which look to ensure individuals are able to design and build their smart object and integrate it within a cloud-based services (Z. Li & Point, 2015). However, this problem isn't isolated to the maker community, as the heterogeneous nature of SO's and sensors in the IoT reflects the issues around standardisation in

commercially available products too. As such, the framework suggested by Fortino et al. earlier could be applicable to bespoke SO and commercial products alike, and by leveraging the benefits of the virtualisation proposed in the model, along with protocol standardisation, the physical technological implementation, whether commercial or bespoke, becomes irrelevant. This would enable greater participation within the IoT.

In conclusion of this section, including humans into the IoT system requires implementation of an unobtrusive, ubiquitous computing paradigm using sensory technology that is indistinguishable from non-smart objects, able to fit current modes of human interaction. It would need to integrate successfully with the wider IoT system, using virtual layers for digital representation that would interact with autonomous services in conjunction with other SO's to provide adaptable SE's. While there are other possible directions to consider in implementation of such an interface, reflecting on the literature discussed thus far serves as justification for the implementation chosen as the focus of this research, and will be discussed next.

## 2.5 SMART FLOORS

One way in which to create a harmonious intermediary is to provide an interface to a technological system or service, such as the IoT or AmI, using a surface that everyone interacts with naturally during their daily activities. As Cheng, Sundholm, Zhou, Hirsch, & Lukowicz (2016) discuss, the “vast majority of human activities are associated with certain types of surface contact (walking, running, etc. on the floor; sitting on a chair/sofa; eating, writing, etc. at a table; exercising on a fitness mat, and many others)” (p. 97). Therefore, it seems logical that a floor's role of being a potential interface to the SE and IoT is researched thoroughly. Before the turn of the century in 1997, the idea of incorporating technology into floor systems using piezoelectric wires (Paradiso, Abler, Hsiao, & Reynolds, 1997) and pressure sensitive tiles (Addlesee, Jones, Livesey, & Samaria, 1997) were proposed. Both were fundamentally limited to the current technologies

capabilities, and though some of the principles remain relevant, investigation of more modern approaches is needed.

More recent research has been conducted regarding a Smart Floor (SF) in applications ranging from entertainment (Chang, Ham, Kim, Suh, & Kim, 2010), to elderly healthcare and monitoring (Muheidat & Tyrer, 2016), mental health (Tanaka, Ryu, Hayashida, Moshnyaga, & Hashimoto, 2015), identification and tracking (Al-Naimi & Wong, 2017), indoor navigation (Gonçalves et al., 2013), as an enabler of SE's (Gonçalves, Carvalho, Pinho, & Roselli, 2014), as well as uses as a non-human interface for robots (Kang et al., 2011) and livestock (Vaughan, Green, Salter, Grieve, & Ozanyan, 2017). Others have looked at the intricacies of human anatomy, and in particular the feet, in regard to HCI (Velloso, Schmidt, Alexander, Gellersen, & Bulling, 2015).

As one paper acknowledges, a lot of this work is aimed at “identifying a user based on gait pattern recognition supported by piezoelectric sensing devices and performing personal identification through estimating body-weight and footsteps” (Chang et al., 2010, p.290). Their implementation involved LCD panels on the floor “and LED light and multiple pressure sensors within a specially designed and fabricated steel-frame module” (Chang et al., 2010, p. 291). However, such fabricated solutions requiring bespoke structural flooring elements would not seem to be an eloquent or viable option for a ubiquitous SF solution, although it must be acknowledged that their application was aimed at engaging entertainment and therefore necessitated a display, differing from being simply a surface interface. Other implementations involve invasive and expensive multi-camera systems in conjunction with simple floor sensors to enable location services (Yu, Wu, Lu, & Fu, 2006). A similar concept using glass panels and optical interface such as the GravitySpace tracking system (Bränzel et al., 2013) have the same inhibiting features of being expensive, unsuitable as a floor used in everyday activities, and the use of cameras being viewed as invasive by users.

There is also an implementation using multiple RFID tags being placed underneath ceramic floor tiles, with RFID scanners placed in peoples shoes (Gonçalves et al., 2014). While cheaper than some implementations, requiring people to alter their wardrobe or wear special equipment to allow them to participate in the SE does not seem an efficient SF solution considering the paradigms of the IoT, AmI, and the accompanying ubiquity and invisibility, as the same functionality could be used by tracking smartphone movement. In fact, it exacerbates some of the concerns raised earlier of a perception of inequality by those unable to benefit from SE due to the lack of necessary items discussed earlier.

Al-Naimi & Wong, (2017) summarise the current paradigms of Smart Floors as fitting into one of three categories:

- (I) Tagged tracking including Radio frequency, Ultrasonic, Infrared based approaches.
- (II) Non-tagged tracking including smart floor, machine vision, and wireless distributed Pyroelectric Infrared sensor (PIR) approaches.
- (III) Multimodal tracking including machine vision with laser scanners, smart floor with machine vision, and smart floor with Radio Frequency Identification (RFID) approaches. (p. 34)

Considering that an everyday smart floor concept may require as little deviation from the current, non-smart flooring systems as possible, for inclusivity, cost, simplicity, and ubiquity sake, this would deem non-tagged tracking architectures as being more appropriate as a floor based interface between humans and the IoT, with multimodal approaches offering precision but less ideal as they “suffer from system hardware complexity and unrealistic overall cost”, and that “user preference and inhabitant's privacy are the main weaknesses in tagged based approaches and machine vision respectively” (Al-Naimi & Wong, 2017, p. 34). In light of this, emphasis will be placed on non-tagged architectures.

### *2.5.1 CARPET INTEGRATED ARCHITECTURES*

Muheidat & Tyrer (2016) explore the possibility of IoT integration using

a signal-scavenging technique wherein a sensor made from a conductive material picks up stray 60 Hz noise to detect presence of the person. It has sensors installed under the carpet, and the electronics can send sensor activation data, which is modified to produce notifications to cell phones or email through the Internet. (p. 5356)

Rather than send the raw data to a remote location for processing apparent in the solutions mentioned thus far, they were able to incorporate computation into the carpet itself and introducing this taxonomy of edge computing into the SF implementation. This would enable easier installation in a variety of SE's without the need for connection or communication to a central node for data processing. This is an important aspect of their research as the primary objective was to enable fall detection with real-time alert mechanisms.

Communication latency for real-time and rapid applications using cloud-based computation and servicing is considered by some as a problem in the IoT (Song, Yau, Yu, Zhang, & Xue, 2017), and some proponents of edge computing suggest “moving service provisioning back to the vicinity of IoT devices becomes a potential way to address the challenges of cloud computing and promotes the emergence and development of edge computing” (Ren, Guo, Xu, & Zhang, 2017, p. 96). This allows communication bandwidth to be used in more optimized servicing requests and transmission, rather than inhibited by mountainous streams of data, a factor that will be discussed in section 2.6. It is important to note that this, and other architectures reviewed in this section, require the materials to flex within the sensory surface, whether integrated into the carpet or as a separate layer beneath. Such implementations would not be suitable for hard floor surfaces such as tile or wood, where other architectures that do not require any flex in the surface perform better.

Another architecture, more in line with the paradigms explored above of ubiquity and invisibility, is presented by Savio & Ludwig (2007), where sensors are literally interwoven into the carpet, consisting of electronic sensing wire connected to a micro controller. Their study found “The algorithms were able to track the trajectory of a subject with an accuracy of 98 %” (p. 6). While very accurate, there are important limitations to note. Firstly, working with the technology of the day, they were only investigating the relatively simple task of tracking user trajectory in an area of 200cm by 240cm. Furthermore, their solution had limited sensory resolution of 180 nodes in this space, with each node operating in a binary modality (on or off) over a time series. Thus, more complicated recognition tasks or multi-agent deployments would not be possible under this implementation. They also experienced failure of modules, short-circuits, and connection issues that would be hard to remedy in a textile version of the carpet, possibly requiring expensive and time-consuming solutions, especially in large instalments.

A similar concept of embedded technologies conducted by Ceballos, Nurgiyatna, Scully, & Ozanyan (2011) uses optical fibres on top of a carpet underlay. Light is shone through these matrices of fibres from one end, where the resulting light output, influenced by any bend in the surrounding material, is collected and analysed at the opposite end of the fibre. Combining and analysing the light delta readings enables reconstruction of the pressures placed on the surface. Borrowing from the medical field of Tomography, they use Guided-Path Tomography “an indirect imaging method, which allows to reconstruct images only from measurements at the carpet’s periphery” (Ceballos et al., 2011, p. 1), instead of having sensors intermittently placed throughout the carpet, as is apparent in many implementations. Apart from the specialised carpet, this solution would require no major alterations to existing floor systems, meaning easier integration into normal environments and closer to a ubiquitous IoT interface.

Another study follows similar principles of peripheral detection above, but uses voltage to calculate the pressure placed on the surface. Cheng,

Sundholm, Zhou, Kreil, & Lukowicz, (2014) implement their solution via use of:

Electrostatic discharge (ESD) protection foam as the sensing material, where conductive carbon powder is mixed into the foam. The density of carbon particles grows when the foam is pressed and the resistance decreases. By attaching parallel conductive stripes on each side of the foam, the crossing points of stripes become resistive pressure sensors. (p. 149)

Combined into a matrix enables additional information other than pressure, such as location, to be recreated and a pressure map of the entire surface possible. However, a major drawback of their implementation was the use of the electrostatic discharge (ESD) foam, which has thickness substitutional enough that it would impede or at less influence the typical use of the surface. They recognise this in the research by investigating the need to build “a carpet with lower height (all 3 layers overall < 0.5mm), higher spatial resolution (1 cm<sup>2</sup>) and high precision (24bit ADC with special designed power supply)” (Cheng et al., 2014, p. 152).

They later released a paper in 2016 accounting for these factors and investigated the use of a general-purpose pressure sensing textile usable on multiple surfaces. Their implementation is simplistic yet effective, using the same principles before, but incorporating a commercial manufacturer to create bespoke textiles for all elements of the surface:

The top and bottom layers are made of the same fabric, composed of evenly spaced parallel metallic stripes, separated by non-conductive polyethylene terephthalate. This fabric was designed by us in collaboration with SEFAR AG [62] and woven by SEFAR AG on normal textile machinery. The middle layer is a pressure sensitive semi-conductive fabric also produced by SEFAR AG (SEFAR CarbonTex [63]). (Cheng et al., 2016, p. 100)

By creating these materials as three woven fabrics, it was possible to reduce the thickness of the surface, resulting in a sensor surface that was “soft,

thin, flexible and air permeable, thus less obtrusion to the user” (Cheng et al., 2016, p. 101). It proved to be a very efficient and effective too with “spacial resolution of 1 cm<sup>2</sup>” and “40 Hz sample rate” enabling the surface to “ detect differences from very small (e.g. ~100 g for an empty plate) to very high weights (e.g. ~100 kg when a person stands on it)” (Cheng et al., 2016, p. 101).

While probably one of the most effective sensory surfaces for a SF amongst this research, there are multiple points of discussion for this architecture. The cost of producing bespoke materials for use in large installations, as well as increased costs were this textile to remain the intellectual property of one specific company, may inhibit uptake of this implementation. This would typically mean more expense than normal carpet, and although costs would come down given time and mass production, cost remains an inhibiting factor of adoption, as discussed previously.

Furthermore, it would work in its current form as a layer under the floor covering of carpet, rather than integrated into it. This would allow for installations of various (and pliable) covering of the persons choice above this sensing surface, and not be restricted to the choices were this technology integrated into it. However, there was no testing done between this sensing fabric as a layer beneath the carpet, making the suitability and sensitivity of such a configuration unknown. Further to this, most interactions within this research seemed to be directly with the surface and human contact beside two instances where it was placed under a thin fabric seat covering and under a pliable gym floor mat. Both of these materials are not traditionally used as floor coverings.

While they did test the sensory surface underneath the fabric of the chair cover, no tests were done analysing sensory clarity when interacted by a human through another object, such as human pressure through a chair on the floor. For an inclusive and ubiquitous SF, it would need to encompass these types of scenarios where a user may only have indirect interaction with the floor, such as a user in a wheel chair, or reclining in a sofa. It could be further argued that this research was aimed at providing a clear

architecture for implementation of a sensory surface and the tests performed were to indicate performance in this regard only, not focused on application of the surface. As such, the various scenarios presented would be impractical to test for.

Despite the fact that these architectures provide an interface for human agents and creating a virtual representation of them, the ability to infer what these agents are doing would enable more accurate representation of human agents and allow for more adaptable SE. The related field known as activity recognition aims to do this and will be discussed next.

### *2.5.2 PHYSICAL ACTIVITY RECOGNITION*

Mozer, (1998) proposed the idea of a sensor-based home environment that would recognize the activities of its inhabitants and adapt accordingly. With such ambition, Kim, Helal, & Cook (2010) discuss that “The goal of activity recognition is to recognize common human activities in real-life settings” affording “societal benefits, especially in real-life, human-centric applications such as eldercare and healthcare” (p. 48). Some of the benefits of activity recognition have been identified in the objectives of various research, ranging from energy efficiency in buildings (Cottone, Gaglio, Lo Re, & Ortolani, 2015) to patient rehabilitation (Lin, Song, Xu, Cavuoto, & Xu, 2017). Klack, Möllering, Ziefle, & Schmitz-Rode (2011) used piezoelectric sensors embedded in the floor to monitor the elderly and their movement, and from this deduce any abnormal activity that may be occurring, indicating the possible need for assistance.

However, the task of “Accurate activity recognition is challenging because human activity is complex and highly diverse” (Kim et al., 2010, p. 48). Despite the challenges, the objective of activity recognition (AR) set by researchers in smart floor architectures mentioned in the previous section have all been achieved to varying degrees.

Activity recognition using technology is not a new topic for researchers (Oliver, Horvitz, & Garg, 2002), and amongst recent literature, more common methods used in this endeavour have ranged from mobile devices,

such as smartphones and wearables, (Ahmad & Nor, 2017; Kalischewski, Wagner, Velten, & Kummert, 2017; Ma & Ghasemzadeh, 2016; Sun, Zhang, Li, Guo, & Li, 2010; Sztyler, Stuckenschmidt, & Petrich, 2017; Pham, Diep, & Phuong, 2017) as well as more unique methods of activity recognition through analysis of eye movement (Bulling, Ward, Gellersen, & Troster, 2011) . However, fixed implementations such as those that are surface based or use cameras (Cheng et al., 2014; Kalischewski et al., 2017; Kolekar & Dash, 2016; Menicatti, Bruno, & Sgorbissa, 2017; R. Serra, Knittel, Croce, & Peres, 2016) seem to also be common in literature, especially when analysing whole body or limb movement to recognise activity.

In regard to the importance of AR in IoT and SE, Wu, Tseng, & Fu, (2013) noted that:

user activities (*i.e.* how users interact with the IoT-based home environment) are the most critical contexts existing in an IoT-based context-aware smart home since users are the center of homes and different user activities usually call for different services. Because of this, activity recognition (AR) becomes the most essential part in context inference mechanism for IoT-based context-aware smart homes. (p. 406)

Interestingly, before the notion of the IoT and SO's become fashionable, researchers were already looking at the implications of tagged objects within an environment being an indicator of human activity. Tapia, Intille, & Larson, (2004) highlight the efficiency of this method by saying that activity recognition “may be more easily recognized not by watching for patterns in how people move but instead by watching for patterns in how people move things” further explaining that “performing activities such as grooming, cooking, and socializing may exhibit more consistency than the way the person moves the limbs” (p. 159). However, a system relying on analysis of object and human interaction to recognise human activity may be more suitable in a multimodal approach, alleviating any deficiency in the activity recognition system due to the absence of human and object

interaction, but allowing more accuracy in recognition of certain activities where objects are involved.

This only promotes the importance of a more general and inclusive system, such as a smart floor, where object use is arbitrary to successful activity recognition, and where all interaction, whether indirectly or directly, involve the ground. As Cheng et al. (2016) similarly observe, “Virtually all human activities involve interaction with surfaces. At the very least, due to gravity, some parts of the body need to be in contact with a supporting surface (ground, chair, bed, etc.)” (p. 97). A smart floor, acting as an interface between humans and the IoT though, would leverage this and have the ability to identify, track, and recognise human activity, allowing digitisation and data on an important component of the IoT. Importantly the development and suitability of a smart floor in activity recognition can be “explained through their inherent unique features including transparency (hidden from the user), reliability (accurate information given to the user), durability (system performance not decreasing over a period of time) and multitasking (being able to satisfy several applications at the same time)” (Serra et al., 2016, p. 5757).

In this regard, activity recognition through a singular modality smart floor can be achieved by analysing the “vibrations, changes in centre of gravity and balance shifts [that] propagate throughout the entire body, causing for example hand actions to influence the pressure distribution of the bottom of the feet on the ground” (Cheng et al., 2016, p. 97). Using this methodology in assessing seven various activities of the upper body, they were able to achieve “a person dependent accuracy of 81.0% using 10-fold cross-validation and a person independent accuracy of 78.7” and when reduced to five activities “emphasising the main directions, viz. up, down, left, right and middle, the person dependent and independent accuracy grows to 86.3% and 83.6%, respectively” (Cheng et al., 2016, p. 109).

Accurate activity recognition such as those shown above rely on analysis of data given by the sensors, regardless of the modality used to produce this data. As mentioned by one author, “a central element when designing a

smart system lies in the sensing elements. A second element of equal importance is proper control or the processing of all gathered data retrieved by sensors. The quality, the accuracy and the number of sensors determine how smart an algorithm has to be to satisfy an application with good performance ratios” (Serra et al., 2016, p. 5760). The next section will discuss techniques from which inferences on activities are determined.

## 2.6 BIG DATA ANALYTICS

IBM calculates that 2.5 quintillion bytes of data is created daily, staggering numbers which mean that “90 percent of the data in the world today were created in the past two years”, which is attributed to “the explosion of mobile phones and other devices that generate data, the Internet of Things (e.g., smart refrigerators), and metadata (data about data)” (Spencer, 2016, p. 27). “These massive amounts of recently created digital data are often referred to as big data” (Alharthi, Krotov, & Bowman, 2017, p. 285).

According to the widely accepted definition by Gartner, (“What Is Big Data? - Gartner IT Glossary - Big Data,” n.d.) Big Data is characterized by three components; Volume, Variety, and Velocity. Alharthi et al., (2017) use the following descriptions to further define what these entail:

*Volume* refers to the vast quantity of structured and unstructured data that is hard to collect, manage, and analyze with the existing IT infrastructure and tools; thus, these massive data sets require new and innovative tools and approaches for capturing, storing, and analyzing data. *Variety* refers to the fact that the data comes from various sources such as spreadsheets, traditional databases, text documents, and digital data streams. *Velocity* refers to the fact that these big data sets are often comprised of and continuously expanded by real-time data streams. (p. 286)

Variety of data is already evident in the previous discussion of IoT, SE, SO, and SF, and relates to the lack of standardisation across IoT as well as the heterogeneity of ‘things’ within the IoT paradigm.

Velocity has also been briefly discussed, with issues relating to data transferal bandwidth as well computation in the cloud or at the edge:

Big data and analytics in IoT require streaming events on the fly and storing streaming data in an operational database. Given that much of these unstructured data are streamed directly from web-enabled “things”, big data implementations must perform analytics with real-time queries to help organizations obtain insights quickly, rapidly make decisions, and interact with people and other devices in real time. (Ahmed et al., 2017, p. 464)

With classical centrally based models, latency becomes a factor in effective services. This is why research has been done on methods, such as MapReduce published by Google, that effectively decrease the need for data transferal bandwidth use by having the computation done by multiple nodes rather than a central point, explaining why edge analysis is important (Lu, Wang, Wu, & Qiu, 2017). Some noteworthy points on data bandwidth are discussed by Bakshi (2016) who writes:

not all the data or sampled data is important, hence summarized or aggregated data can suffice. Additionally, in a network bandwidth with constrained environments, it may not be feasible to move large volumes of data in a timely fashion to a central location like a data center, where one can store and analyze. Furthermore, several use cases require instant access of analytics for insight at the edge location, which cannot wait for delays and lags for results from a central analytics system. (p. 2)

Thus, a SF detecting and recognizing user activities instantly would benefit from an analysis at the edge, avoiding latency. Furthermore, it could be argued that monitoring the changes in a human’s activity would be more important than constant data streams if the same activity is being repeated. Therefore, conducting analysis at the edge would mean limiting data transferal to a central location for these activity changes or in events when more IoT system services need to be induced, keeping edge activity recognition instantaneous and reducing bandwidth need. It would also

distribute the computing power needed for analysis, able to be done at the edge with multiple SF embedded with appropriate computing power, scaling with SF size accordingly. All these would be immensely important in large-scale multi-user environments with multitudes of data production (Ahmed & Rehmani, 2017).

Some of these factors are also relevant in Volume, which incorporates all points of data existence, from collection to storage to analysis. For example, the copious amounts of data have forced new paradigms of data storage and management, away from traditional organized data stores to architectures like data lakes. It has become impractical to structure data using existing paradigms, creating a necessity for evolving data stores consisting of “both structured and unstructured raw data. Data structures emerge with usage over time [and] ... provides flexibility and agility to deal with business requirements in a dynamic environment.” (Halter, Kromer, Kutemperor, & Soares, 2016, p. 44).

Even though storage of data, along with the related reliability and privacy issues are important, the opportunities in IoT data lie with the inferences drawn from the data produced by ‘Things’ through various data analytics. Since “the data analytics spending alone [is] estimated to be worth \$500 billion by 2020, there are real incentives for all sectors to be involved with this technology” (Accenture, 2015, "Introduction", para. 1). This is to be expected, considering the ubiquity and proliferation of things consisting of the IoT and the possibility of value of data extrapolation.

Ahmed et al. (2017) describe some of applications and benefits of data analysis of IoT systems shown in Table 3.

**Table 3: Benefits of data analytics for IoT applications (Ahmed et al., 2017, p. 466)**

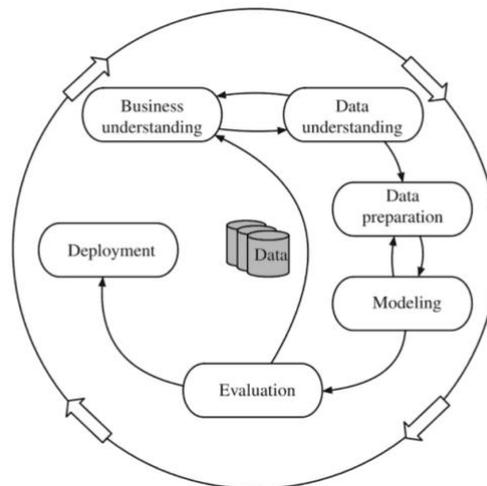
<b>IoT application</b>	<b>Benefits of data analytics</b>
<b>Smart transportation</b>	(a) Reduce the number of accidents by looking into the history of the mishaps (b) Minimize traffic congestion (c) Optimize shipment movements (d) Ensure road safety
<b>Smart healthcare</b>	(a) Predict epidemics, cures, and disease (b) Help insurance companies make better policies (c) Pick up the warning signs of any serious illnesses during their early stages
<b>Smart grid</b>	(a) Help design an optimal pricing plan according to the current power consumption (b) Predict future supply needs (c) Ensure an appropriate level of electricity supply
<b>Smart inventory system</b>	(a) Detect fraudulent cases (b) Strategically place an advertisement (c) Understand customer needs (d) Identify potential risks

While the benefits of data analytics can be seen in this table and demonstrates the true value of data from the IoT, not all data is created equal. The IDC released a report stating:

In 2013, only 22% of the information in the digital universe would be a candidate for analysis, i.e., useful if it were tagged (more often than not, we know little about the data, unless it is somehow characterised or tagged – a practice that results in metadata); less than 5% of that was actually analyzed. By 2020, the useful percentage could grow to more than 35%, mostly because of the growth of data from embedded systems (IDC, 2014, "New Findings", para. 4).

This suggests that data is simply being collected without the current necessity, business intelligence, or capability to process it all. There are other barriers to Big Data, such as the human element, privacy, infrastructure, and organisational readiness among others, as well as data complexity (Alharthi et al., 2017). However, big data complexity can be

addressed by Data Mining (DM), “the process of discovering patterns in data” (Witten, Frank, Hall, & Pal, 2016, p. 6). In DM, there is huge potential for machine learning (ML) techniques, along with current computing power, to analyse and extrapolate these patterns from vast streams of data produced in the IoT, thus providing value to the data (Alharthi et al., 2017). In discussion of the life cycle of a DM project, Witten et al. (2016) use the diagram in Figure 1 to describe the various components involved; it demonstrates that the data itself, while central to DM as a whole, is superfluous to the initial development and path of the DM project lifecycle, as data may exist prior to or during realisation of the application or business understanding.



*Figure 1: Life Cycle of a Data Mining project (Witten et al., 2016, p. 29)*

Thus, their model begins with business understanding and data understanding, which they describe as:

investigating the business objectives and requirements, deciding whether data mining can be applied to meet them, and determining what kind of data can be collected to build a deployable model. In the next phase, “data understanding,” an initial dataset is established and studied to see whether it is suitable for further processing. (p. 29)

Clear business objectives and understanding of data may be the instigators of a DM project in realisation of application, but in regard to the data itself, it is the data preparation and modelling where ML helps alleviate the challenge of analysing Big Data.

In regard to SF Analytical algorithms for data inferences and associated machine learning techniques, those that can be used for SF activity recognition are of importance and will be discussed next.

### *2.6.1 MACHINE LEARNING*

While there was undoubtedly attributing research and development beforehand, one of the early instances of successful Machine Learning (ML) is attributed to Arthur Samuel, who wrote the first computer learning program for the game of checkers while at IBM in 1952 (Arthur L. Samuel, 2013). Rather than programming the steps and choices needed to for the computer to win, the computer adapted and improved its program with the more games it played, assimilating strategies that allowed it to win. This adapting functionality of improvement is at the core, what ML entails.

A subset of Artificial Intelligence (AI), Machine Learning is “essentially a form of applied statistics with increased emphasis on the use of computers to statistically estimate complicated functions and a decreased emphasis on proving confidence intervals around these functions” (Goodfellow, Bengio, & Courville, 2016, p. 96). Because of this, ML applications extend to diverse domains where DM may provide benefit, including healthcare (Kavakiotis et al., 2017; Nair, Shetty, & Shetty, 2017) environmental management (Park, Mukherjee, & Zhao, 2017; Hsieh, Cannon, Lima, Hsieh, & Cannon, 2015) energy efficiency (Essl, Ortner, Haas, & Hettegger, 2017; Rajasekaran, Manikandaraj, & Kamaleshwar, 2017) and even in detection of art forgeries (Polatkan, Jafarpour, Brasoveanu, Hughes, & Daubechies, 2009), to name a few. Its use in uncovering value in data of different domains is in pattern recognition, explained succinctly by Mullainathan & Spiess, (2017) who write:

The appeal of machine learning is that it manages to uncover generalizable patterns. In fact, the success of machine learning at intelligence tasks is largely due to its ability to discover complex structure that was not specified in advance. It manages to fit complex and very flexible functional forms to the data without simply overfitting; it finds functions that work well out-of-sample. (p. 88)

However, ML still requires thought for optimal outcomes. While specific algorithm discussion is out of the scope of this research, selecting the most appropriate Machine Learning algorithm is very dependent on the nature of data and objectives. As Iqbal Muhammad & Zhu Yan (2015) found in activity recognition applications:

The performance of SVM and Neural Networks is better when dealing with multidimensions and continuous features. While logic-based systems tend to perform better when dealing with discrete/categorical features. For neural network models and SVMs, a large sample size is required in order to achieve its maximum prediction accuracy whereas NB may need a relatively small dataset. (p. 951)

Related to this dataset size, there are usually three categories of Machine Learning associated with activity recognition data analysis reliant on the type of data and conversely the meta-data available. These are supervised, unsupervised, and semi-supervised. Supervised learning has all the data labelled, where “The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data” either by predicting the category (y) of data (x) (classification), or where the “output variable is a real value, such as “dollars” or “weight”” (regression); unsupervised learning has all the data unlabelled and the objective is “to model the underlying structure or distribution in the data in order to learn more”, such as “grouping customers by purchasing behaviour” (clustering), or to “discover rules that describe large portions of your data, such as people that buy X also tend to buy Y”

(association) ; Semi-supervised learning contains a mixture of both unlabelled and labelled data, meaning both unsupervised and supervised learning techniques can be used (Brownlee, 2016). With this delineation of data in mind and noting the importance of Big Data in the IoT and ML's role in data analytics, the following section will discuss these approaches in relation to AR in SF architectures.

### *2.6.2 MACHINE LEARNING IN ACTIVITY RECOGNITION*

As one author notes "HAR [human activity recognition] can be treated as a typical pattern recognition (PR) problem" through the use of "machine learning algorithms such as decision tree, support vector machine, naive Bayes and k nearest neighbours" (Lara & Labrador, 2013). The majority of AR research to date, including floor sensory systems, uses a supervised approach in ML. For example, Muheidat & Tyrer (2016) used a "Naïve Bayes, Logistic, Multilayer perceptron, and decision tree J48 (also known as C4.5 classifier)" (p.5357). Cheng et al. (2014) used a "Random Forest classifier using 10-fold cross validation in WEKA" (p. 149). Both examples used labelled data. While it is out of the scope of this paper to discuss all possible ML algorithms other than those used in related floor sensory activity recognition systems, Guan, Ma, Yuan, Lee, & Sarkar, (2011) discuss researcher's preference for supervised approaches in activity recognition research:

Supervised learning requires labeled data. Conversely, unsupervised learning tries to directly construct models from unlabeled data, either by estimating the properties of its underlying probability density or by discovering groups of similar examples. Because of using labeled data, supervised learning usually outperforms unsupervised learning; therefore, supervised learning is the predominant approach for WSAR [wearable sensor based activity recognition]. (p. 423)

Therefore, where possible, supervised learning is seen as the preferred appropriate approach. However, while agreeing that the majority of AR approaches use supervised methods, Gu, Chen, Tao, & Lu, (2010) discuss

the benefits of the alternative, unsupervised approach in regard to activity recognition:

learning from training data typically requires human labeling. Application developers are required to label both the underlying sensor system and the activities associated with a set of training data. Considering a large number of activities to be recognized in our daily lives, manual labelling of training data may place a significant burden to any individual involved in data collection. Hence, supervised learning approaches may have limitations in real-life deployment where scalability, applicability and adaptability are highly concerned. (p. 534)

In their implementation of an unsupervised learning model for AR, they were able to achieve an accuracy of 91.4%, comparable to the supervised implementation they tested at 93.5% accuracy. While these results would seem to qualify the suitability of an unsupervised approach, it must be noted that their implementation of an unsupervised learning model for activity recognition was accomplished with RFID tagged objects and their use along with wearable readers, an issue for a ubiquitous and unobtrusive computing taxonomy. However, others have also successfully used unsupervised methods in combination with various sensors, such as accelerometers attached to a user's chest (Trabelsi, Mohammed, Chamroukhi, Oukhellou, & Amirat, 2013) or the accelerometers in smart phones (Kwon, Kang, & Bae, 2014).

However, rather than resorting to unsupervised methods, semi-supervised models could overcome the scalability and human domain specialist issues apparent in labelling. Bicocchi, Mamei, Prati, Cucchiara, & Zambonelli, (2008) demonstrate the use of this taxonomy in AR via two sets of sensors (in this case, a camera and accelerometer); the camera is trained using classification algorithms and “can sample data from the environment and produce high level events based on the classification of their inputs”, with labels uploaded to the tuple space; the worn accelerometer sensor is untrained, simply collecting data, until it “enters the camera field of view,

[where] it subscribes to the tuple space to receive the labels coming from the camera with information about the user motion state’, meaning that the “Acceleration data and camera-based classification are paired together and a model to classify the user motion state on the basis of the sampled acceleration data is built on the fly” (Bicocchi et al., 2008, p. 63). This can then be used to further train the classification model, in essence creating a self-labelling learning system. However as Wu et al., (2013) caution, “this method will work well only if the performance of the initial classifier is good enough since the wrong classification result will degrade the accuracy of the newly trained classifier” (p. 461), which can be an issue with such single-classifier semi-supervised approaches.

Therefore, increasing classifier confidence for labelling in semi-supervised activity recognition can be achieved through the use of multiple classifiers; either by using two independent classifiers in a multi-view method (where feature-independent classifiers are used for a single classification) and choosing the label for new data by the classifier with higher confidence (co-training), or using multiple classifiers in a single-view method (with the same feature vector but different ML algorithms) and using the consensus of all the classifiers output for any new data labelling (co-learning) (Wu et al., 2013). For both of these methods, the newly labelled data can be added to the training set to further increase classifier confidence.

An example of such an implementation is conducted by Wu et al. (2013), who in relation to smart homes activity recognition, describe two features form which activities can be deduced:

*Spatial Features:* refer to how an activity interacts with smart homes, *e.g.* what appliances are involved in, how these appliances are used, what effects are caused, ..., etc.

*Temporal Features:* refer to when and how often an activity occurs in smart homes, *e.g.* the daily total occurrence duration, the average occurrence duration, the daily occurrence times, the usual occurrence time, ..., etc. (p. 461)

In their proposal, a supervised approach is first conducted from their sensory data to build an initial classification model. Any new data is first analysed using this classification model and labelled appropriately if similarity with an existing action is sufficient, further undergoing a temporal feature verification analysis based on the AR models. If the temporal analysis is not what is expected, it will be added to cluster of activities based on temporal features. However, if the initial classification failed to recognise the activity, a spatial feature analysis will be performed to create clusters of similar, unrecognised activities. Then, pseudo-activities are created by grouping both clusters together, performing spatial and temporal feature analysis on these, before comparing the analysis with known temporal and spatial features of AR models. The result can be used to update these models for future AR of new data.

More recently, there has been some focus on Deep Learning methods, a subset of ML, to facilitate AR. Other than acknowledging work in this field, the topic is beyond the scope of this research. However, it is important to note the reasoning behind this modality shift in ML in relation to AR. As Chen, Hoey, Nugent, Cook, & Yu, 2012 (p. 801) note:

Current work on activity recognition has mainly focused on simplified use scenarios involving single-user single-activity recognition. In real-world situations, human activities are often performed in complex manners. These include, for example, that a single actor performs interleaved and concurrent activities, multiple actors perform a cooperative activity, and/or a group of actors interact with each other to perform joint multiple activities. The approaches and algorithms described in previous sections cannot be applied directly to these application scenarios. As such, research focus on activity recognition has shifted toward this new dimension of investigation. (p. 801)

In relation to a SF becoming an interface to the IoT in a ubiquitous and pervasive fashion, with many interactions, activities, smart-objects, and human and non-human agents, the challenge to compute AR accurately will

rely on deep learning as it “reduces the dependency on human-crafted feature extraction and achieves better performance by automatically learning high-level representations of the sensor reading” (J. Wang, Chen, Hao, Peng, & Hu, 2017). Future work would likely form interesting outcomes in this area.

## 2.7 SUMMARY

The first portion of this chapter provides the underlying inspiration for the research, starting with a brief history of the IoT, encompassing technologies and related technological paradigms and the evolving of these paradigms in more modern literature. While evaluating this literature, what has become paramount to the discussion in this research is the need for human inclusion in the IoT via an interface that adheres to the ubiquitous and pervasiveness nature prevalent in literature. Further to this, understanding the implications of including humans within the IoT framework warrants discussion of some of the factors limiting acceptance and willingness for such inclusion, for example autonomy.

As demonstrated in the reasoning through the later portion of this chapter, this research assumes two things. Firstly, that that a ground-based interface overcomes the majority of factors raised in the previous discussion and is thus identified as being a suitable candidate for a human IoT interface. Secondly, that activity recognition is a vital extension in providing digitisation of biological, analogue mechanics via the ground-based interface, allowing the required interface for humans in the IoT. Analysis of current research within this domain highlights progress made thus far, as well as inefficiencies in complexity, cost, pervasiveness, and accuracy in some instances. Therefore, this research aims to address some of these inefficiencies through investigation of one possible implementation of a ground-based activity recognition interface.

### 3 METHODOLOGY

This chapter discusses the research objectives and the underlying research question. The methodology used and the structure of the research are also reviewed.

#### 3.1 RESEARCH OBJECTIVES

Activity recognition (AR) is an active field of interest for many researchers. However, on the basis of literature reviewed in Chapter 2, there is clearly scope for further research in AR with specific application within the IoT paradigm, beyond the current existence within isolated environments or predominantly driven by technical opportunities.

Therefore, this research will address AR through the IoT lens. This encompasses creating an AR interface that would serve the ideals of ubiquity, pervasiveness, unobtrusiveness, and connectivity foremost. Furthermore, simplicity, adaptability, and cost are factors considered to mitigate issues related to acceptance and deployment within the general public. In regard to accuracy, reliability, and timeliness related to real-time AR, these will be considered within an IoT framework, rather than determined by an exercise in technological discovery. With such a large scope and many varying environments for possible application, this research centres on investigating AR within a typical environment in the context of workplace scenario, and more specifically in the confines of an office desk space using only a ground-based surface as indicators of human activity. However, this focus does not limit the applicability of the approach in future endeavours.

Given these factors, the question addressed by this research is as follows: To what extent can a simple ground-based sensor acting as an IoT device be used to identify human motion?

This notion of simplicity is in regard to the sensor being easily reproducible and inexpensive, using freely available components rather than bespoke manufactured materials. It also relates to the simplicity in use, or the

invisibility of the sensor for the user, while considering the ability to reliably predict activities without user disturbance.

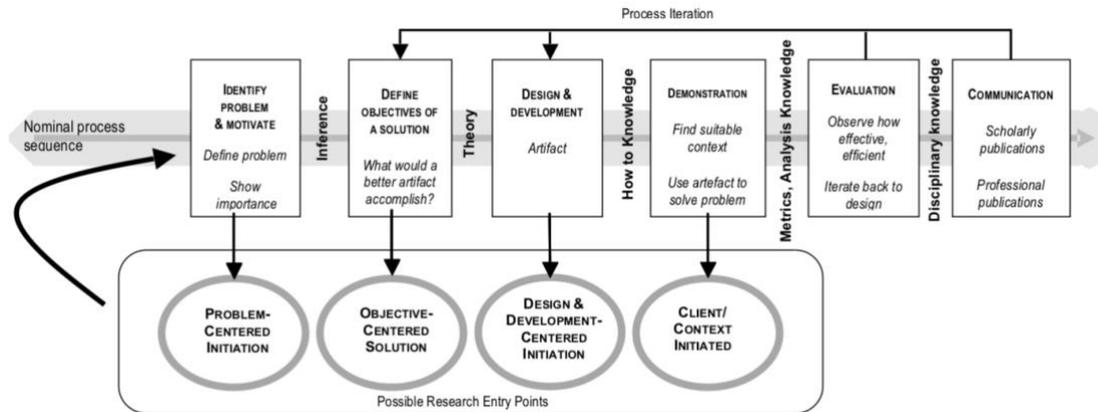
### 3.2 RESEARCH METHODOLOGY

While the analysis of related literature aided in the understanding of a particular problem with current IoT interfaces and the potential of an AR system to alleviate some of the problems, the development and implementation path of such an interface pursuant to the ideals mentioned in the research objective is unclear. Thus, what is needed is an approach that allows for iterative refinement toward development of an artefact, with each iteration informing the next possible approach towards realisation of the research objective. In this vein, Design Science Research is chosen as the research methodology.

Within Information Systems (IS) Research, Design Science Research (DSR) is a methodology that is widely used. Given this research focuses on the paradigm of IoT, DSR is used to formulate the research approach for this paper. Furthermore, this methodology is appropriate for use where artefacts are developed to provide insight into an identified problem and add to the knowledge of the field (Hevner, March, Park, & Ram, 2004). Because of the constructivist and pragmatic nature of research conducted under the paradigm of DSR, the pitfall of falling into routine design is evident. It is therefore important to consider that valid DSR research must extend knowledge in the related field through this iterative development and evaluation of an artefact.

Development of an artefact that enables knowledge contribution is achieved through iterative and incremental research cycles, where a possible solution to the identified problem can be created and evaluated. It is viable for this iterative process to cease at any time, completing the research (Hevner et al., 2004). In the research presented in this thesis, incremental development of both hardware and software is undertaken in conjunction with evaluation of this interface. Using quantifiable statistical analysis leads to further understanding of the feasibility a ground-based interface provides in digitising human activity within a wider IoT framework. Shown in Figure

2, this iterative process is defined succinctly by Peffers, Tuunanen, Rothenberger, & Chatterjee, (2007).



*Figure 2: The DSR Process (Peffers et al., 2007, p. 54)*

In terms of this research, identifying the problem was conducted through an objective-centred research entry point, this being the inclusion of humans more successfully into the IoT. Through consideration of the literature to further define the problem, obvious objectives of a potential solution became apparent, and these objectives were discussed within the literature review. Further to this, the literature informed the possibility of AR from which human inclusion within the IoT could be achieved and required further literature dissemination on this field. The development, demonstration, and evaluation of the artefact are further elaborated on in the following sections.

### 3.3 RESEARCH DESIGN

The guidelines for a DSR approach are described by Hevner, March, Park, & Ram, (2004) and consist of the seven elements shown in Table 4. These guidelines will be used to form the research design and are discussed in this section.

*Table 4: Design-Science Research Guidelines (Hevner et al., 2004, p. 83)*

<b>Guideline</b>	<b>Description</b>
<b>Guideline 1: Design as an Artifact</b>	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
<b>Guideline 2: Problem Relevance</b>	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
<b>Guideline 3: Design Evaluation</b>	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
<b>Guideline 4: Research Contributions</b>	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
<b>Guideline 5: Research Rigor</b>	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
<b>Guideline 6: Design as a Search Process</b>	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
<b>Guideline 7: Communication of Research</b>	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

### 3.3.1 ARTEFACT

Foremost within the DSR approach is the construction of an artefact that provides new knowledge within the field, and possibly prompts further research into associated problems (Hevner, 2007). The proposed artefact in

this research is a ground-based sensory system capable of detecting movement within a small environment and that uses AR to digitise human agents for inclusion into a larger system, namely the IoT. It is important to note that while this artefact can operate and be evaluated independently of the IoT, the solution objectives outlined earlier insist on developing the artefact in accordance with the predetermined notion of inclusivity within the IoT. This is why such instantiations requiring cameras or bespoke flooring supports have already been discounted as possible solutions, as an artefact not adhering to the IoT principles discussed and only concerned with AR would not have these same directives or limitations to consider. These objectives for consideration were discussed within the literature review and are: Invisibility, Unobtrusive, Complexity, Accuracy, Cost, Reliability, Responsiveness, Adaptability, and Connectivity. These are applicable to the ground-based surface, software implementation, and human perception alike. The artefact introduced in this research will fulfil the guidelines in DSR by providing an instantiation of the proposed solution with these objectives in mind.

### *3.3.2 PROBLEM RELEVANCE*

DSR dictates that technology based solutions provide knowledge on real-world problems and further that there is emphasis on “construction of innovative artifacts aimed at changing the phenomena that occur” (Hevner et al., 2004, p. 84). As discussed, this research aims to provide insight into possible barriers of human inclusion within the IoT and change this by providing an interface that adheres to the principals of the IoT paradigm that arose in the literature discussion. The relevance exists in realisation of limited instantiations of AR in respect to the IoT, and with the IoT experiencing exponential growth, the importance of providing this interface that can be implemented in various applications is apparent. While this research does not intend to provide a definitive solution, it aims to address some of these issues and allow avenues for further discussion and research through its instantiation.

### *3.3.3 DESIGN EVALUATION*

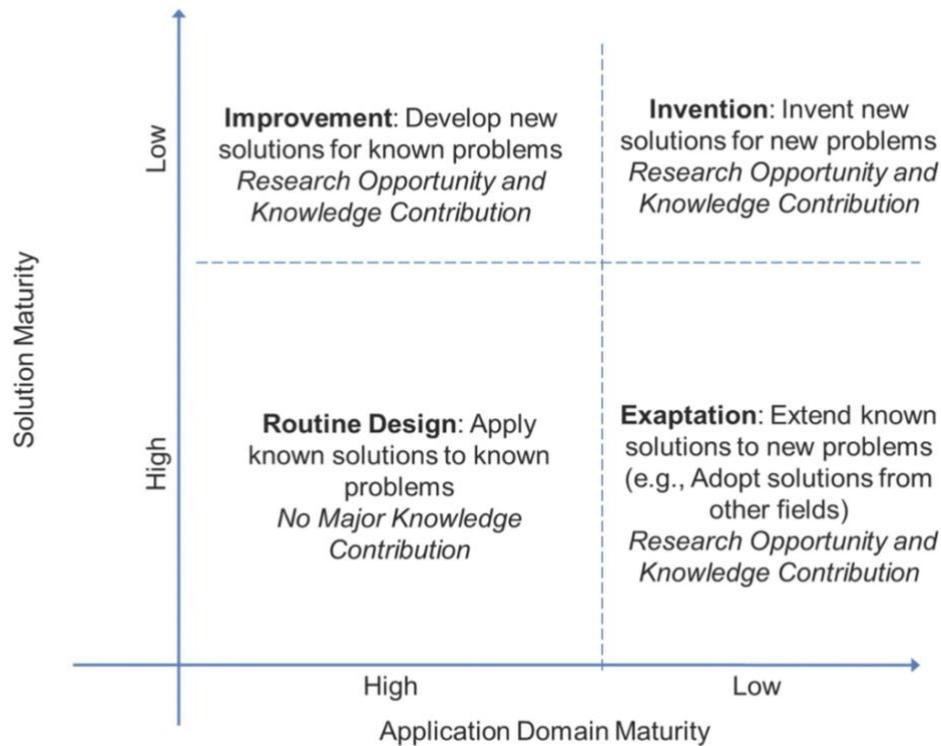
Rigorous evaluation of the artefact involves analysis “in terms of functionality, completeness, consistency, accuracy, performance, reliability, usability, fit with the organization, and other relevant quality attributes” (Hevner et al., 2004, p. 85). There are two components that need to be evaluated in this research, the physical sensory system in relation to IoT objectives and human integration in conjunction with the AR component. A typical workplace environment chosen as the usage scenario will help in observance of the objectives in the evaluation phase and determine the appropriate methods of evaluation. The qualitative assessment in the evaluation of the physical properties pertaining to the ground-based sensor are performed using video-based observation and feedback. The AR component uses a quantitative approach, with statistical significance implying accuracy of predicted activities. For both of these, it is important to note that:

Because design-science artifacts are often the machine part of the human machine system constituting an information system, it is imperative to understand why an artifact works or does not work to enable new artifacts to be constructed that exploit the former and avoid the latter. (Hevner et al., 2004, p. 88)

Useful in the initiation of DSR and a new artefact, this emphasis on evaluation is able to applied continuously throughout the iterative development of a single artefact, with each iteration able to inform incremental improvements until the research is completed.

### *3.3.4 RESEARCH CONTRIBUTION*

Gregor & Hevner (2013) provide a simple framework for assessment of viable outcomes following a DSR approach. This framework is shown in Figure 3.



**Figure 3:** Knowledge contribution assessment (Gregor & Hevner, 2013, p. 345)

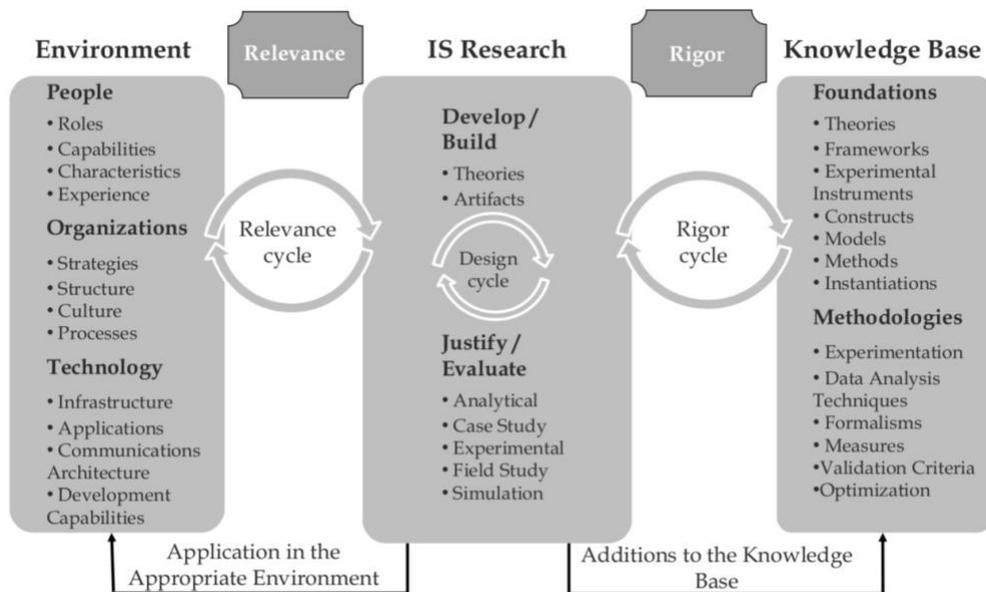
The research must show either improvement, invention, or exaptation to be considered as contributing knowledge. The creation of an artefact “may extend the knowledge base or apply existing knowledge in new and innovative ways”, or be “creative development of novel, appropriately evaluated constructs, models, methods, or instantiations that extend and improve the existing foundations in the design- science knowledge base” (Hevner et al., 2004, p. 87). The research within this paper could be considered to be exaptation, using the field of AR within an IoT framework, and improvement by further development of existing AR technology in respect to social inhibitors that encompass the IoT paradigm and necessary objectives.

### 3.3.5 RESEARCH RIGOR

While “Design-science researchers must constantly assess the appropriateness of their metrics and the construction of effective metrics” (Hevner et al., 2004, p. 88), DSR is highly pragmatic in its approach to research, ultimately leaving appropriate method selection to the researcher.

For example, Venable, Pries-Heje, & Baskerville, 2014 describe two terms for method application within a DSR approach; Artificial evaluation and Naturalistic evaluation. Artificial evaluation “includes laboratory experiments, simulations, criteria-based analysis, theoretical arguments, and mathematical proofs” (p. 80) while Naturalistic evaluation methods involve “case studies, field studies, field experiments, surveys, ethnography, phenomenology, hermeneutic methods, and action research” (p. 81). The use of seemingly dichotomous methods is apparent in pragmatic approaches such as DSR and while artefacts within DSR are usually evaluated using scientific rigor via statistical analysis and mathematical means, artefact evaluation within its intended environment may employ the use of more qualitative methods (Vaishnavi & Kuechler, 2015).

In elaboration of Hevner’s three cycle approach to DSR (cf. A. Hevner & Chatterjee, 2010, p. 16), the diagram of Marheineke (2016) shown in Figure 4 highlights the importance of using the correct methods in establishing rigor within the domain of Information System research.



**Figure 4:** Information system research framework (Marheineke, 2016, p. 14)

However, this diagram shows that rigor is not only reliant on the appropriate choice of methods. The use of a “Rigor Cycle” in DSR “provides past knowledge to the research project to ensure its innovation” and is

“contingent on the researchers to thoroughly research and reference the knowledge base in order to guarantee that the designs produced are research contributions and not routine designs based on the application of known design processes and the appropriation of known design artifacts” (A. Hevner & Chatterjee, 2010, p. 18). The literature review and exploration of relevant existing solutions undertaken in the previous section are used to inform and ground the research, as DSR is “grounded on existing ideas drawn from the domain knowledge base” (A. Hevner & Chatterjee, 2010, p. 18).

### 3.3.6 DESIGN AS A SEARCH PROCESS

A search process within DSR is primarily to “discover an effective solution to a problem” by “utilizing available means to reach desired ends while satisfying laws existing in the environment” (Hevner et al., 2004, p. 88). In further defining of this; “Means are the set of actions and resources available to construct a solution. Ends represent goals and constraints on the solution. Laws are uncontrollable forces in the environment” (Hevner et al., 2004, p. 88). Further to this, Hevner (2004) explains methods to address means, ends, and laws:

Means are represented by decision variables whose values constitute an implementable design solution. Ends are represented using a utility function and constraints that can be expressed in terms of decision variables and constants. Laws are represented by the values of constants used in the utility function and constraints. (Hevner et al., 2004, p. 89)

In this research, means are discussed in the following artefact design section, while ends are established in the research objectives and further elaborated in evaluation of the artefact and consideration of its suitability in fulfilling the objectives. Laws are the constraints placed within development of the artefact (including IoT constraints and use-case constraints) established in this section and adhered to in the development of the artefact. Furthermore, these three aspects are informed from the literature review.

Interestingly, while the initial scope of the research is vast and would seem to be implausible to address, DSR recognises that desired ends may often be simplifications of a “problem by explicitly representing only a subset of the relevant means, ends, and laws or by decomposing a problem into simpler subproblems” (Hevner et al., 2004, p. 89). This research does not intend to evaluate all possible ground-based interfaces within multiple environments, but as stated, abstracts principles applicable to wider environments for implementation in a more restricted environment within the office workplace to enable “constructing an artifact that works well for the specified class of problems (Hevner et al., 2004, p. 89). This is common concept known as satisficing (Simon, 1996). Assurance that the artefact works can be achieved through the methods of evaluation which have been discussed previously, and although not generalizable across all environments, the purpose of the design process and implementation of the artefact is to “first establish that it does work and to characterize the environments in which it works” to allow “practitioners to take advantage of the artifact to improve practice and provides a context for additional research aimed at more fully explicating the resultant phenomena” (Hevner et al., 2004, p. 90). It is the intention of this research to provide avenues for further investigation within this field using improved implementations of the artefact.

### *3.3.7 COMMUNICATE RESEARCH*

Research finding should be communicated to audiences who are both technology-orientated, such as domain experts, researchers and academics, as well as those who are less technology-oriented but may be stakeholders to some degree, such as management within specific organizational structures (Hevner et al. 2004). This is to enable technology practitioners to implement and benefit from it themselves, and for researchers to evaluate the artefact and to develop further based on the contribution to knowledge it provides. For more management-orientated audiences, the ability to understand the potential benefits for implementations into real-world business contexts adds to the importance of relevance of the artefact (Hevner et al. 2004).

This research project intends, through the development of the artefact and this thesis, to provide insight into a cost effective, reliable, consistent and unobtrusive ground based interface. Communication of the successes will be important for both parties, allowing further evaluation and research by technology practitioners, while giving reason for management-oriented audiences to implement or at least consider appropriate inclusion of AR within work environments. As seen in the literature review, some information discussed therein may already be known by technology practitioners. However, as well as establishing grounds for the investigation, it would be an oversight to assume the audience is aware of these factors, and as per DSR guidelines, is required to be discussed for the benefit of more management-orientated audiences.

## 4 ARTEFACT DESIGN AND IMPLEMENTATION

Given the objectives and the chosen research methodology, the following chapter details the implementation and development of the ground-based sensor and its ability in AR with respect to some IoT principles discussed earlier. Since this process requires iterative progression as per the DSR approach, each step will be detailed and analysed. Typically, evaluation within literature signals the conclusion of the formal portion of research and invites further discussion. However, as this research follows a DSR approach, evaluation occurs within each iteration to inform decisions within the next artefact implementation, and thus evaluation will be briefly discussed within this section. However, evaluation discussed in the next chapter will elaborate on the findings of the final implementation. Therefore, in accordance with guidelines of DSR dissertation, the following section will detail the design process and development of the artefact (Strode & Chard, 2014).

### 4.1 USE-CASE: WORKPLACE ENVIRONMENT

As discussed, a use case for implementation and evaluation of this artefact is dependent on a typical scenario within an office environment. No generalizations are intended to be discovered that would have application in multiple environments. However, the use-case should at least provide an insight into the suitability for such an artefact in this scenario with implications for other environments. As such, and according to the constraints described in section 2.4 regarding some factors of IoT usability, an office environment which features no extra peripherals other than the artefact itself are permitted. This also adheres to the notion of invisibility and pervasiveness previously discussed in section 2.1. Considering these aspects has led to the assumption that a floor based sensory system would be the most appropriate approach for inclusion of humans into the IoT. However, as discussed in section 2.5, the current SF implementations fail to address multiple factors of usability from an IoT perspective, providing opportunities for further investigation.

The decision to use this approach is also influenced by the need for the artefact to not inhibit normal human motion or activity. In the given scenario of an office environment, approximately 75% of an employee's time involves sitting at a desk, typically in front of a computer (Buckley et al., 2015). As evidenced in the literature review, most AR tends to involve spatial tracking, where variation in locations are easier to define. However, this type of tracking would not be suitable in the given scenario where most of the time is spent in one general location. Indeed, some differentiate between two types of recognition; activity recognition involving high-level, possibly multi-user tasks; and action recognition involving a single-user performing a single task (Aggarwal & Ryoo, 2011; Turaga, Chellappa, Subrahmanian, & Udrea, 2008) while others create yet more rubrics from which to define classes of activity recognition (Vrigkas et al., 2015). According to these definitions then, the artefact in this research would be more aligned to Action Recognition and Atomic Actions within the restricted environment. However, in the current use-case of this research, it is not important to differentiate between the terms as may be the need in larger environments, as no perceived large spatial movements in regard to activity recognition are being assessed. Furthermore, while it may be useful in disseminating literature, it would be irrational to disregard the relevant knowledge within implementations simply because of the various use of terminology, as most seem to use the terminology interchangeably (Castellano, Villalba, & Camurri, 2007).

Accompanying the premise that the majority of time being spent in an office workplace is seated in front of a desk, AR in this scenario using a ground-based sensor would have to accommodate indirect contact, typically through a chair. As discussed in section 2.5, while there are examples of ground-based AR sensors and those with sensors implemented in the chair, indirect AR examples are rare. Therefore, in this use-case, the ability to infer human activity indirectly is of interest and contributes to the knowledgebase within AR.

The use-case has also influenced the choice of activities that are to be recognised. Since the physical dimensions of the ground sensor are of an appropriate size for an office desk and chair setting at slightly larger than one square meter, only single user activities are researched in this paper. While multi-user implementations are important, within the given scenario these situations are less likely to occur. This does not negate the reality that larger scale implementations would need to accommodate multi-user environments and high-level activities but are not covered as they are beyond the scope of this paper.

Given that single-user activities are being assessed within a small space and based on a general observation made, the types of activities for recognition are as follows:

- Neutral – Sitting upright on the chair
- Relaxed – Leaning back on the chair
- Typing – Using the computer keyboard with two hands while seated
- Mouse – Using the mouse of the computer (right hand only), seated
- Left – Any activity that requires the user to interact to the left of the keyboard
- Right – Any activity that requires the user to interact to the right of the keyboard
- Stand – The user is no longer seated but still within the sensors space
- Away – The user is no longer in the space

It is important to note that these actives are mutually exclusive and do not account for all activities observed in this environment. Furthermore, combinations of activities are categorised into either left or right and could have been further divided into more specific activities. However, if the recognition of these simple activities is successful, it is likely that more simple activities could be added to for recognition, and higher-level activities in this environment (e.g. talking on the phone) could be inferred, thus resulting in increased and consistent function both in environment and recognition mutability. Otherwise, these actions were also chosen to represent a different range of variance between expected sensor readings,

from spatially similar actions (e.g. mouse and keyboard use) and spatially distinct actions (e.g. left and right).

## 4.2 INITIAL EXPLORATION

Initiation of the artefact construction included investigation into existing instantiations into human inclusion within the IoT. As discussed in the literature review, the need for this inclusion is a necessary component for successful IoT pervasiveness, and is usually accomplished through interactions with smart objects, particularly the smart phone. Conversely, the field of AR is and has been of interest for many researchers, and while AR holds promise in digitisation of human activities into more complex systems, the emphasis on technological advancement in accuracy and reliability lead to instantiations that would be difficult to implement in real world situations. From the perspective of IoT principles, including pervasiveness and invisibility, the majority of these implementations are unsuitable. Furthermore, cost and complexity issues in some of these instantiations also create barriers to acceptance and therefore pervasiveness.

Through investigation of other implementation in the literature review, it was determined that six components would need to be explored in realisation of the research objectives. These include a mechanism for pressure sensing and correlated spatial pressure awareness, materials enclosing the pressure sensing material and providing a surface for user and chair contact, as well as to stabilize the sensor on the ground. Investigation into data acquisition from the sensor, and visualisation or storage of the data are other components to consider. Finally, exploration of a suitable machine learning algorithm for use with the artefact as part of the sensor design, but as this is indicative of the sensors performance in the evaluation phase, this will be elaborated on separately.

#### *4.2.1 SEMI-CONDUCTIVE POLYMER COMPOSITE FOR PIEZORESISTIVE PRESSURE SENSING*

The initial goal was to obtain a material that allowed pressure sensing that was both simple to implement and cheap and easy to obtain, and as selection of the pressure sensing material affects other decisions made, such as the choice of an appropriate material for enclosing the sensing layer. An implementation offered by J. Cheng et al. (2014) used a conductive foam that changed resistance when pressure is applied. Use of this material has advantages in cost effectiveness while displaying reliable subtle activity recognition. It also performed its AR functions without intrusion of the user motion when direct contact with the sensor occurred. As such, an implementation involving a piezoresistive pressure sensor based on conductive foam was implemented.

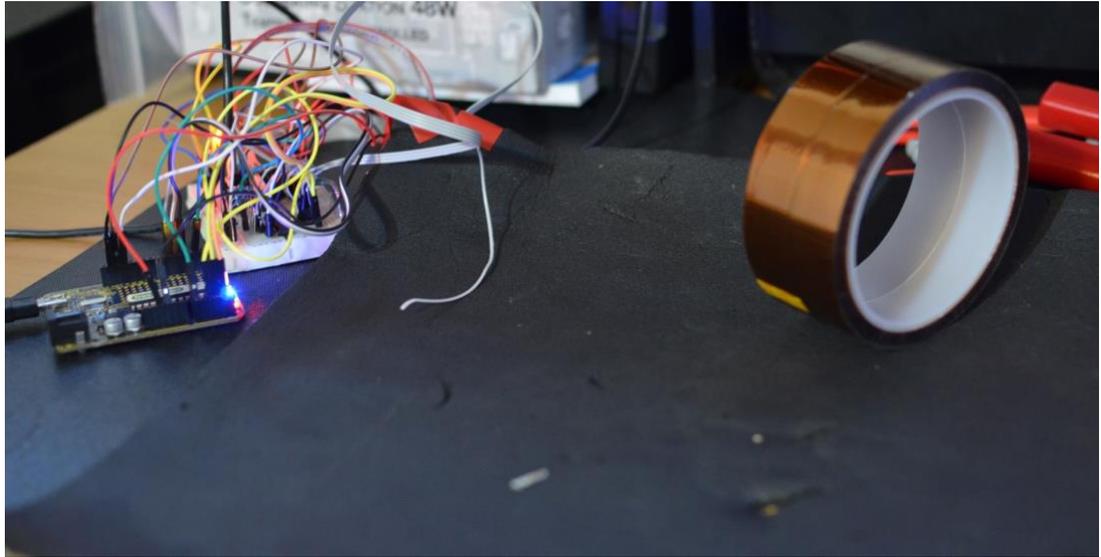
The initial investigation including the use of the conductive foam material to provide pressure mapping provided some good insights. While the foam had good pliability and therefore offered the possibility of high sensitivity, as a surface for the given use-case in this research it was less suitable. When attempting to move the chair while seated, the movement was hampered by the thickness and pliability of the foam. This was expected as office chair castors typically require a more solid surface for movement. A new material offering the same properties but allowing for potential movement would need to be sourced.

#### *4.2.2 ELECTRODE MATRIX AND GROUND SENSOR CONFIGURATION*

A small-scale implementation using the above material was created and assessed for its feasibility in sensing expected pressures given the use-case and is shown in Figure 5.

For quick prototyping purposes and to test the foam's pressure sensitivity, the surface used to mount the copper tape consisted of two materials that were available, an electrostatic discharge (ESD) mat used for electrical components on the bottom layer, and a thin butyl-based foam for the top layer with the copper tape connected by soldered wires to pins on an

Arduino Uno. Between the intersection of each copper row and column, a conductive foam piece was placed forming an electrode, shown in Figure 6.



---

**Figure 5:** Initial implementation, assessing the sensitivity of the sensor using the rolling motion of the tape

---



---

**Figure 6:** A single conductive foam piece, one of many placed between the copper electrodes

---

Shown in Figure 7, ten rows of copper on one surface and ten columns on the other were used to create a sensor with 100 individual electrodes in this feasibility test. 5V is provided to the matrix, and the prototype showed that the voltage delta was sufficient enough to provide information on the

varying force applied in different locations, and along with the microcontroller configuration, could theoretically read values between 0 and 1023 using a 10-bit Analog to Digital Converter (ADC).



---

**Figure 7:** Top layer and bottom layer with copper rows and columns exposed. Wires are soldered to each one and connected to the Arduino

---

A simple measurement of voltage using a pair of electrodes is used. Voltage is fed to the circuit through a voltage divider with a fixed resistor of 1k ohm and the carbon loaded foam acting as a variable resistor. After experimentation with various resistors on small scale test implementations, it was found that a 1k ohm value is suitable for the application and provides a single pressure reading across the electrode position. These resistors were used instead of the microcontroller's internal pullup resistors on the analogue pins to make experimentation with different resistor values easier to perform. Multiple electrodes were implemented using rows of 25mm wide copper tape approximately 5mm apart from each other on a separate surface above the foam, and a similar surface lined with copper columns orientated perpendicular to the above surface which is then placed underneath the foam so both copper surfaces are facing each other. Each of these copper column and rows is wired to a pin on a microcontroller. By selecting one pin to output to the corresponding column, the input pin on the microcontroller can read the values for each row consecutively. Once this is completed the

process can be repeated for the next column. This provides pressure sensitive areas at each intersection of the rows and columns, effectively creating a matrix of sensory nodes that is efficient in cost and construction time.

The butyl-based foam, used for its availability and pliability, became the top layer in contact with the user and chair. With the copper matrix, it proved effective in allowing spatial pressure to affect the foam pieces below.

However, it was less than durable when in contact with chair castors, as any pivot of the castors distorted the foam and resulting in a damaged surface that would not be reliably used over long durations, and the results are shown in Figure 8.



---

**Figure 8:** *Damaged exhibited on the top surface after pivoting of a chair castor*

---

Moreover, the ESD mat used as the bottom layer worked well in providing a mounting point for the copper tape but did not always stay fixed in place on the floor, allowing the bottom layer to shift from its initial orientation.

#### **4.2.3 DATA ACQUISITION AND PROCESSING**

Processing and data acquisition is done on a remote computer connected to the microcontroller via USB. As well as collecting the data, the main function of this processing phase is visualization of the data, making it

easier to see real-time pressure values and to diagnose any faults with the sensor, a preferable choice compared to viewing numeric tabular data. As it was on hand, an Arduino Uno was used as the microcontroller for this stage of the research.

Before the electrodes are polled for pressure values, a 2-dimensional array is instantiated on the microcontroller as virtual representations of the sensor and its electrodes. This array holds calibration information. This calibration occurs after initialisation of the microcontroller's pins and is implemented to account for non-zero readings and anomalies apparent in the electrodes when no pressure is applied. This is achieved by zeroing of the array first. Next, a value is obtained from each electrode once and stored in the calibration array. This is repeated 100 times, with each value being added to the existing value in the appropriate index. Once a full pass has been completed 100 times, the total value in each cell of the array is divided by 100 to assign a base value for each electrode. When pressure values are read, these base values are used in approximating actual pressure by subtracting the appropriate base value from the pressure reading.

Furthermore, the highest value obtained among all the electrodes during the calibration phase is used in implementation of a minimum value filter. As discovered during testing, the calculated base value is not always accurate in ascertaining a zero pressure reading over time. This was expected as the base value was calculated using an average of 100 passes of the sensor, meaning pressure values would frequently report miniscule fluctuations above or below the calculated base value when no pressure was being applied. To accommodate this factor the actual pressure value, calculated after the base value correction has been applied, is then assessed against the minimum value filter. If a value is lower than the highest reading obtained during calibration, the value is passed to the processing program as zero. A value that is higher remains unaltered. This reduced many of the insignificant pressure values and noise present in the pressure readings prior to the implementation of this filter.

A program on the remote computer waits for serial information supplied by the microcontroller. To initiate communication, the microcontroller sends a single character. When this character is seen by the computer, the program clears the serial buffer, and as it is the first data seen from the microcontroller, responds with another character. When this is received by the microcontroller, and communication between the two established, the microcontroller begins the calibration phase. After this phase, a single byte of data is sent. This single byte is the value obtained from a single electrode after accounting for the calibration result, or base value, in the microcontrollers array described in the preceding paragraph. This value is stored in an array on the remote computer along with any other existing values passed over serial. Since this program is unaware of the highest or lowest value polled by the microcontroller, it determines what these values are twice, one for the entire length of the program execution, and for one each pass of the sensor. This is done to determine if there are any erroneous readings that may be occurring after transferal of the byte. After each byte is received, another is then requested from the microcontroller, and this continues until the storage array on the remote computer is full.

Once this array is full, the program recognises that a full pass of the sensor has been completed. A visualisation phase, shown in Figure 9 and 10, takes the data from the array and arranges it in an easy to see form that is used to debug any errors. Because the occurrence of a full-read of the sensor is very fast (around 50ms), to a human viewer the visualisation always looks visible as each issuing of a screen refresh occurs. Values within the array are visualised by mapping each value onto separate rectangles aligned next to each other to form a virtual representation of the sensor. Some examples of errors detected were weak contact points with either the copper electrodes or incorrectly programmed pins. However, as the visualisation was created with a 3D effect to provide visual cues such as depth, it was harder to view information from all electrodes at the same time, as an electrode with a higher value would be in front of other electrodes values.

At this stage of the design search process, AR was not attempted as the pressure sensing performance and usability would need development before this could be realised.

#### *4.2.4 INITIAL ITERATION SUMMARY*

On completion of this phase, a mat is constructed consisting of a foam based top layer and an ESD mat as the bottom layer. These two layers are lined with 5 copper strips, orientated perpendicular and facing each other, with a piece of conductive foam placed between the intersections of each. The copper strips are each wired to an Arduino Uno, which is connected to a computer over USB. On this computer, a program stores each value supplied by the microcontroller. When the program has all values from each electrode, a graphical representation of the data is displayed on screen. This process repeats, displaying a graphical representation of each pass.

The following observations were made that worked well with the current implementation and would require little or no modification:

- Copper columns and rows provide good spatial recognition.
- Microcontroller and related programming worked well with the copper matrix to retrieve and store data.
- Visualisation worked well to show pressure readings, but the 3D effect applied to the visualisation could hide pertinent information.

There were many aspects that did not function as intended and would need modification:

- Conductive foam pieces inhibit movement. A different resistive material would need to be sourced.
- Butyl-based top layer deformed and unsuitable under chair castors. A more resilient material is needed which would still allowed for pressure to reach the sensor without ruining the material.
- Bottom layer of the mat would need to be replaced with somethings that provides a firm base for the sensor that stays fixed in place.
- Visualisation needs the ability to change settings, to allow for easier diagnosis in different situations.

The following iteration would need to implement a more adequate surface as the primary objective, as development of other features would be irrelevant without a more usable mat.

### 4.3 SECOND ITERATION

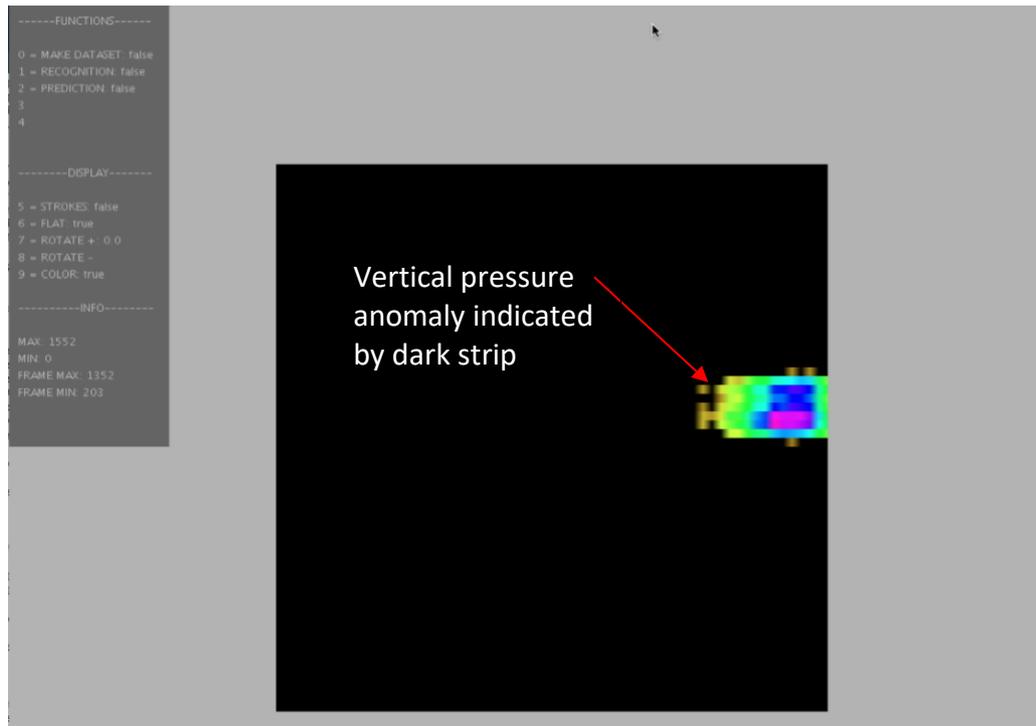
The initial exploration demonstrated areas for investigation in multiple aspects of the sensor. Addressing of these issues are discussed below.

#### *4.3.1 DATA PROCESSING*

To solve the problem of electrode values inhibiting the view of other electrodes' values in the visualisation, some GUI options were created. These allow toggling between a 3D view and 2D top down view of a plane, representative of the sensor itself. Also implemented is the ability to visually rotate the sensor to varying degrees between the 3D perspective and flat perspective.

Furthermore, an option to toggle between three-dimensional pressure visualisation (where higher-pressure values create taller peaks) and colour-gradient pressure-based pressure visualisation (where no height is applied to values, but variance is shown instead by colour indicators) was created. Also, the ability to toggle between showing defining lines between electrode spaces was added.

None of these are essential to the operation of the sensor as an AR interface to the IoT, however they are very useful in helping debug issues that may be occurring with the sensor or software. An example of this visualisation process is shown in Figure 9.



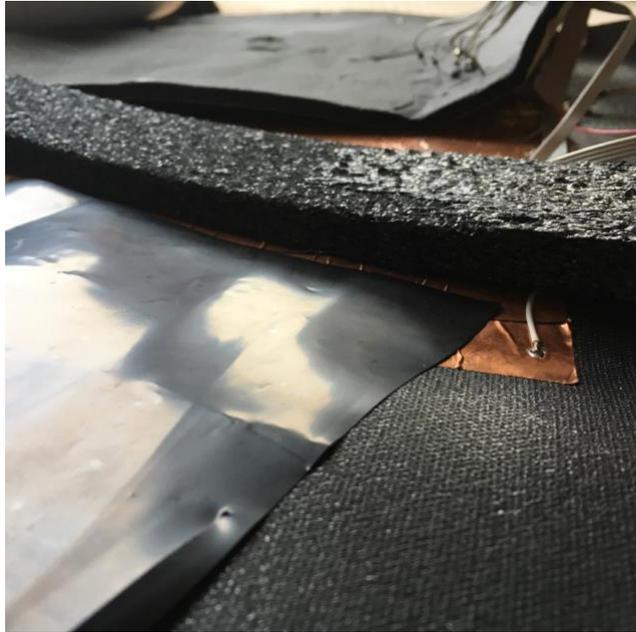
*Figure 9: Visualisation of a fist pressed onto the surface of the sensor, with indications of a malfunctioning copper contact within the system*

#### *4.3.2 SEMI-CONDUCTIVE POLYMER COMPOSITE FOR PIEZORESISTIVE PRESSURE SENSING*

Referring to literature again, an implementation by Cheng et al. (2016) produced a solution which altered their previous design (cf. J. Cheng et al., 2014) to include a different material with piezoresistive properties. One of the reasons given for this change in material use was similar to the finding of this research; thickness of the foam inhibiting movement and compactness. However, the new piezoresistive material they used in this later artefact was bespoke, manufactured to the researcher's specifications. That approach deviated from the intended research objectives of this research, as a material that is simple, relatively cheap, and be readily available would realise these objectives more clearly. However, the use of a more accessible thin layer of a carbon laden material was considered as a viable alternative for application in this research.

A widely available commercial material known as Velostat or Linqstat (dependant on manufacturer trademark) is used in this research. Typically

found in packaging of electric components for protection against electrostatic discharge, it is a volume-conductive, carbon loaded polyolefin. A comparison of material thickness between the previously used conductive foam and Velostat is shown in Figure 10.



---

**Figure 10:** Comparison of material thickness, with Velostat in the foreground and conductive foam behind it

---

Produced industrially allows it to be a cost effective solution for many applications. It is also very thin at a minimum height of 0.1mm, and available in various dimensions. This allows adherence to the invisibility parameters already discussed by being thin enough that its existence within the system is unnoticeable and would not impede activity or motion on the surface in comparison to materials with a larger volume. Volume resistance of Velostat is <500 ohms with surface resistance of <31,000 ohms/sq cm (Adafruit Industries, n.d.). As resistance through the material is less than across the surface, a single large sheet of the material can be used across multiple distinct electrodes with minimal voltage leak, an important aspect in simplifying the construction of the artefact as individual pieces between the electrodes, as used with the conductive foam, can be avoided. Due to the cost, minimal volume, wide availability, and piezoresistive performance, other researchers have used the material in various applications and

seemed suitable for the research objectives in this paper too (Giovannelli & Farella, 2016). It is important to note that the manufactures of Velostat did not intend it to be used in piezoresistive applications, and as such is not constructed to perform consistently in this regard. However, in the use-case for this research, and similar applications not requiring precise pressure measurement and where close approximations will suffice, Velostat use as a piezoresistive layer provides an effective solution. Across multiple electrodes too, this deficiency can also be negated by increasing spatial resolution.

#### *4.3.3 ELECTRODE MATRIX AND GROUND SENSOR CONFIGURATION*

The second iteration used the same materials for the electrode and matrix configuration as these worked well, but a change in the electrode housing material was needed to keep the sensor in place on the floor. The top layer also needed to be changed to be pliable enough to allow force to be applied without dispersing the pressure below, while being durable and firm enough as to not affect user movement underfoot or in a chair. Both the butyl-based foam and the ESD mat were replaced by 5mm thick rubber matting, with the copper tape acting as the electrodes attached directly to them, shown in Figure 11.



---

*Figure 11: Rubber mat used in replacement of top and bottom layers*

---

Usability testing demonstrated that they provided the required pliability to register spatial pressure and kept its position and orientation even under

movement. However, the tactile properties of the rubber used in the top layer felt unnatural and hindered movement, especially when pivoting on the matting or when the chair castors had to move. The bottom layer rubber secured the orientation of the sensor to the ground firmly, and worked on either hard-floor surfaces, carpet tiles, and soft-pile carpets. However, future large-scale implementations should be able to work on bare floor surfaces without finishing's too, although this will be discussed in a later chapter.

Simultaneously, the spacing between each row of copper tape was reduced to 3mm and the width of the copper tape reduced from 25mm to 15mm, as shown in Figure 12. This was done in an effort to increase fidelity needed in AR between actions with very subtle differences in pressure and as a result of using a sheet of Velostat as opposed to individual conductive foam pieces. Furthermore, the textured surface of the rubber mat did not seem to effect pressure measurements when compared to those used in the previous iteration.



---

**Figure 12:** Comparison of the width of copper columns and rows between the initial sensor (bottom) and the second iteration (top)

---

#### *4.3.4 SECOND ITERATION SUMMARY*

The main objectives of this iteration were to find a more suitable pressure sensing material and more appropriate materials for the top and bottom layers. These were achieved in the following ways:

- Velostat provides a good material for pressure sensing and does not inhibit movement of user or chair castors. Moreover, no alterations to other parts of the sensor are required, making it a direct replacement to the conductive foam.
- Bottom rubber layer provides a good fixed base underneath the user and supports the layers above it.
- Top rubber layer does not deform like the previous iteration.
- Options in visualisation allow for multiple points of view, and thus detection of anomalies visually becomes more efficient.

However, one aspect would need more investigation:

- Top rubber layer inhibits movement of the chairs castors and thus the user, due to the tackiness of the rubber.

While this issue is investigated in the next iteration, a simultaneous attempt to advance the sensor toward a more final implementation would be conducted in the iteration. This is predominantly the need to scale the sensor to a larger size and the mitigation of possible issues this presents.

### *4.4 THIRD ITERATION*

Considering that the bottom rubber layer and the use of the Velostat, along with the matrix configuration and related software performed well in previous iterations, these remain unchanged. However, changes may be inevitable when scaled to a larger size.

#### *4.4.1 ELECTRODE MATRIX AND GROUND SENSOR CONFIGURATION*

In investigation of the problems with the previous iteration, the top rubber layer was removed and replaced with a carpet tile layer with a thin bonded urethane backing typically used in commercial environments, with the copper tape placed onto this backing, shown in Figure 13. Usability was

vastly improved with ease of movement apparent both directly and indirectly through the office chair, able to move freely and feeling neutral underfoot. The carpet was pliable yet less viscous than the rubber. However over prolonged periods, the lighter weight of this layer compared to the rubber layer beforehand meant it was prone to alter its orientation slightly, affecting the position of the electrodes and therefore the clarity and reliability in values.



---

**Figure 13:** *Carpet and backing, replacing the top rubber layer*

---

To overcome this, Velcro strips of 10mm in width along the edges of all three layers were used to bind them together. More permanent fixings of the layers were decided against and the Velcro chosen to enable easier repairs or changes. However, having Velcro strips in place along the edges of the layers deemed these areas unusable for obtaining sensor data from and added to the height dimensions along these edges, resulting in a height of approximately 12mm along these edges compared to the uncompressed height of the sensor elsewhere of approximately 8mm. With more permanent implementations, an improved fixing method binding all layers

together would need to be considered, although a later section will discuss improvements for future large-scale deployments.

The physical dimensions of the sensor in this iteration were 1200mm by 1000mm at a thickness of between 8mm and 12mm. The 1.2m x 1m were used as it is the default size in which the carpet is available from most manufactures, and closely fits the observed space needed for a user working at a desk in front of their computer. The sensor with these dimensions is shown in Figure 14.

Since the initial physical implementation and copper matrix worked well, this configuration was kept but extended to fit the new dimensions. At 15mm width spaced at approximately 3mm apart, there are 64 rows and 58 columns for a total of 3968 individual nodes. Space was left between the layers for wires, soldering, and Velcro, meaning the copper tape fit into a space measuring 1150mm by 1000mm. While this creates a sensor density  $>1\text{sq cm}$ , more disperse than is present in many of the implementations discussed in the literature review, for the use-case and objectives of this research, the sensor density is fit for purpose.

Moreover, it is possible that cost and energy use can be reduced by having less fidelity, and if no adverse effects to AR performance are seen, optimized to produce a cheaper and energy efficient solution, important ideals within the IoT paradigm. However, optimal sensor density was not explored in this artefact, and instead was determined primarily by the space, dimensions, and material available.



---

**Figure 14:** Sensor placed underneath chair in use-case scenario. Numbers indicate the relative position of electrodes

---

#### 4.4.2 MICROCONTROLLER AND MULTIPLEXERS

While providing a rather rapid proof of concept, other issues were identified in scaling of the sensor. The use of an Arduino Uno resulted in two recognizable issues.

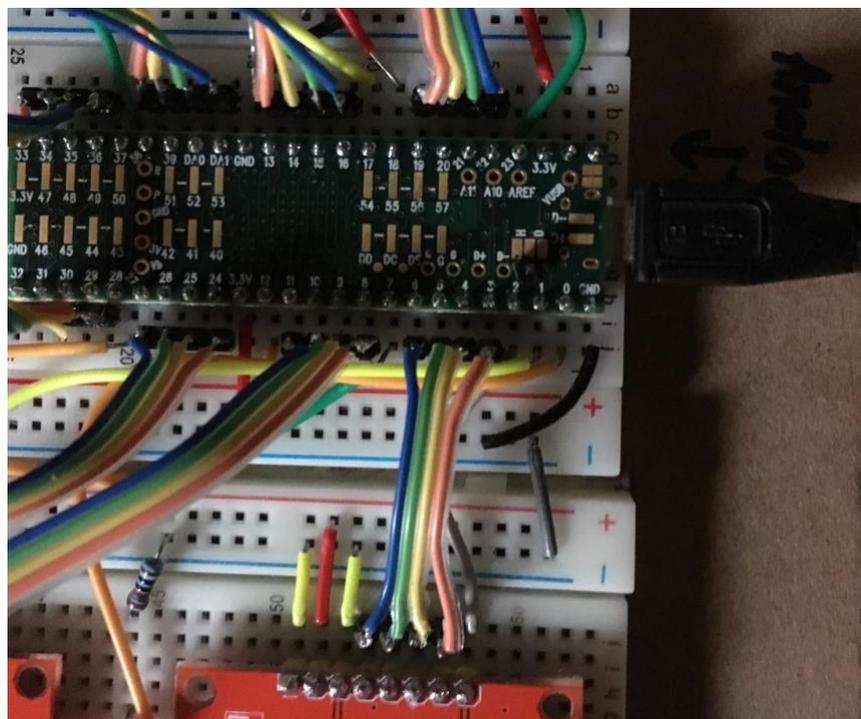
The issue with the larger surface area meant that more copper rows and columns would need to be implemented to keep the spatial resolution constant with the previous iterations. This introduces a problem with the Arduino Uno; the sample rate. With a clock speed of 16Mhz, the default sample rate of the Arduino Uno means a theoretical maximum sampling

rate of approximately 10,000 analogue reads per second, notwithstanding other operations the microcontroller may be performing (“Arduino Reference,” n.d.). This can be calculated by taking the 16 MHz clock speed and dividing by the default prescaler of 128 to give 125 KHz. With each analogue to digital conversion in the Arduino AVR taking 13 ADC clock cycles, 125Khz enables a sampling rate of 9615 Hz for analogue reads (“Arduino Reference,” n.d.). For a sensor with 3968 individual electrodes, this equates to less than three passes of the entire sensor matrix every second, with evaluation showing a full pass approximately every half a second. While slower sample rates and therefore clock speed would conserve energy use, an important factor in IoT applications, having the microcontroller at maximum performance (when considering 10-bit ADC with no modifications from the default settings) meant no possibility of increasing the sample rate if this was too low to provide reliable variance in pressure sensor values. Instead, a microcontroller capable of higher sample rates that could be slowed to conserve energy but could be operated at higher sample rates would be optimal, able to provide a contingency in case the need for an increased sample rate is apparent. Further to this, with a default 10-bit ADC, mapping of analogue reads to digital integers was limited to 1024 values at 5v input. To increase bit values would require microcontroller modification, possibly at the expense of noise. It was becoming apparent that the Arduino Uno, while capable enough during the feasibility testing and initial iterations, may struggle with the larger and more accurate requirements of AR in the research artefact.

The second obvious issue is the lack of pins available on an Arduino Uno which limited the rows and columns possible. By increasing the number of electrodes in the artefact, the number of pins would need to increase for a real-world implementation, and this was not possible using a single Arduino Uno. Eight CD74HC4067 16 channel analogue multiplexers were used to solve this issue. Four of these multiplexers were used for the top layer copper columns, and four used on the bottom layer copper rows. Each layer in this configuration could consist of a maximum 64 columns and 64 rows. However, while all eight multiplexers were used, not all channels on one of

the multiplexers were needed. In future implementations with a larger surface area or with higher electrode density, there would be no issue increasing the number of columns to use these channels or conversely increasing the number of multiplexers if required.

Due to these two factors, the Arduino Uno was replaced with a Teensy 3.6 shown in Figure 15. The Teensy was chosen over other suitable microcontrollers due to the number of available pins and ADC resolution (57 in total, with two 13-bit ADC multiplexed to 25 analogue inputs) and a much faster clock speed (180Mhz compared to the 16Mhz of the Arduino Uno) with the ability to vary the clock speed for testing. The CD74HC4067 multiplexers continued to be used in conjunction with the Teensy, and with four control pins and one signal pin used per multiplexer, the Teensy was easily able to accommodate all 40 pins needed to read values from the 3968 electrodes. Another reason the Teensy was chosen was because of its compatibility with the Arduino development environment and many of the libraries too via Teensyduino, allowing the reuse of code from the previous iterations using the Arduino Uno with only minor changes.

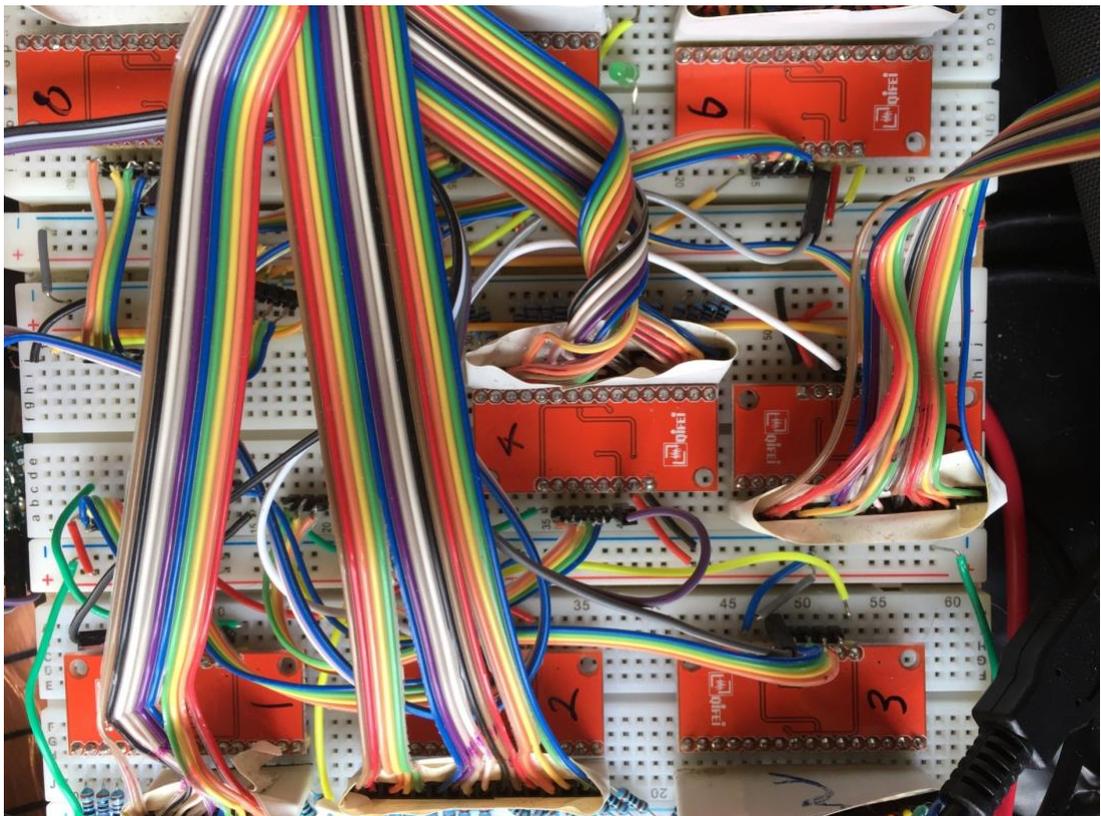


---

**Figure 15:** *Arduino Uno replaced by the Teensy 3.6*

---

The Teensy is connected to each of the CD74HC4067 multiplexers (on a breakout board for simplicity) in the following configuration. A single pin is used to enable or disable all channels, with four output pins connected to the multiplexers address select pins, enabling cycling through the various 16 channels, each channel connected to a single copper column or row. Sending a specific combination of bits (in either high or low) from the microcontroller to the address select pins chooses a specific channel to read a value from. Four of these multiplexers are used for the rows, and four used for the columns, allowing a maximum 64 x 64 matrix for 4096 electrodes. GND and VCC are wired to ground and 3V power source respectively. For simplicity, compactness, and explorative reasons, the power was supplied through the microcontroller instead of an external power source present in various other implementations. This arrangement is partially shown in Figure 16 without the microcontroller connected.



**Figure 16:** The set of multiplexers used for connection to the sensor rows and columns, with multiplexers 1 – 4 used for the top layer, and multiplexers 5 – 8 used for the bottom layer. The microcontroller is removed.

One channel of the 64 channels (on the first group of four multiplexers) is selected, while one channel is selected from the other group of multiplexers. A value is made available to an analogue input on the microcontroller via the multiplexers signal out. In this application, this value is sent to a computer via USB for storage and processing. The next channel is selected on the second group of multiplexers and the same process occurs. After each channel on the second group of multiplexers has been consecutively polled for values, the next channel on the first group of multiplexers is selected, and the process involving the second group of multiplexers is initiated again, allowing a full pass of the sensor.

When polling each electrode via the multiplexer, a write and read process occurs. First a write process is initiated with a channel parameter. Then all multiplexer channels associated with the columns are disabled. Depending on the channel (0 - 63) being written to (controlled by the main program loop) the appropriate multiplexer will be enabled along with its signal pin, while the others remain disabled. An array representative of the pins associated with the current multiplexer's address select pins are copied to a local array to be used in the write process. While the initial channel value passed for write indicates any one of the columns to write to, a modulo 16 operation on the channel value allows the correct channel for a particular multiplexer to be selected. For example, channel 39 (the 39<sup>th</sup> column), would equate to multiplexer seven, channel seven in this configuration. With the correct multiplexers address select pins available in the local array, a secondary array holding the binary value of the multiplexer's channels (0-15) are used in a write operation to set each of these address selection pins. This allows the voltage of the signal pin enabled earlier to pass through the necessary channel on the multiplexer. This remains static until a complete pass of one row finishes. A similar process occurs when polling the rows. After being passed a channel value, all multiplexers for the rows are disabled. The appropriate multiplexer is enabled, and the address pins for that multiplexer are selected, along with the multiplexers signal pin. The channel for that multiplexer is enabled using the same techniques as the

write method. However, an analogue value is now read from the selected multiplexers channel, which is then returned for further processing.

#### *4.4.3 THIRD ITERATION SUMMARY*

The third iteration resulted in the following findings:

- Carpet surface fastened with Velcro to the underlying layers provides good spatial pressure performance.
- Teensy replacement of Arduino Uno is suitable, providing more pins and faster sampling.
- Use of the multiplexers to extend number of electrodes extends functionality for larger dimensions.

There were also issues identified through the changes in this iteration:

- With the same amount of force exerted on the sensor, pressure values with the Teensy appear lower than those observed when using the Arduino.

This issue is addressed in the next iteration. Furthermore, processing has been limited to data visualisation only while development of the physical sensor and related software are configured, with visualisation used for debugging purposes. For the final implementation, data pre-processing must occur in readiness for the machine learning algorithms, as these will be needed in AR.

#### *4.5 FOURTH ITERATION*

To address the first issue, it was assumed that the change in microcontroller caused the drop in pressure readings. However, this could have been due to a number of factors, such as the extra load placed on the power supply with the addition of the multiplexers and extra electrodes, or the effect of this load on the ADC reference voltage. To negate these affects, a more reliable and direct supply of power to the sensor could be provided, or the use of an external ADC, rather than relying on the microcontroller only. While the actual cause of the lower values was not established, the simplest solution which still enabled the sensor to be powered solely by the microcontroller, was to make use of the higher bit ADC available in the Teensy, allowing for

greater sensitivity and therefore produce similar pressure values as those seen using the Arduino. This option was chosen to solve this issue.

Also addressed in this iteration is creation of a pre-processing mode to ensure the data is ready for AR, while still allowing for visualisation of the data in-case of further refinements.

#### *4.5.1 DATA ACQUISITION*

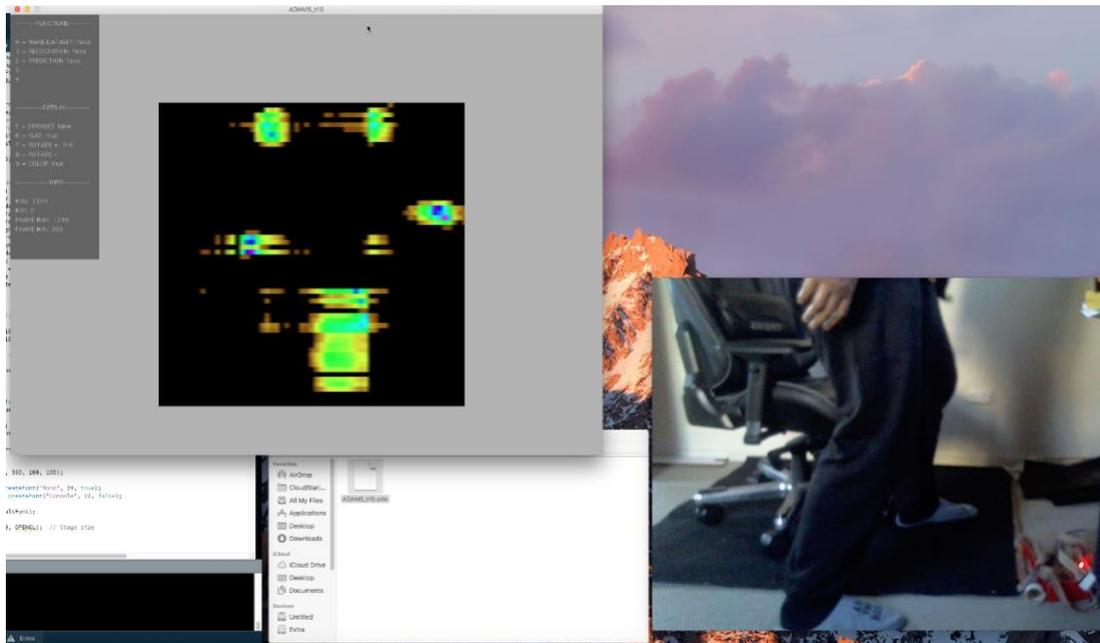
To allow for more variance between electrode values, and to negate the drop in analogue variance due to the decrease of supplied voltage, increasing the number of bits used with the conversion of analogue to digital values was implemented. Rather than using external ADC's, the Teensy's inbuilt components were used. Solely using the Teensy's inbuilt ADC's instead of separate ADC's like other implementations in the literature review, as well as using the voltage supplied through the microcontroller itself, allows for a more compact and cost efficient architecture. These ADC's within the microcontroller are configured to use 12-bit resolution, performing better than the 10-bit ADC on the Arduino Uno by allowing more variance between pressure values. While higher bit resolution greater than 12-bit is possible on the Teensy and would theoretically allow for more clarity of the pressure readings, the internal ADC becomes more susceptible to outside noise and interference, especially with higher sample rates. This was evident when higher bit resolutions were tried and did not provide more obvious or necessary clarity than 12-bit once this was accounted for.

Changing to 12-bit values presents some problems. One problem is that the serial bus is only capable of sending a single byte at a time, and the remote computer expected only one byte as an indicator of a pressure value. Thus, since any 12-bit value is greater than a single byte, the value from the electrode is split into a hi-byte and lo-byte, and sent separately over serial. The computer waiting for data recognises the first byte, waits for the second byte, and when received reassembles this byte into a single value which is consequently stored in the array of values. Furthermore, the visualisation component needed to be altered to account for higher pressure values.

#### 4.5.2 DATA PROCESSING

While wanting to keep the data visualisation aspect of the program usable, the need for pre-processing of the data into appropriate forms for machine learning was also considered in this iteration. When the processing program continues to run, eventually the array holding all values read from serial will become full, as was present in the previous iterations, indicating a full-pass of the sensor.

Now, instead of simply creating the GUI and the related visualisation options, other options become available. The GUI enables selection of three different modes to operate in. The default mode, used in the previous iterations, is used for rapid functionality testing and ensuring the various parts are operating as expected by simply enabling data acquisition via serial and displaying this data with the above options. These options can be seen in Figure 17.



**Figure 17:** Example of pressure map produced by the sensor, with further options to initiate different modes in the GUI

The second mode is a staged mode. In this mode, the user is given a visual alert that a training period will begin within a period of time, and serial reading is halted. The visual alert will also tell the user which type of activity it wishes to record next out of the eight predetermined activities.

When the countdown has finished, and with the user performing the defined activity, the program will start reading input from serial and processing it as necessary following the procedure above. After a number of samples (changeable in the GUI) have been read and processed, the program will once again pause and display another activity for the user to begin doing. After another countdown allowing the person to move from one activity to another, the program will begin to read the next data from serial. This process continues until all eight predetermined activities have the same number of samples each. Implemented this way, data size can increase or decrease evenly across all activities.

Another important aspect of this GUI is the ability to choose between data structures for post-processing of the pressure values. There are two file types that can be chosen to suite various machine-learning preferences. These are .arff (used for software such as WEKA) and .csv (a file format across many different programs). Depending on the file type selected, the program will create a file and write to that file any necessary header or meta information. At each full pass of the sensor and once the program has an array full of values, the array values are output to a file in the appropriate format. When first initiating either a staged mode or free mode, the file is created and the filename is timestamped. Any data collection from the sensor will be written to this file, until the collection is stopped. If data collection is started again, a new file will be created with the new timestamp, and data written to this file. Modularising this function allows other file types and corresponding analysis tools to be implemented more rapidly.

To assess whether the data files were in the correct format for WEKA, a J48 Decision tree was used on a selection of files created by the sensor. After a few fixes (such as correcting the header information of the .arff files) WEKA was able to analyse the files, and produce a model that could be used for AR. Initial testing done with Zero-R produced some expected results, and further tests with decision trees produced results with higher than expected accuracy, which are discussed in section 5.

### 4.5.3 *FOURTH ITERATION SUMMARY*

The iterations main alterations were to use the ADC's of the Teensy, and to allow for the data to be stored and analysed by machine learning algorithms. This involved pre-processing of the data and creating a method for labelling each pass of the sensor. The following highlight the results:

- 12-bit ADC provides higher variation between electrode values without the need for external components.
- Passing two-byte values over serial and altering the software on the remote computer to reassemble two-byte values.
- GUI option to enable writing of data to a file while keeping visualisation options available.
- Creating a method for labelling the data when writing to a file.

During the testing of this mat, it became apparent that the current labelling system was inappropriate in realising the research objective. While it was very useful in determining if the write function operated correctly, the number of instances between activities is identical and could potentially skew the results of algorithms that account for the occurrence of instances in creating a model. Thus, further development is needed that addresses the following issue:

- Create an additional mode for labelling activity data that does not use a pre-determined number of instances but can still successfully label data irrespective of duration of time spent in a certain activity, or in transition from any activity to any other.

This should be possible with slight variation of the existing labelling function.

## 4.6 *FINAL ITERATION*

The objective of the final iteration is to create a mode that allows for unrestricted recording of data and subsequent labelling. The instigating factor is that of realism, as the data recorded will be of a user performing tasks as they would normally, with no prompting from the sensor.

#### *4.6.1 DATA PROCESSING*

Beside the default visualization and staged modes of the previous iteration, a free mode is also created. Unlike the staged mode, there is no predetermined number of samples collected per activity. Instead, all data is captured in one continuum. Because the preference is to work with labelled data as well as label the data in real-time to limit post-processing work, labelling is done by key presses of 1 to 7. This can be achieved by the user interacting with the sensor and pressing the corresponding key while switching activities, or by a second person with their own keyboard watching the user interact with the sensor and pressing the number corresponding to the activity the person changes to. There is no limit to the length of time recording this data and the user is able to end data capture via the GUI. Unlike the modes in the previous iteration, there is a possibility to investigate temporal analysis given the capture of time and frequency along with the pressure data in this mode, and although this is not explored in this research, may benefit algorithms which are able to formulate models based on variation between instance occurrences.

#### *4.6.2 FINAL ITERATION SUMMARY*

At the end of this iteration, the sensor is performing as envisioned. Rudimentary exploration of AR using machine learning algorithms also shows the ability to recognise activities from the sensor supplied data. As per the guidelines of DSR, iterative artefact development can be terminated after any number of iterations (Hevner et al., 2004). At this stage, no further development of the artefact is required. More in-depth evaluation will inform any further changes to the artefact, although pre-evaluation AR testing indicate that this will not be necessary.

## 5 ACTIVITY RECOGNITION EVALUATION VIA MACHINE LEARNING CLASSIFICATION

The output data files discussed in the previous section are analysed using the machine learning tool Weka. Weka is used for its simplicity and efficiency in automating statistical analysis with various algorithms, which is important in this research as a range of algorithms are used to evaluate the performance of the sensor as an AR interface. The probability classification algorithms used within this research will be briefly discussed, followed by the evaluation results for given algorithms, the sample size, and selected processing mode. As all data captured by the sensor is labelled in the process, supervised classification models will be used to assess the sensor as an AR interface. As a sensor and AR interface, linearity, stability, hysteresis, homogeneity, and repeatability are all aspects that must be addressed in its evaluation, and machine learning will allow analysis of some of these factors and will be discussed later. Furthermore, the models and any inferences created can be included as part of the AR system itself depending on the accuracy of prediction.

### 5.1 DATA ANALYSIS PRE-PROCESSING

There are two categories of data collected from the sensor; staged mode data, with equal occurrence of instances for each activity; free mode data, with variance between the number of occurrences between activities, mimicking real-world scenarios e.g. where a user may be using the mouse more than relaxing over a period of time. Also, instance numbers between datasets vary while the feature vectors for these instances are constant and equal the number of electrodes in the sensor. Analysis of the datasets is initially undertaken in WEKA to select appropriate algorithms for classification. Many different datasets were created (>100) and from these, seven were chosen as being representative of the majority of the datasets, based on capture mode, instances, and file size.

The use of multiple datasets with various algorithms is to ensure completeness in evaluation. The variation between instance totals was used

to evaluate algorithm accuracy under differing scenarios. This is important as part of an AR sensor in real-world scenarios. For example, if computation for AR classification in a real-world implementation is done via an embedded solution, energy, computational and storage limitations dominate performance evaluation, effecting such things as model complexity, data storage availability, and classification latency. However, if classification modelling and classification itself are performed remotely by more powerful computing systems, then data transferal latency becomes a bigger issue. Therefore, reducing dataset file size seemed a necessary step to optimization. As seen in Table 5, dataset four has reduced in file size compared to dataset three, despite containing more instances. Dataset five, six, and seven continue to use this file size optimization which is achieved by simply storing and sending a zero-pressure reading more efficiently and, given that a zero-pressure value is the most common value encountered, provided an easy way to implement file size reduction.

Since the machine learning algorithm for AR in this artefact is decoupled from the sensor itself and does not assume suitability of any particular algorithm (apart from those suitable for classification), different scenarios can be evaluated to predict the best routes for future research given both lab and real-world implementations.

*Table 5: Overview of the seven datasets used in the machine learning evaluation of AR. Datasets are in the ARFF file format.*

Dataset	Mode	Instances	File Size (MB)
1	Staged	400	8.1
2	Staged	1200	23.9
3	Staged	2800	55.7
4	Staged	4000	37.4
5	Free	3966	32.9
6	Free	5516	45.9
7	Free	17096	153

While not an exhaustive analysis of the data or suitable algorithms, there are some general aspects of the data to note. The first aspect that is apparent is the sparse vector representation, with many of the nodes always 0, indicating they never have any pressure on them at all. This means, when combined with the high feature space, any chosen algorithm would benefit from feature selection or should use dimensionality reduction to improve accuracy if manual data pre-processing is to be avoided. In this regard, a decision tree or a related ensemble approach may be useful. Furthermore, in the staged mode data sets there are obvious clusters formed within data points of the same class. Therefore, since the data is linearly separable, a linear classifier such as Naïve Bayes may provide good accuracy. However, this same attribute is not so well-defined in the recording mode, as transitions between activities cause more variation in vector values. Therefore, a Support Vector Machine capable of handling non-linearity in the data may be more useful in this regard. On the basis of this evaluation, four approaches are considered as candidates for further investigation.

### *5.1.1 J48 DECISION TREES*

A statistical classifier based on the ID3 and further C4.5 algorithms developed by Ross Quinlan, J48 is an open source Java implementation (Quinlan, Morris, Jackson, & O'Connell, 1993). Simply put, a decision tree is created top-down, with a training dataset split into subsets that contain instances of similar values. Splitting the dataset on attributes to create subsets that are the most homogeneous and decrease entropy provide what is known as information gain. The attributes with the highest information gain are chosen as the attributes to split the data on. When a branch has an entropy of 0, this leaf node signals that no further splitting is needed along this branch. However, any entropy value other than 0 signals uncertainty of target values, and another split of the subset will occur and the process of splitting continued. This process creates the decision tree model. New data can be provided to this model and it will predict the appropriate classification of each instance.

The J48 algorithm has been used extensively in a wide range of research applications, including analysing e-governance data (Rajput, Aharwal, Dubey, Saxena, & Raghuvanshi, 2011), mining software repositories (Finlay, Connor, & Pears, 2011), predicting fish stocks (Zarkami, 2011) and fault analysis (Muralidharan & Sugumaran, 2013) to name but a few. In general, decision trees are considered simple and fast to implement, given that they formulate the most important features of the data automatically during the search for highest information gain. However, they can also be problematic in overfitting, where models fit the training data so precisely that new data highlights any inflexibility in the model. This overfitting is exaggerated with the tree's complexity. As each dataset in this research contains over 3000 different attributes (each electrode) for each instance (with some datasets having over 10,000 instances), tree depth is likely to be large, increasing the chances of overfitting. This could present itself as an issue in realising a more general model applicable across unseen datasets, which would be the case in real-world implementations of an AR sensor. Furthermore, assuming the decision tree uses a greedy approach to choosing optimal nodes (Hunt's algorithm), optimal local choices are guaranteed. Ignorance of the rest of the tree, however, means that the local optimum may not be the best choice globally, and can lead to suboptimum decision trees, and in some cases the worst possible solution in relation to tree depth, requiring techniques such as tree pruning to resolve these issues (Norouzi, Collins, Johnson, Fleet, & Kohli, 2015). However, as an indicator of the sensors appropriateness in relation to linearity, drift, hysteresis, homogeneity, and repeatability, testing with a decision tree should allow some of these important elements of any sensor to be explored and will be discussed later (Giovanelli & Farella, 2016).

### *5.1.2 RANDOM FOREST*

Random forests is an ensemble machine learning method using decision trees, first implemented by Tin Kam Ho and extended by Leo Breiman and Adele Cutler (Breiman, 2001). Random forests, as the name suggests, uses multiple decision trees created from a differing random subsample of the

training data. Classification of a new data point happens independently on each tree, with each tree predicting the appropriate class. Assuming a majority vote, the class predicted by the majority of trees for the data point is chosen as the prediction.

Further, while there are implementations of decision tree models that grow with the introduction of new data, online random forests enable easier inclusion of new data into the model by having the ability to generate new decision trees with subsets of training data that include these new data points (Cassidy & Deviney, 2014). Decision trees that are underperforming and classifying consistently far from the majority could also be dropped. These aspects strengthen the ability for the classifier to evolve and would be an important aspect in long-term AR installations. However, creating multiple decision tree instances obviously adds to the compute time required for model creation.

Beside this, the core benefit of using Random Forest classification instead of a single decision tree is to prevent the overfitting problem mentioned above, creating a more general model and increasing accuracy in unseen data (Breiman, 2001). The Random Forest approach has been applied to a range of application areas, including gene selection (Díaz-Uriarte & De Andres, 2006), remote sensing (Pal, 2005) and land cover classification (Rodríguez-Galiano, Ghimire, Rogan, Chica-Olmo, & Rigol-Sanchez, 2012).

For the data in this research, this approach will aid in realisation of a more general model that may help in real world applications of an AR sensor and is tested to show any improvement against single decision tree models.

### *5.1.3 NAÏVE BAYES*

Naïve Bayes, based on the Bayes' theorem named after Thomas Bayes, is a probabilistic classifier used in machine learning. Because it is a class conditional independent algorithm, it assumes that the occurrence of a feature is independent of the occurrences of other features and does not consider any correlation between features, therefore naive. (Lewis, 1998). This is done by calculating class probabilities and conditional probabilities,

or the frequency of each feature value for a given class value, divided by the frequency of instances with that class as the value. However, the feature values in this research are numerical and thus continuous, where traditional Naïve Bayes algorithms expects categorical values. Because WEKA is being used to implement these algorithms, it is important to note that a Gaussian distribution is assumed for numerical attributes by default. Otherwise, converting numerical attributes to nominal attributes can be achieved via supervised discretization, among others.

Implementing this algorithm to allow for continuous data has its disadvantages though. For example, when using supervised discretization, information from the data can be lost as values are “binned”. This same issue is also apparent in decision trees too. Furthermore, the assumption that the data follows a Gaussian distribution may be erroneous, however there is allowance for non-Gaussian distribution among features values using a kernel estimator among others. Knowledge of the data distribution is key in ensuring appropriate use of the Naïve Bayes algorithm. The Naïve Bayes algorithm is also widely utilised, with many applications including heart disease prediction (Palaniappan & Awang, 2008), text classification (Kim, Han, Rim, & Myaeng, 2006) and location prediction (Anagnostopoulos, Anagnostopoulos, Hadjiefthymiades, Kyriakakos, & Kalousis, 2009).

In relation to the data in this research, as each class is represented at least once (at least in staged mode), evaluation will not suffer from the zero-occurrence issues sometimes apparent with Naïve Bayes. Because the staged mode data points for any given class have little deviation from the mean, Naïve Bayes should perform well. However, the free mode data having higher deviation among data points of the same class, may perform poorly. However, given the IoT and invisibility paradigm, a Naïve Bayes approach may be suitable in this application because it is able to converge faster than other algorithm, meaning training data size can be smaller than other algorithms and model creation is rapid. It also tends to have an advantage over other algorithms when the number of classes is large, and

while the datasets here only have a small number of classes, a real-world implementation where many different activities are possible would need to accommodate for this.

#### *5.1.4 SUPPORT VECTOR MACHINE*

An algorithm that may prove useful in evaluation of AR capabilities of the sensor, as well as its difference in approach from the other algorithms mentioned, is Support Vector Machine (SVM). Unlike the probabilistic method of Naïve Bayes, Support Vector Machine is a non-probabilistic linear classification algorithm that uses regression to form boundaries between data points. This boundary indicates the separation between one class and another. SVM creates this boundary by simply selecting a few of the data points (support vectors) for defining boundaries of classes (or hyperplanes in higher dimensions) and proceeds to find the boundary that fits with the highest margin between the points of different class.

Traditionally, this is a linear boundary. However, kernel manipulation can create boundaries (or hyperplanes) that are non-linear, and able to fit more complex data with high-dimensionality making it very versatile, although choosing the appropriate kernel function is not always clear. This flexibility also bodes well for datasets that cannot be linearly separated, which is typically evident among the real-world datasets. There are other benefits too. Unlike Naïve Bayes which makes distribution assumptions, SVM is likely to be beneficial when data does not follow a Gaussian distribution.

Unlike decision trees, SVM is resilient to overfitting because the dependence is only on the support vectors to calculate the boundaries rather than every data point. This also entails efficient memory usage, especially in larger datasets. It does have its disadvantages though. As well as needing an appropriate kernel function choice, large data sets can mean training of the model takes longer than other algorithms.

As with other ML algorithms, SVM is widely used in a diverse set of application areas that include, but are not limited to, face detection (Osuna, Freund, & Girosit, 1997), fault diagnosis (Widodo & Yang, 2007) and cooling load prediction (Q. Li, Meng, Cai, Yoshino, & Mochida, 2009).

It should be noted that the default mechanism for implementing SVM in WEKA is via John Platt's Sequential Minimal Optimization algorithm (SMO). The following therefore applies: replaces missing values; nominal attributes are transformed into binary attributes; attributes are normalized; multi-class datasets are classified using pairwise classification.

## 5.2 ALGORITHM COMPARISON AND EVALUATION

This section will detail the results of executing the machine learning algorithms given the specific dataset. Again, this is not a comprehensive evaluation of all algorithms, but rather an insight into suitable algorithms for AR given specific types of data with respect to IoT applications. WEKA allows an efficient way of comparing algorithms, and unless otherwise stated, the default settings of WEKA in this regard are used.

### 5.2.1 ACCURACY

Each of the staged and free datasets will be evaluated using a randomized training set and test set split of 60% and 40% respectively. The criteria that will be assessed are accuracy and correct classification, elapsed training time, and elapsed testing time. Each evaluation is completed ten times per algorithm per dataset.

For the following tables (6 – 25), the columns from left to right are (1) J48 decision tree; (2) random forest; (3) Naïve Bayes; (4) Support Vector Machine (Sequential Minimal Optimization). The first four rows relate to the staged mode datasets (1 – 4), with free mode datasets visible in the last three rows (5 - 7), unless otherwise stated.

All algorithms performed well where frequency of instances across the classes were identical (staged mode), with no significance difference in correct classifications among the smaller datasets. The results shown in Table 6 indicate that J48 decision trees perform universally well for any given dataset, able to correctly classify between 99.88% and 94.81% irrespective of instance number or instance frequency.

**Table 6:** Percentage correctly classified with a significance level of 0.001 (two-tailed confidence level of 99.9%)

Dataset	Decision Tree	Random Forest	Naïve Bayes	SVM
1	99.88	100.00	100.00	100.00
2	99.79	100.00	100.00	100.00
3	99.78	100.00	99.67	100.00
4	99.18	99.90	97.13	99.99
5	95.23	98.04	54.47	97.88
6	95.68	98.11	53.62	97.50
7	94.81	97.12	35.30	97.14

This result was achieved using pruning with subtreeRaising and a two-tailed confidenceFactor of 0.001. This reduces tree depth and at lower instance numbers, doesn't seem to have an effect. As can be seen in Table 7, there is no difference between a pruned and unpruned tree for this dataset.

**Table 7:** Staged dataset with 2800 instances and 60/40 split show identical tree structure irrespective of pruning.

Pruning	Root Node	Tree Size	Leaves	Correctly classified (%)
none	1144	15	8	99.75
subtreeRaising	1144	15	8	99.75

When evaluating the largest dataset however, the J48 algorithm using pruning is a little more efficient, with the size of the tree slightly smaller and with less leaves, but only with an insignificant increase in classification accuracy, as shown in Table 8. While the difference is minuscule, this could be useful in creating a model that is less complex and therefore is better at avoiding overfitting and thus more useful for generalisation. Interestingly, most of the tree structure is relatively similar and is not very balanced, regardless of the pruning occurring or not.

**Table 8:** Free dataset with 17096 instances and 60/40 split show the effect of pruning on more complex data

Pruning	Root Node	Tree Size	Leaves	Correctly classified (%)
<b>none</b>	544	631	316	94.0333
<b>subtreeRaising</b>	544	575	288	94.1065

However, to confirm that the accuracy remains similar while reducing tree complexity and to see if any difference can be determined as a cause of the dataset itself, a holdout method consisting of a randomized separate training and test data set from dataset 4 is used, with the results shown in Table 9.

**Table 9:** Staged dataset with 4000 (2400 training and 1600 test) instances and using a holdout method of 60/40 to show the effect of pruning different data

Pruning	Root Node	Tree Size	Leaves	Correctly classified (%)
<b>none</b>	34	23	12	98.875
<b>subtreeRaising</b>	34	19	10	99.125

The results of this test with a staged dataset of 2400 instances and 1600 test instances using the holdout method confirm earlier results, with accuracy not significantly changing and the tree marginally reducing in size.

To reduce the tree size further and test the difference between post-pruning and online pruning, a smaller confidence factor of 0.001 (from the default 0.25) was used, along with an increase in the minimum number of instance per leaf to 5 (from the default of 2). All trees were using subtreeRaising pruning. The effect of inducing more aggressive pruning to achieve a smaller tree is seen in Table 10.

These evaluations with the free mode dataset reveals that varying the minimum instance values while keeping the confidence factor static resulted in accuracy decline. This was also reflected when the confidence factor was

altered, and overall smaller trees created fewer correctly classified instances. Therefore, the initial J48 decision tree with Weka’s default settings resulted in a relatively optimized model for this data.

**Table 10:** Free dataset with 5516 (randomized exclusive 3309 training and 1985 test) instances and using a holdout method of 60/40 to show the effect of different values in conjunction with pruning

Confidence Factor	Minimum Instances per leaf	Root Node	Tree Size	Leaves	Correctly classified (%)
<b>0.25</b>	2	814	117	59	99.7985
<b>0.25</b>	5	814	93	47	99.2947
<b>0.25</b>	10	814	79	40	98.136
<b>0.25</b>	20	814	45	23	95.1637
<b>0.001</b>	2	814	103	52	99.3955
<b>0.001</b>	5	814	67	34	97.3804
<b>0.001</b>	10	814	55	28	96.1713
<b>0.001</b>	20	814	37	19	94.3577

Referring to the initial comparison in Table 6 which shows there is no significant difference between algorithms on lower instance datasets, there is a significant difference in correct classification among the free mode datasets consisting of more instances. Both Random Forest and SVM outperform J48 with this type of dataset. However, there is no significant difference between the Random Forest and SVM in accuracy. Again, using the holdout training and test dataset from before, the Random Forest algorithm was able to achieve 100% correct classification accuracy, and SVM able to achieve an accuracy of 99.49%, as noted in Table 11. No further optimizations of the algorithm in accuracy of classification are needed, as these two algorithms are performing classification well. This can be seen in the low Mean absolute error (MAE) and similarly the low Root mean squared error (RMSE).

**Table 11:** Free dataset with 5516 (randomized exclusive 3309 training and 1985 test) instances and using a holdout method of 60/40 to show comparing RandomForest and SVM in WEKA's default configuration

Classification algorithm	Kappa	MAE	RMSE	Correctly classified (%)
RandomForest	1	0.0858	0.0227	100
SVM	0.9915	0.1875	0.2913	99.462

Because this is a multi-class classification problem and class balance needs to be taken into account, the Kappa metric reflects very good performance in most cases against random classification. Here, almost all algorithms have very high kappa values  $> 0.93$  regardless of the dataset.

**Table 12:** Kappa statistic for each algorithm across all datasets when using a two-tailed significance of 0.001 and 10-fold cross-validation

Dataset	Decision Tree	Random Forest	Naïve Bayes	SVM
1	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00
4	0.99	1.00	0.97	1.00
5	0.94	0.97	0.45	0.97
6	0.93	0.97	0.38	0.96
7	0.93	0.96	0.24	0.96

However, this is not true for Naïve Bayes and the free mode datasets. Interestingly with the free mode datasets, Naïve Bayes kappa values are 0.45, 0.38, 0.24 respectively as instance numbers increase. Accuracy drops dramatically to below 54% for the free mode dataset, despite having less instances than the largest staged mode dataset where it correctly classified 97.13% instances. This makes Naïve Bayes the worst performing algorithm in real-world cases. Furthermore, it begins to incorrectly classify at a faster rate than the others with a lower Kappa value across four of the datasets. These values are probably due to the Gaussian distribution assumption used as the default in WEKA for continuous features.

Further investigation using supervised discretization shows that Naïve Bayes is able to correctly classify significantly better, achieving almost 79.57% for correct classification of dataset 5 and 75.60% for dataset 7, a dramatic improvement from the Gaussian implementation of 54.66% and 35.90% respectively, as shown in Table 13 and Table 14.

**Table 13:** Comparison of Gaussian and Supervised Discretization Naïve Bayes with dataset 5

Distribution	Kappa	MAE	RMSE	Correctly classified (%)
Gaussian	0.4506	0.1132	0.3364	54.6658
Supervised Discretization	0.7476	0.0512	0.2255	79.57

**Table 14:** Comparison of Gaussian and Supervised Discretization Naïve Bayes with dataset 7

Distribution	Kappa	MAE	RMSE	Correctly classified (%)
Gaussian	0.2362	0.1615	0.4019	35.905
Supervised Discretization	0.6879	0.061	0.2466	75.6069

Observing the confusion matrix in Table 15 shows that despite the improvement, Naïve Bayes does still not perform as well as the others in classification. For a few classes there is a high recall value but with low precision, indicating that it may be identifying the majority of true positive cases correctly but also including false positive results too as it assumes a wide distribution of data points.

**Table 15:** Confusion matrix for Supervised Discretization Naïve Bayes using dataset 7

Classification	a	b	c	d	e	f	g	h
<b>a = neutral</b>	1407	22	117	144	174	0	44	0
<b>b = relaxed</b>	42	1088	93	28	43	70	7	0
<b>c = typing</b>	0	0	906	127	0	0	0	0
<b>d = mouse</b>	220	7	444	1445	51	14	0	0
<b>e = left</b>	1	2	2	2	81	0	0	0
<b>f = right</b>	0	0	0	0	0	35	0	0
<b>g = standing</b>	0	0	0	0	0	0	69	11
<b>h = away</b>	0	0	0	0	0	0	2	139

Interestingly, when comparing the rates of false negatives in Table 16 with the rates of false positives results in Table 17 across all datasets and accounting for class occurrence, the metrics for Naïve Bayes show higher rates of false negatives, indicating the models were more likely to classify an instance as not belonging to a particular activity, when in reality it was. However, for the other algorithms, particularly Random Forest and SVM, the models had almost similar rates of classifying instances as belonging to an activity when they did not.

**Table 16:** False negative rates across all datasets and algorithms using a significance factor of 0.001(two tailed) 10-fold cross validation

Dataset	Decision Tree	Random Forest	Naïve Bayes	SVM
<b>1</b>	0.00	0.00	0.00	0.00
<b>2</b>	0.01	0.00	0.00	0.00
<b>3</b>	0.00	0.00	0.01	0.00
<b>4</b>	0.00	0.00	0.00	0.00
<b>5</b>	0.06	0.02	0.37	0.02
<b>6</b>	0.03	0.01	0.30	0.02
<b>7</b>	0.03	0.01	0.25	0.01

**Table 17:** False positive rates across all datasets and algorithms using a significance factor of 0.001(two tailed) 10-fold cross validation

Dataset	Decision Tree	Random Forest	Naïve Bayes	SVM
1	0.00	0.00	0.00	0.00
2	0.01	0.00	0.00	0.00
3	0.00	0.00	0.01	0.00
4	0.00	0.00	0.00	0.00
5	0.02	0.01	0.04	0.01
6	0.04	0.02	0.10	0.02
7	0.01	0.01	0.05	0.01

When implemented, the use of a kernel estimator did not yield results as accurate as supervised discretization for the Naïve Bayes algorithm.

With Naïve Bayes improvement implemented, another analysis is performed and compared against each other, this time using ZeroR as a base case to compare the algorithms against the simplest classification possible beside randomly assigning classes. These results are shown in Table 18.

**Table 18:** Percentage correctly classified for all algorithms after optimization across all datasets with a significance level of 0.05 (two-tailed), using 60/40 holdout train and test sets

Dataset	Zero R	Decision Tree	Random Forest	Naïve Bayes	SVM
1	12.50	99.98	100.00	100.00	100.00
2	12.50	99.79	100.00	100.00	100.00
3	12.50	99.78	100.00	99.28	100.00
4	12.50	99.18	99.90	99.01	99.99
5	32.45	95.23	98.01	80.79	97.88
6	56.15	95.68	98.11	72.07	97.50
7	31.98	94.81	97.12	75.26	97.14

While slightly optimizing Naïve Bayes resulted in classification improvement and being better than the base case offered by ZeroR, it still was significantly less optimal than the others. As a final test in accuracy,

instead of using a hold train and test sets, a 10-fold cross validation is used to validate the results. These are shown in Table 19.

**Table 19:** Percentage correctly classified for all algorithms after optimization across all datasets with a significance level of 0.05 (two-tailed), using 10-fold cross validation

Dataset	Zero R	Decision Tree	Random Forest	Naïve Bayes	SVM
1	12.50	99.50	100.00	100.00	100.00
2	12.50	99.83	100.00	100.00	100.00
3	12.50	99.93	100.00	99.36	100.00
4	12.50	99.57	99.98	99.02	100.00
5	32.45	96.09	98.39	82.98	98.31
6	56.15	96.19	98.71	72.95	98.01
7	31.98	95.98	97.64	75.95	97.44

These results are relatively consistent with previously completed tests, and therefore, choosing either of the three good performing algorithms for classification based on accuracy in this scenario would seem appropriate. However, given the requirements of this particular AR sensor in relation to the IoT paradigm discussed earlier, more than just analysis of accuracy and kappa metrics are needed.

### 5.2.2 EFFICIENCY

Because the proposed solution is to be implemented in real-world scenarios in the context of an AR interface for the IoT, the value of efficiency discussed here is concerned with more than just the algorithms accuracy. Considering the possibility of real-time AR and embedded solutions, or alternatively remote computation, there is a need to assess algorithms performance on training time, classification time, CPU usage, and model size. Using the 10-fold evaluation results, WEKA provides some metrics that enable this analysis. This is also not a statistical analysis of the metrics here, but rather a general observation of the values to guide further investigative work in the future.

Time taken to train the model is important to consider. Depending on the strategy, model training time may be insignificant if it is only done sporadically. For the given scenario, for example, data acquisition could be consistently active during the day as AR occurs, with models updated overnight while the workplace is empty and used in the consecutive days. For other scenarios, however, where online model training is done for constant improvement during the day, the time taken to compute the models becomes a bigger factor. This is not meant to be an indicator of the actual time it would take an algorithm to perform classification, as this is too dependent on hardware configuration and algorithm implementation. Rather, it is valuable in comparing the algorithms with each other in relation to accuracy, and helpful in establishing the use of an appropriate algorithm given factor other than accuracy only.

The time taken to train each algorithm, indicated by the CPU time spent during training, are shown in Table 20. These results may seem comparatively insignificant considering that the slowest training of a model took only 420 seconds. However, this needs to be considered in terms of two aspects. The first is hardware capability, as these results were completed on a high clock-speed, high-core count water-cooled computer system. While remote model training, such as those completed here would cope with the variance shown in training time, an enclosed energy efficient embedded system without this computing power (and preferred in some IoT applications) would be orders of magnitude slower. Furthermore, this data has been collected from a single user in approximately a square meter of space. If a larger installation covering hundreds of square meters, with possibly hundreds of users simultaneously creating data (the essence of the IoT being available to anyone, anywhere) then this disparity in training time becomes increasingly important as data exponentially grows. This is without considering that only a limited range of activities are accounted for in this testing when there are possibly hundreds or thousands of activities that could be implemented. In the context of this research, the difference between Random Forest and SVM for the largest dataset is greater than 15 times slower, which is not significant at datasets of this scale, but could be

in real-world applications. Thus, it is the comparison to each other that is being considered here, rather than just the actual CPU time spent training. It is fairly obvious that time spent increases as the datasets become larger, and as complexity of the data increases from staged mode data to free mode data. Interestingly, this is not necessarily true for Naïve Bayes, where training time increases seemed only to correspond with instance numbers, as seen in the decrease of training time of the smallest free mode dataset and the largest staged mode dataset. This is apparent too in the free mode datasets, where Naïve Bayes was comparatively fast at training in two of three datasets, and only slightly slower for the largest dataset.

*Table 20: Comparison of CPU time (seconds) taken for model training*

<b>Dataset</b>	<b>Decision Tree</b>	<b>Random Forest</b>	<b>Naïve Bayes</b>	<b>SVM</b>
<b>1</b>	0.22	0.17	0.14	0.13
<b>2</b>	1.36	0.38	0.65	0.45
<b>3</b>	4.01	1.18	2.95	1.24
<b>4</b>	8.42	1.97	4.84	1.99
<b>5</b>	13.81	6.23	3.88	7.32
<b>6</b>	36.85	9.14	5.30	29.52
<b>7</b>	274.68	27.00	28.64	420.41

However, it must be remembered that there was a marked drop in accuracy between data modes, and while it was the worst performing algorithm overall for accuracy, on the smallest staged mode data where accuracy among the algorithms was similar, it is a very fast algorithm to train being only slightly slower than the SVM. It is important to consider that at this low instance number Naïve Bayes could potentially outperform the other algorithms when training time is included as a metric for performance. However, in future work where more activities are to be classified, datasets are only likely to grow larger rather than smaller, and so the performance of algorithms on these smaller datasets may be irrelevant and is another factor in consideration of the importance of training time.

Another interesting aspect is that a J48 decision tree is significantly slower to train than a Random Forest, seemingly unintuitive given that the Random Forest is an ensemble method consisting of decision trees. However, considering that the data points are equal for any given dataset, the computational expense involved in pruning and subtreeRaising to prevent a deep data tree and hence worse case scenarios is relatively worse than the Random Forest approach. Instead, creating subsets of random features and creating more, yet shallower trees on these subsets proves far more efficient, as evident in the exponentially smaller times of the Random Forest approach. Again, while this may not be excessively important depending on the AR modelling approach taken, it is a factor to consider if online model creation is considered or large data training sets are used.

While SVM proved to have excellent accuracy, time taken to train a model using this approach seems to deteriorate on the free mode datasets, despite being one of the fastest algorithms on the staged datasets. This is especially true for the largest dataset, where the algorithm takes substantially longer to train than the next worst algorithm. To assess whether this could be due to the effect of sparse data, a single SMO model was trained using the same holdout train and test set used to improve Naïve Bayes, while disabling normalization and standardization both separately and together, as well as investigating an SVM with a different kernel. The results are shown in Table 21.

**Table 21:** Time taken training an SVM model using normalization, standardization, and an alternative kernel, when used with the 60/40 holdout test with dataset 5

Normalize training data	Standardize training data	Kernel	Time Training (sec)
yes	no	PolyKernel	23.96
no	yes	PolyKernel	23.41
no	no	PolyKernel	25.44
no	no	Normalized PolyKernel	104.75

These results show that while accuracy remained relatively consistent, no remarkable differences in time taken to train were evident. In fact, disabling both normalization and standardization had an adverse effect, although relatively insignificant.

However, shown in Table 22, there was more of a pronounced improvement on time taken to test the models when standardization was used, or when both standardization and normalization were disabled. This is possibly more pertinent to the scenario of real time, real world AR application. For an AR sensor to truly reflect the activities of a user, the ability to use the model to quickly classify what activity the user is participating in before a change of activity occurs is important. This is even more paramount in an IoT environment where inferences about the user’s activity may influence and inform other components of the system, and delays in classification could lead to a poor performing system and frustrated users. However, considering that the worst model for a SVM using a PolyKernel is about twice as slow as the best performing model in terms of testing, the observable impact would be minimal given that the units of measurement are in seconds. It would be interesting to see the effect that weaker computing power would have on these metrics, and to see if there exists a linear correlation between the variance shown here and computing capability.

*Table 22: Time taken evaluating the SVM models from Table 21*

<b>Normalize training data</b>	<b>Standardize training data</b>	<b>Kernel</b>	<b>Time Testing (sec)</b>
<b>yes</b>	no	PolyKernel	0.27
<b>no</b>	yes	PolyKernel	0.14
<b>no</b>	no	PolyKernel	0.13
<b>no</b>	no	Normalized PolyKernel	28.36

To assess this, the metrics WEKA provides are the time taken to complete testing and the cpu time used in testing. Again, this is not an in-depth

analysis but provides a good observation of the values in comparison to each of the algorithms. Of further note is that the values given indicate the length of testing all instances in the dataset, not per-instance, meaning classification of a single instance would be faster than those shown. However, it is the comparative values that are being discussed here rather than the values themselves, as these are too hardware and dataset dependant to provide any generalization about classification performance. The results are shown in Table 23.

**Table 23:** Time taken (seconds) to classify the test set of each dataset

Dataset	Decision Tree	Random Forest	Naïve Bayes	SVM
1	0.00	0.00	0.09	0.00
2	0.00	0.00	0.26	0.02
3	0.00	0.01	0.62	0.04
4	0.00	0.01	0.89	0.07
5	0.00	0.03	0.88	0.05
6	0.00	0.04	1.22	0.07
7	0.01	0.10	3.87	0.36

It is evident that J48 decision trees are by far the fastest algorithm in testing, followed by Random Forest, SVM, and Naïve Bayes. Decision trees being the fastest algorithm for classification is unsurprising in optimally sized trees, where classification is but a simple choice of a path. Likewise, Random Forest using the same procedure, but having some extra steps, such as the majority vote, means it is unsurprisingly efficient in classifying a new instance too. Naïve Bayes, while very fast at training compared with the other models, is much slower than the others at classifying. As seen with the improvements possible in testing time with SVM, it still lags slightly behind both of these. Therefore, when considering classification time, especially if data transferal latency for remote computation is to be considered, a decision tree-based implementation seems the most efficient. Another metric that should be considered is model size. This is particularly of interest with an embedded implementation, which may become relevant

in a true IoT implementation. Such an implementation would have lower computational capability which may exacerbate slow training and classification times as discussed earlier, but would also be affected by storage limitations.

Shown in Table 24 are the serialized model sizes. It is evidently clear that J48 decision trees consistently create the smallest model regardless of the dataset type or size. However, for the Random Forest approach, each training model size gets larger with size, and there is a marked increase in model size between the two different types of data, with an enormous model for the larger dataset in comparison to the other models. Naïve Bayes rather interestingly starts with a much larger model than the other algorithms in the smaller datasets, but more or less remains around this size and even decreases for the smaller free mode dataset that consists of more instances, increasing in size for the largest dataset. After Naïve Bayes, SVM has the largest model for the smaller datasets, shrinking for the first few free mode datasets, and increasing for the last.

**Table 24:** Observed average model size (bytes) for each algorithm and dataset (Java Serialized Objects) after ten iterations

Dataset	Decision Tree	Random Forest	Naïve Bayes	SVM
1	204565.00	355692.70	2987017.50	856795.20
2	204565.00	494114.90	3093029.30	803025.00
3	204565.00	801807.90	3548124.30	881950.20
4	207102.80	1026173.30	3821126.50	980416.20
5	240443.60	5199984.20	2881481.20	707916.20
6	247750.00	6359232.30	2796980.00	657036.20
7	364325.00	17227015.20	4526798.90	1090934.40

Therefore, if model size is a factor in assessing algorithm suitability and assuming free mode type of datasets in a real-world implementation, a decision tree or SVM implementation may be an optimal solution.

### 5.3 SUMMARY

In an AR sensor acting as an interface to the IoT, machine learning is an important component of the system, and accuracy of classification is the most important metric to consider in machine learning. Without accurate classification, it would not be performing its main objective, making discussion of the other metrics futile.

However, selecting the correct algorithm based on accuracy is dependent on the datasets. Assuming a real-world implementation with moderately sized datasets and averaging accuracy performance on the free mode datasets, then of the algorithms tested the Random Forest (98.25%) or SVM (97.92%) approach is best, followed closely in performance by a J48 decision tree (96.09%). Naïve Bayes (77.29%) is much lower than these other algorithms even after limited tuning.

Considering the other metrics in relation to free mode datasets, however, show that Naïve Bayes averaged 12.61 seconds to train, while Random Forests took only slightly longer at 14.12 seconds on average. J48 decision trees averaged 108.45 seconds, while SVM took the longest at 152.42 seconds. For both J48 and SVM, as the instance numbers increased, the training time needed also increased dramatically, distorting these averages, and while Naïve Bayes and Random Forests training time did increase too, the increase in time to train was more linear. If considering this metric in correlation with the accuracy, a Random Forest approach may seem to be the optimal choice.

If consideration of time taken to classify is more important in offering a responsive real-time AR classification system, then a J48 Decision tree was easily the faster of the algorithms (0.03 sec), followed by Random Forest (0.06 sec), SVM (0.16 sec), and Naïve Bayes (1.99 sec). While hardware used would affect these values, with the system and sensors current implementation, a request to the microcontroller followed by a full pass of the sensor and consequent write to serial was observed to take 0.04 to 0.05 seconds. Ignoring the overheads due to processing and writing to file, only one algorithm is able to complete a classification within this time. This

means it could be possible to classify an existing instance while issuing a reading from the sensor, and have this classification completed before the next sensor reading is presented. However real time classification was not attempted here, and the discussion of these metrics are used to inform further investigation. Likewise, with the models not limited by the constraints of hardware or real-time classification problem, model size is less of a concern. However, considering the large variation between the serialized model size of each algorithm, it is important component for any future investigation. It showed that while a Random Forest approach is one of the better algorithms in accuracy and time taken for training and classification, the models produced are relatively large in comparison, possibly having consequences for embedded implementations or where hardware restrictions exist (while noting these are serialized Java Objects).

## 6 LIMITATIONS, FUTURE RESEARCH AND CONCLUSION

This research has evaluated the implementation of an interface using readily available and cost-effective parts in an artefact that is both non-intrusive and intuitive for a user in this particular use-case. While these aspects were based on fundamental principles of the IoT discussed in section 2.1 and 2.4, the evaluation results of section 5 show that these components, when combined with an appropriate machine learning algorithm, are able to provide highly accurate recognition of a limited amount of activities.

This was achieved by first conducting a design process to find the optimal construction of a suitable physical layer that would not impede activity or require changes to normal activity in the given scenario. The design process led to a base layer consisting of thin rubber with a backing lined by copper rows. This interfaced with the ground, and was able to stay fixed in its location, whether it was a soft or hard surface. The top layer consisted of a layer of carpet with a thin bonded urethane backing lined with copper columns. This layer provided the interface between a user, either directly or through furniture, and the base layer. A piezoresistive material commercially known as Velostat rests between the base layer and top layer, providing the change in resistance and therefore pressure applied. The intersections of the bottom layers rows and the top layers columns provide a mechanism for locating the pressure. The layout of these columns and rows provide a sensing electrode every 1.8 square centimetre. The spatially located pressure data provided by the sensor in this configuration allowed for a high level of classification accuracy, as is evident in the results.

In selection of an appropriate microcontroller to drive the sensor, a Teensy 3.2 was used, providing ample sampling rates, sufficient pins connected to two ADC with appropriate high bit resolution, as well as being able to supply 3.3V to the sensor itself. The use of eight multiplexers was used to accommodate all the rows and columns of the sensor. This microcontroller writes to serial, where a program is waiting for this input and is written to a file for machine learning processing.

An analysis of suitable machine learning classification algorithms was performed, assessing accuracy, training time, testing time, and model size. This showed that Naïve Bayes was not as suitable a candidate for the data produced from the sensor due to the lower accuracy, while J48 Decision Trees, Random Forests, and SVM provided remarkable accuracy, but each had drawbacks depending on the importance placed upon the various metrics. For this implementation where post processing occurred, the Random Forest algorithm offered the best performance among the majority of metrics considered, only suffering from a larger model size than that of the other algorithms.

## 6.1 LIMITATIONS

There are a number of limitation that are of note.

### *6.1.1 SAMPLE SIZE*

While many datasets were constructed with different chair orientations and start points, all datasets were the result of one user's interactions. The assumption is made that while variation between user data would be inevitable due to variables such as weight or how users perform the activity, as well as the variability produced by different chairs or office environments, if accuracy levels achieved are consistently well for one user, then another pattern from any other user's activity would be identifiable by the machine learning algorithm, and another model created based on this new data. However, it is unclear whether the data produced between various users would produce similarly distributed data points and thus whether the optimal algorithm found in this research would produce outcomes that conclude with the same results.

This is also evident within the testing of the physical design of the interface itself. While only a little different to a typical carpet mat, the conclusion that it was intuitive to use and non-inhibitive of general movement is only based on the observation of a single user. For others the experience may differ depending on how they operate within the given scenario space.

Importantly, it must be noted that the user in this research was also the researcher. While effort was made to ensure that activities were performed naturally and without bias (such as creating many different datasets and randomly selecting seven to use), knowledge of the sensor's configuration and operation may have led to the unconscious exaggeration and positioning of pressure in order to amplify the differences between the activities and increase classification accuracy. It would be prudent in future research to use multiple participants in evaluation of the sensors performance.

### *6.1.2 LIMITED ACTIVITIES*

The restricted number of activities for recognition were chosen based on an observation of office workers and their activities. The selection was further influenced by the spatial variation between activities and the prospect to assess the sensors capabilities between subtle weight shifts (e.g. from keyboard to mouse) and more pronounced changes in pressure distribution (e.g. relaxing to away). However, the activities chosen did not categorically represent all observed activities and were restricted to eight classes due to time restrictions in the research. For a ubiquitous and accurate AR sensor, the ability to classify from a multitude of possible activities would be imperative and is unfortunately not implemented nor assessed here. In that regard, only the foundational work for a general AR sensor has been addressed.

Related to this, the use-case of an office desk worker only offers a limited perspective on the effectiveness of the sensor in AR applications outside this scenario. While classification with only subtle pressure distribution changes were highly successful, other use-cases may have such minimal variation between activities that in the sensors current implementation, they are undetectable and inhibit accurate classification.

### *6.1.3 DATASET AND ALGORITHM TRAINING*

While many datasets were created and could have been used in the machine learning process to improve classification, time restraints related to data complexity and training meant only a select number were chosen. While the chosen datasets were meant to be a representative selection of all available

datasets, a bias existed toward the smaller datasets. This was to enable faster optimization and re-training, because as the dataset sizes grew so did the training time, and optimization and exploration of the algorithms was becoming prohibitively expensive as even the slightest change meant creation of a new model. However, using more of the datasets to test the sensor's classification ability would have provided a more comprehensive and significant result.

This is also true of the chosen algorithms. While the approach is clearly one of classification, not every algorithm capable of classification was assessed in this research. The conclusion therefore can only state which of the used algorithms is sufficient for the task but cannot generalize about the same algorithm being the most efficient in this scenario.

#### *6.1.4 IOT INCLUSION*

The research first proposed the digitisation of human activity for inclusion into the IoT system. While the research used principles of ubiquitous computing and the IoT paradigm in informing design choices, the research did not investigate actual IoT inclusion or its effectiveness in an IoT system. Indeed, after the physical component was completed, the core consideration of the research become orientated toward AR strategies and feasibility for effective classification in an IoT paradigm. While the exploration and assessment in this research is required to even begin investigation into the effectiveness of this AR sensor as an IoT interface, concluding the research in this state leaves one of the more foremost questions of this research unanswered.

#### *6.1.5 REAL-TIME VS POST-PROCESSING OF DATA FOR CLASSIFICATION*

Related to effectiveness within an IoT system is the difference in real-time classification or delaying the classification procedure. This research, for simplicity and time restraints, chose classification as a post-data collection operation. If understanding trends or behaviours within a given environment are the proposed outcomes, then the sensor in this implementation is adequate. However, as with any interface, responsiveness from the view of the user would require a real-time classification approach

and could also extend to real-time training of new data. As prevalent in the literature review, the inclusion of humans within the IoT was important, and this sensor does not address this. While it does offer a foundational exploration into AR with guiding principles from the IoT paradigm, not implementing real-time classification further deviates away from exploring an IoT interface and more toward an alternative AR artefact.

Given that vast amounts of data would be generated in a large ground-based installation, a possible avenue to explore would be data stream mining. This alleviates problems associated with the storage of the generated data, simply accessing it once and discarding it. This would be an effective approach where limited capabilities are a factor, such as in small embedded microcontrollers with minimal storage. Furthermore, the possibility of online training using data stream mining to further improve classification in an evolving environment, could be achieved with such an approach, and would be interesting to investigate further.

## 6.2 FUTURE RESEARCH

As discussed, this research has explored existing domains of research and viewed it from another context, that of the IoT. While somewhat successful in creating an AR interface, the previous discussion demonstrated that integration into the IoT system and operation as a competent human interface will require further exploration.

### 6.2.1 PHYSICAL ARTEFACT DESIGN

Cost, usability, invisibility, and reliability were four factors considered in the physical construction of the sensor. These could be further investigated by addressing some of the issues mentioned in the previous section.

However, other factors that could impact the physical design of the sensor should be considered as well.

For example, the observed area of user movement in the given use-case dictated the physical size of the sensor. Using these dimensions, the assumption was made that a higher density of electrodes would offer higher classification accuracy, and the electrodes were consequently constructed

using the highest density possible. An investigation into the minimal number of electrodes needed to still provide high classification accuracy, thereby reducing complexity and cost of the sensor, would prove valuable in creating a more cost-effective implementation.

Further still, this sensor used a carpet-like interface in the assumption that it offered the most ubiquitous and invisible properties needed for real-world implementation. However, hard-floor surfaces are also used in the commercial environments and investigating other possible surfaces in relation to this AR sensor system would provide a more complete evaluation in creating a more general and ubiquitous AR sensor.

While this research was conducted within a short period of time, in more permanent installs where it replaces the current ground covering, this sensor may be required to operate for years. Investigation into the accuracy and its operating ability and accuracy drift over time would be interesting.

### *6.2.2 SENSOR ARCHITECTURE*

This research consists of four distinct components; the physical sensor; electrical components; data acquisition; data classification. These are physically separate, with the last two components conducted on a remote computer. While no assumptions were made as to the optimal architecture for such an AR implementation, investigating possible alternatives could provide a more effective solution in context of the IoT. For example, exploring the merits of a fully-enclosed system from pressure detection to classification, rather than distinct components, would advance some of the principles of ubiquity discussed in the literature review, and an embedded implementation would help advance that ubiquitous vision.

Further to this, architecture investigation could provide even insight into multiple instances of the sensor that combine their classification efforts, allowing assessment of the effectiveness of a modular sensor when cooperating in larger environments. Exploring this avenue would help determine feasibility and possible unknown areas for deployment.

Also, the sensor is limited to single-user based AR, given the use-case. However, as many activities outside of this scenario include multi-user engagement, a general, ubiquitous, and invisible AR system would need to have the mechanisms to allow for these types of activities. Exploration into multi-user AR is already a domain of interest for researchers, but again using more intrusive techniques such as cameras. A multi-user AR system more aligned with some of the principles discussed here could yield some interesting results.

### *6.2.3 MACHINE LEARNING FOR AR CLASSIFICATION*

There are many opportunities to explore within AR in regard to machine learning algorithms. Of late, deep learning techniques in machine learning have become the fascination of many researchers from various domains, including AR as discussed in the literature review. While an attempt was made in this research, it was not successfully instantiated due to time constraints and complexity. However, as seen in some of the remarkable research to date involving this technique, investigating the performance on a sensor such as this would be useful. Beside investigating other machine learning techniques, further optimization of the current algorithms for other environments and use-cases would prove useful in further development of a general SF interface for the IoT.

An alternative possibility would be the investigation of data stream mining techniques as a means to reduce the computational cost in building the classification models. Various techniques for mining data streams exist that revolve around identifying concept drift and then incrementally rebuilding the classifier only when the input data is different than previously observed in the data stream.

Another interesting avenue for machine learning exploration would be the use of classification techniques within embedded instantiations. With specialised machine learning hardware available, the use of these within AR to realise a modular system could provide insight into the differences between the implementation presented here and a more specialised system, in terms of aspects such as efficiency in classification.

An important aspect that would provide a more responsive interface is the possibility of predicting activity before it occurs, with traditional AR machine learning more concerned with accurate classification of current or previously recorded activities. Investigation into this through the use of a sensor with the same restrictions would allow further exploration into the viability of the sensor as an IoT interface.

#### *6.2.4 STAKEHOLDER ENGAGEMENT*

While the feasibility of the sensor's physical design was explored from a single end-user's perspective, there was little discussion on the actual deployment of the sensor in a real-world setting, neglecting this for due to time restraints. Therefore, stakeholder's perspectives would be influential in realisation of a ubiquitous sensor. These could include participants from across the spectrum, from analysis of the sensor in action with multiple end-users, to participation from potential implementers of a large-scale sensor. Rather than performing this research in the lab, having these parties input into the development and deployment of a sensor would be both interesting and vital for further exploration.

#### *6.2.5 IOT VALUE*

As explained in the literature review, the IoT is a very encompassing system. While this research and consequent sensor provide an intuitive and non-intrusive way into digitising human activity with the intention of human inclusion into the IoT, more research is needed into the effectiveness of this solution and its role as an interface for this system. This could be approached from many different perspectives. For example, it could include exploration of low-level hardware and communication technologies between other devices within the IoT system. A more qualitative perspective could investigate the perceived value to end users of a ubiquitous AR sensor in providing automation among other IoT devices, or alternatively observe and investigate the cognitive process of users when somewhat personal information is being constantly shared within a larger IoT system. The possibilities are broad.

### 6.3 CONCLUSION

The view of this research, based upon analysis of related literature, is that more investigation into a suitable way to include humans into the IoT is needed. This research proposes that an activity recognition, ground-based sensor, capable of digitising human actions and inferring activities may be a possible path to achieving this.

It further proposes, based upon more literature, that such a sensor needs to be cost effective, invisible (unperceivable to the user), and accurate. The implementation of a ground-based AR sensor within this research offers an architecture that aims to meet these needs.

This was attempted by assessing design choices in material and the sensors physical properties in construction of a sensor for an office workspace. Next was the use of a microcontroller and supporting software to perceive the pressure and position of objects on its surface, and consequently the digitisation of its user's actions.

Then, using machine learning software, these acquired data points were converted into a set of eight chosen activities in the workplace, and the performance of the mat in relation to AR was assessed.

When combined with the right machine learning algorithm, results show the ground-based AR sensor in this research, produced cost effectively and not too dissimilar from a traditional carpet mat as to remain invisible, was capable of very high accuracy in recognition of a restricted number of activities pertinent to the given use case. Future research would investigate the sensors effectiveness in an IoT scenario.

## 7 REFERENCES

- Accenture. (2015). Industrial Internet Insights Report for 2015. GE. Retrieved from <https://www.ge.com/digital/sites/default/files/industrial-internet-insights-report.pdf>
- Adafruit Industries. (n.d.). Pressure-Sensitive Conductive Sheet (Velostat/Linqstat). Retrieved February 4, 2018, from <https://www.adafruit.com/product/1361>
- Addlesee, M. D., Jones, A., Livesey, F., & Samaria, F. (1997). The ORL active floor [sensor system]. *IEEE Personal Communications*, 4(5), 35–41. <https://doi.org/10.1109/98.626980>
- Aggarwal, J. k., & Ryoo, M. s. (2011). Human activity analysis. *ACM Computing Surveys*, 43(3), 1–43. <https://doi.org/10.1145/1922649.1922653>
- Ahmad, H., & Nor, N. A. M. (2017). Detection and analysis of activity recognition for and rehabilitation. In *Control and System Graduate Research Colloquium (ICSGRC), 2017 IEEE 8th* (pp. 246–250). IEEE.
- Ahmed, E., & Rehmani, M. H. (2017). Mobile Edge Computing: Opportunities, solutions, and challenges. *Future Generation Computer Systems*, 59. <https://doi.org/10.1016/j.future.2016.09.015>
- Ahmed, E., Yaqoob, I., Hashem, I. A. T., Khan, I., Ahmed, A. I. A., Imran, M., & Vasilakos, A. V. (2017). The role of big data analytics in Internet of Things. *Computer Networks*, 129(Part 2), 459–471. <https://doi.org/10.1016/j.comnet.2017.06.013>
- Alaa, M., Zaidan, A. A., Zaidan, B. B., Talal, M., & Kiah, M. L. M. (2017). A review of smart home applications based on Internet of Things. *Journal of Network and Computer Applications*, 97(Supplement C), 48–65. <https://doi.org/10.1016/j.jnca.2017.08.017>
- Alharthi, A., Krotov, V., & Bowman, M. (2017). Addressing barriers to big data. *Business Horizons*, 60(3), 285–292. <https://doi.org/10.1016/j.bushor.2017.01.002>
- Al-Naimi, I., & Wong, C. B. (2017). Indoor human detection and tracking using advanced smart floor. In *2017 8th International Conference on Information and Communication Systems (ICICS)* (pp. 34–39). <https://doi.org/10.1109/IACS.2017.7921942>
- Anagnostopoulos, T., Anagnostopoulos, C., Hadjiefthymiades, S., Kyriakakos, M., & Kalousis, A. (2009). Predicting the location of mobile users: a machine learning approach. In *Proceedings of the 2009 international conference on Pervasive services* (pp. 65–72). ACM.

- Arduino Reference. (n.d.). Retrieved February 7, 2018, from <https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/>
- Arthur L. Samuel. (2013). Some studies in machine learning using the game of Checkers.
- Ashraf, Q. M., & Habaebi, M. H. (2015). Introducing autonomy in internet of things. *Recent Advances in Computer Science, WSEAS Publishing*, 215–221.
- Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787–2805. <https://doi.org/10.1016/j.comnet.2010.05.010>
- Atzori, L., Iera, A., & Morabito, G. (2017). Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, 56(Supplement C), 122–140. <https://doi.org/10.1016/j.adhoc.2016.12.004>
- Bakshi, K. (2016). Big data analytics approach for network core and edge applications. In *2016 IEEE Aerospace Conference* (pp. 1–10). <https://doi.org/10.1109/AERO.2016.7500560>
- Balta-Ozkan, N., Davidson, R., Bicket, M., & Whitmarsh, L. (2013). Social barriers to the adoption of smart homes. *Energy Policy*, 63(Supplement C), 363–374. <https://doi.org/10.1016/j.enpol.2013.08.043>
- Barbosa, J. L. V. (2015). Ubiquitous computing: Applications and research opportunities. In *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)* (pp. 1–8). <https://doi.org/10.1109/ICCIC.2015.7435625>
- Batchelor, J. (2017). You’ve Been Hacked! *Science World*, 73(9), 6–9.
- Bello, O., & Zeadally, S. (2017). Toward efficient smartification of the Internet of Things (IoT) services. *Future Generation Computer Systems*. <https://doi.org/10.1016/j.future.2017.09.083>
- Bergesio, L., Bernardos, A. M., & Casar, J. R. (2017). An Object-Oriented Model for Object Orchestration in Smart Environments. *Procedia Computer Science*, 109(Supplement C), 440–447. <https://doi.org/10.1016/j.procs.2017.05.415>
- Bicocchi, N., Mamei, M., Prati, A., Cucchiara, R., & Zambonelli, F. (2008). Pervasive Self-Learning with Multi-modal Distributed Sensors. In *2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops* (pp. 61–66). <https://doi.org/10.1109/SASOW.2008.51>

- Bränzel, A., Holz, C., Hoffmann, D., Schmidt, D., Knaust, M., Lühne, P., ... Baudisch, P. (2013). GravitySpace: Tracking Users and Their Poses in a Smart Room Using a Pressure-sensing Floor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 725–734). New York, NY, USA: ACM.  
<https://doi.org/10.1145/2470654.2470757>
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32.  
<https://doi.org/10.1023/A:1010933404324>
- Brownlee, J. (2016, March 16). Supervised and Unsupervised Machine Learning Algorithms. Retrieved December 9, 2017, from <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>
- Buckley, J. P., Hedge, A., Yates, T., Copeland, R. J., Loosemore, M., Hamer, M., ... Dunstan, D. W. (2015). The sedentary office: an expert statement on the growing case for change towards better health and productivity. *Br J Sports Med*, 49(21), 1357–1362.  
<https://doi.org/10.1136/bjsports-2015-094618>
- Bulling, A., Ward, J. A., Gellersen, H., & Troster, G. (2011). Eye Movement Analysis for Activity Recognition Using Electrooculography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4), 741–753. <https://doi.org/10.1109/TPAMI.2010.86>
- Cassidy, A. P., & Deviney, F. A. (2014). Calculating feature importance in data streams with concept drift using Online Random Forest. In *2014 IEEE International Conference on Big Data (Big Data)* (pp. 23–28). <https://doi.org/10.1109/BigData.2014.7004352>
- Castellano, G., Villalba, S. D., & Camurri, A. (2007). Recognising human emotions from body movement and gesture dynamics. In *International Conference on Affective Computing and Intelligent Interaction* (pp. 71–82). Springer.
- Ceballos, J. C., Nurgiyatna, N., Scully, P., & Ozanyan, K. B. (2011). Smart carpet for imaging of objects' footprint by photonic guided-path tomography. In *AFRICON, 2011* (pp. 1–6). <https://doi.org/10.1109/AFRCON.2011.6071986>
- Cerf, V., & Senges, M. (2016). Taking the Internet to the Next Physical Level. *Computer*, 49(2), 80–86. <https://doi.org/10.1109/MC.2016.51>
- Chang, S., Ham, S., Kim, S., Suh, D., & Kim, H. (2010). Ubi-Floor: Design and Pilot Implementation of an Interactive Floor System. In *2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics* (Vol. 2, pp. 290–293). <https://doi.org/10.1109/IHMSC.2010.172>

- Chen, L., Hoey, J., Nugent, C. D., Cook, D. J., & Yu, Z. (2012). Sensor-Based Activity Recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6), 790–808. <https://doi.org/10.1109/TSMCC.2012.2198883>
- Cheng, J., Sundholm, M., Zhou, B., Hirsch, M., & Lukowicz, P. (2016). Smart-surface: Large scale textile pressure sensors arrays for activity recognition. *Pervasive and Mobile Computing*, 30, 97–112. <https://doi.org/10.1016/j.pmcj.2016.01.007>
- Cheng, J., Sundholm, M., Zhou, B., Kreil, M., & Lukowicz, P. (2014). Recognizing subtle user activities and person identity with cheap resistive pressure sensing carpet. In *Proceedings - 2014 International Conference on Intelligent Environments, IE 2014* (pp. 148–153). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/IE.2014.29>
- Cicirelli, F., Fortino, G., Guerrieri, A., Spezzano, G., & Vinci, A. (2017). Metamodeling of Smart Environments: from design to implementation. *Advanced Engineering Informatics*, 33(Supplement C), 274–284. <https://doi.org/10.1016/j.aei.2016.11.005>
- Cicirelli, F., Guerrieri, A., Spezzano, G., Vinci, A., Briante, O., Iera, A., & Ruggeri, G. (2017). An edge-based approach to develop large-scale smart environments by leveraging SIoT. In *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)* (pp. 738–743). <https://doi.org/10.1109/ICNSC.2017.8000182>
- Cottone, P., Gaglio, S., Lo Re, G., & Ortolani, M. (2015). User activity recognition for energy saving in smart homes. *Pervasive and Mobile Computing*, 16(Part A), 156–170. <https://doi.org/10.1016/j.pmcj.2014.08.006>
- Cradlepoint. (2017, November 14). Global Study Reveals IoT Adoption is at 69 Percent Despite Security Concerns. Retrieved November 15, 2017, from <http://globenewswire.com/news-release/2017/11/14/1185889/0/en/Global-Study-Reveals-IoT-Adoption-is-at-69-Percent-Despite-Security-Concerns.html>
- Da Xu, L., He, W., & Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4), 2233–2243.
- Delsing, J., Eliasson, J., Van Deventer, J., Derhamy, H., & Varga, P. (2017). Enabling IoT automation using local clouds. In *2016 IEEE 3rd World Forum on Internet of Things, WF-IoT 2016* (pp. 502–507). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/WF-IoT.2016.7845474>
- Díaz-Uriarte, R., & De Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1), 3.

- Dossey, L. (2014). FOMO, Digital Dementia, and Our Dangerous Experiment. *Explore: The Journal of Science and Healing*, 10(2), 69–73. <https://doi.org/10.1016/j.explore.2013.12.008>
- Duffy, J. (2014, June 26). 8 Internet things that are not IoT. Retrieved October 26, 2017, from <https://www.networkworld.com/article/2378581/internet-of-things/8-internet-things-that-are-not-iot.html>
- Essl, A., Ortner, A., Haas, R., & Hettegger, P. (2017). Machine learning analysis for a flexibility energy approach towards renewable energy integration with dynamic forecasting of electricity balancing power. In *2017 14th International Conference on the European Energy Market (EEM)* (pp. 1–6). <https://doi.org/10.1109/EEM.2017.7981877>
- Fauquex, M., Goyal, S., Evequoz, F., & Bocchi, Y. (2015). Creating people-aware IoT applications by combining design thinking and user-centered design methods. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)* (pp. 57–62). <https://doi.org/10.1109/WF-IoT.2015.7389027>
- Finlay, J., Connor, A., & Pears, R. (2011). Mining software metrics from Jazz. In *Software Engineering Research, Management and Applications (SERA), 2011 9th International Conference on* (pp. 39–45). IEEE.
- Fortino, G., Guerrieri, A., Russo, W., & Savaglio, C. (2015). Towards a Development Methodology for Smart Object-Oriented IoT Systems: A Metamodel Approach. In *2015 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1297–1302). <https://doi.org/10.1109/SMC.2015.231>
- Gartner Says 8.4 Billion Connected. (n.d.). Retrieved October 12, 2017, from [//www.gartner.com/newsroom/id/3598917](http://www.gartner.com/newsroom/id/3598917)
- Giovanelli, D., & Farella, E. (2016). Force Sensing Resistor and Evaluation of Technology for Wearable Body Pressure Sensing [Research article]. <https://doi.org/10.1155/2016/9391850>
- Giri, A., Dutta, S., & Neogy, S. (2017). Enabling agricultural automation to optimize utilization of water, fertilizer and insecticides by implementing Internet of Things (IoT). In *2016 International Conference on Information Technology, InCITE 2016 - The Next Generation IT Summit on the Theme - Internet of Things: Connect your Worlds* (pp. 125–131). IEEE Inc. <https://doi.org/10.1109/INCITE.2016.7857603>
- Gonçalves, R., Carvalho, N. B., Pinho, P., & Roselli, L. (2014). Smart environment technology as a possible enabler of smart cities. In *2014 IEEE MTT-S International Microwave Symposium (IMS2014)* (pp. 1–3). <https://doi.org/10.1109/MWSYM.2014.6848676>

- Gonçalves, R., Reis, J., Santana, E., Carvalho, N. B., Pinho, P., & Roselli, L. (2013). Smart floor: Indoor navigation based on RFID. In *2013 IEEE Wireless Power Transfer (WPT)* (pp. 103–106). <https://doi.org/10.1109/WPT.2013.6556893>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Gračanin, D., Matković, K., & Wheeler, J. (2015). An approach to modeling Internet of Things based smart built environments. In *2015 Winter Simulation Conference (WSC)* (pp. 3208–3209). <https://doi.org/10.1109/WSC.2015.7408473>
- Gram-Hanssen, K., & Darby, S. J. (2018). Research paper: “Home is where the smart is”? Evaluating smart home research and approaches against the concept of home. *Energy Research & Social Science*, *37*, 94–101. <https://doi.org/10.1016/j.erss.2017.09.037>
- Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, *37*(2), 337-A6.
- Gu, T., Chen, S., Tao, X., & Lu, J. (2010). An unsupervised approach to activity recognition and segmentation based on object-use fingerprints. *Data & Knowledge Engineering*, *69*(6), 533–544. <https://doi.org/10.1016/j.datak.2010.01.004>
- Guan, D., Ma, T., Yuan, W., Lee, Y.-K., & Sarkar, A. M. J. (2011). Review of Sensor-based Activity Recognition Systems. *IETE Technical Review*, *28*(5), 418–433. <https://doi.org/10.4103/0256-4602.85975>
- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660. <https://doi.org/10.1016/j.future.2013.01.010>
- Halter, O., Kromer, M., Kutemperor, N., & Soares, D. (2016). Dipping a Toe into Data Lakes. *Business Intelligence Journal*, *21*(2), 40–46.
- Han, S. N., & Crespi, N. (2017). Semantic service provisioning for smart objects: Integrating IoT applications into the web. *Future Generation Computer Systems*, *76*, 180–197. <https://doi.org/10.1016/j.future.2016.12.037>
- Hatfield, J., & Murphy, S. (2007). The effects of mobile phone use on pedestrian crossing behaviour at signalised and unsignalised intersections. *Accident Analysis & Prevention*, *39*(1), 197–205. <https://doi.org/10.1016/j.aap.2006.07.001>
- Hernández, M. E. P., & Reiff-Marganiec, S. (2016). Towards a Software Framework for the Autonomous Internet of Things. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)* (pp. 220–227). <https://doi.org/10.1109/FiCloud.2016.39>

- Hevner, A., & Chatterjee, S. (2010). *Design Research in Information Systems: Theory and Practice*. Springer Science & Business Media.
- Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian Journal of Information Systems*, 19(2), 4.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), 75–105.
- HP News - HP Study Reveals 70 Percent of Internet of Things Devices Vulnerable to Attack. (n.d.). Retrieved November 15, 2017, from <http://www8.hp.com/us/en/hp-news/press-release.html?id=1744676#.WgzD47aB2-s>
- Hsieh, W., Cannon, A., Lima, A. R., Hsieh, W. W., & Cannon, A. J. (2015). Nonlinear regression in environmental sciences using extreme learning machines: A comparative evaluation. *ENVIRONMENTAL MODELLING & SOFTWARE*, 73, 175–188.
- Hu, F. (2016). *Security and privacy in internet of things (IoTs): models, algorithms, and implementations*. Boca Raton : CRC Press, [2016].
- Iannacci, J., Sordo, G., Serra, E., & Schmid, U. (2015). A novel MEMS-based piezoelectric multi-modal vibration energy harvester concept to power autonomous remote sensing nodes for Internet of Things (IoT) applications. In *2015 IEEE SENSORS* (pp. 1–4). <https://doi.org/10.1109/ICSENS.2015.7370550>
- IDC. (2014, April). Executive Summary: Data Growth, Business Opportunities, and the IT Imperatives | The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things. Retrieved August 30, 2016, from <http://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm>
- Iqbal Muhammad, & Zhu Yan. (2015). Supervised Machine Learning Approaches: A Survey. *ICTACT Journal on Soft Computing, Vol 5, Iss 3, Pp 946-952 (2015)*, (3), 946.
- Kalischewski, K., Wagner, D., Velten, J., & Kummert, A. (2017). Activity recognition for indoor movement and estimation of travelled path. In *2017 10th International Workshop on Multidimensional (nD) Systems (nDS)* (pp. 1–5). <https://doi.org/10.1109/NDS.2017.8070613>
- Kang, S. H., Kim, Y. H., Lee, E. J., Lee, S. G., Min, B. C., An, J., & Kim, D. H. (2011). Implementation of Smart Floor for multi-robot system. In *The 5th International Conference on Automation, Robotics and Applications* (pp. 46–51). <https://doi.org/10.1109/ICARA.2011.6144854>
- Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., & Chouvarda, I. (2017). Machine Learning and Data Mining Methods in

- Diabetes Research. *Computational and Structural Biotechnology Journal*, 15, 104–116. <https://doi.org/10.1016/j.csbj.2016.12.005>
- Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *Computer*, 36(1), 41–50. <https://doi.org/10.1109/MC.2003.1160055>
- Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012). Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges. In *2012 10th International Conference on Frontiers of Information Technology* (pp. 257–260). <https://doi.org/10.1109/FIT.2012.53>
- Kim, E., Helal, S., & Cook, D. (2010). Human Activity Recognition and Pattern Discovery. *IEEE Pervasive Computing*, 9(1), 48–53. <https://doi.org/10.1109/MPRV.2010.7>
- Kim, S.-B., Han, K.-S., Rim, H.-C., & Myaeng, S. H. (2006). Some effective techniques for naive bayes text classification. *IEEE Transactions on Knowledge and Data Engineering*, 18(11), 1457–1466.
- Klack, L., Möllering, C., Ziefle, M., & Schmitz-Rode, T. (2011). *Future care floor: A sensitive floor for movement monitoring and fall detection in home environments* (Vol. 55 LNICST). [https://doi.org/10.1007/978-3-642-20865-2\\_27](https://doi.org/10.1007/978-3-642-20865-2_27)
- Kolekar, M. H., & Dash, D. P. (2016). Hidden Markov Model based human activity recognition using shape and optical flow based features. In *2016 IEEE Region 10 Conference (TENCON)* (pp. 393–397). <https://doi.org/10.1109/TENCON.2016.7848028>
- Kortuem, G. ( 1 ), Kawsar, F. ( 1 ), Fitton, D. ( 2 ), & Sundramoorthy, V. ( 3 ). (2010). Smart objects as building blocks for the internet of things. *IEEE Internet Computing*, 14(1), 44–51. <https://doi.org/10.1109/MIC.2009.143>
- Krishna, G. (2015). *The Best Interface Is No Interface: The simple path to brilliant technology*. New Riders.
- Kwon, Y., Kang, K., & Bae, C. (2014). Unsupervised learning for human activity recognition using smartphone sensors. *Expert Systems with Applications*, 41(14), 6067–6074. <https://doi.org/10.1016/j.eswa.2014.04.037>
- Lara, O. D., & Labrador, M. A. (2013). A Survey on Human Activity Recognition using Wearable Sensors. *IEEE COMMUNICATIONS SURVEYS AND TUTORIALS*, 15(3), 1192–1209.
- Le, J., Liao, X., & Yang, B. (2017). Full autonomy: A novel individualized anonymity model for privacy preserving. *Computers & Security*, 66, 204–217. <https://doi.org/10.1016/j.cose.2016.12.010>

- Lee, W., Cho, S., Chu, P., Vu, H., Helal, S., Song, W., ... Cho, K. (2016). Automatic agent generation for IoT-based smart house simulator. *Neurocomputing*, 209, 14–24. <https://doi.org/10.1016/j.neucom.2015.04.130>
- Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In *Machine Learning: ECML-98* (pp. 4–15). Springer, Berlin, Heidelberg. <https://doi.org/10.1007/BFb0026666>
- Li, Q., Meng, Q., Cai, J., Yoshino, H., & Mochida, A. (2009). Applying support vector machine to predict hourly cooling load in the building. *Applied Energy*, 86(10), 2249–2256.
- Li, Z., & Point, J. C. (2015). COAST: A Connected Open plAtform for Smart objecTs. In *2015 2nd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)* (pp. 166–172). <https://doi.org/10.1109/ICT-DM.2015.7402060>
- Lin, F., Song, C., Xu, X., Cavuoto, L., & Xu, W. (2017). Patient handling activity recognition through pressure-map manifold learning using a footwear sensor. *Smart Health*, 1–2, 77–92. <https://doi.org/10.1016/j.smhl.2017.04.005>
- Liu, H., Bolic, M., Nayak, A., & Stojmenovic, I. (2008). Taxonomy and challenges of the integration of RFID and wireless sensor networks. *IEEE Network*, 22(6).
- Lu, Z., Wang, N., Wu, J., & Qiu, M. (2017). IoTDeM: An IoT Big Data-oriented MapReduce performance prediction extended model in multiple edge clouds. *Journal of Parallel and Distributed Computing*. <https://doi.org/10.1016/j.jpdc.2017.11.001>
- Ma, Y., & Ghasemzadeh, H. (2016). An asynchronous multi-view learning approach for activity recognition using wearables. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 3105–3108). <https://doi.org/10.1109/EMBC.2016.7591386>
- Marangunić, N., & Granić, A. (2015). Technology acceptance model: a literature review from 1986 to 2013. *Universal Access in the Information Society*, 14(1), 81.
- Marheineke, M. (2016). *Designing Boundary Objects for Virtual Collaboration*. Springer.
- Mattern, F., & Floerkemeier, C. (2010). From the Internet of Computers to the Internet of Things. *From Active Data Management to Event-Based Systems and More*, 242–259.
- Mazzei, D., Fantoni, G., Montelisciani, G., & Baldi, G. (2014). Internet of Things for designing smart objects. In *2014 IEEE World Forum on*

- Internet of Things (WF-IoT)* (pp. 293–297). <https://doi.org/10.1109/WF-IoT.2014.6803175>
- Menicatti, R., Bruno, B., & Sgorbissa, A. (2017). Modelling the influence of cultural information on vision-based human home activity recognition. In *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)* (pp. 32–38). <https://doi.org/10.1109/URAI.2017.7992880>
- Meyers, R. J., Williams, E. D., & Matthews, H. S. (2010). Scoping the potential of monitoring and control technologies to reduce energy use in homes. *Energy & Buildings*, *42*, 563–569. <https://doi.org/10.1016/j.enbuild.2009.10.026>
- Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, *10*(7), 1497–1516. <https://doi.org/10.1016/j.adhoc.2012.02.016>
- Mobile and tablet internet usage exceeds desktop for first time worldwide. (2016, November 1). Retrieved October 16, 2017, from <http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide>
- Mozer, M. C. (1998). The neural network house: An environment that adapts to its inhabitants. In *Proc. AAAI Spring Symp. Intelligent Environments* (Vol. 58).
- Muheidat, F., & Tyrer, H. W. (2016). Can we make a carpet smart enough to detect falls? In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 5356–5359). <https://doi.org/10.1109/EMBC.2016.7591937>
- Mullainathan, S., & Spiess, J. (2017). Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives*, *31*(2), 87–106. <https://doi.org/10.1257/jep.31.2.87>
- Muralidharan, V., & Sugumaran, V. (2013). Selection of discrete wavelets for fault diagnosis of monoblock centrifugal pump using the J48 algorithm. *Applied Artificial Intelligence*, *27*(1), 1–19.
- Nair, L. R., Shetty, S. D., & Shetty, S. D. (2017). Applying spark based machine learning model on streaming big data for health status prediction. *Computers and Electrical Engineering*. <https://doi.org/10.1016/j.compeleceng.2017.03.009>
- Nasar, J. L., & Troyer, D. (2013). Pedestrian injuries due to mobile phone use in public places. *Accident Analysis & Prevention*, *57*(Supplement C), 91–95. <https://doi.org/10.1016/j.aap.2013.03.021>

- Nazari Shirehjini, A. a., & Semsar, A. (2017). Human interaction with IoT-based smart environments. *Multimedia Tools and Applications*, 76(11), 13343–13365. <https://doi.org/10.1007/s11042-016-3697-3>
- Nitti, M., Pilloni, V., Colistra, G., & Atzori, L. (2016). The Virtual Object as a Major Element of the Internet of Things: A Survey. *IEEE Communications Surveys Tutorials*, 18(2), 1228–1240. <https://doi.org/10.1109/COMST.2015.2498304>
- Norouzi, M., Collins, M. D., Johnson, M., Fleet, D. J., & Kohli, P. (2015). Efficient non-greedy optimization of decision trees. *ArXiv:1511.04056 [Cs]*. Retrieved from <http://arxiv.org/abs/1511.04056>
- Nuamah, J., & Seong, Y. (2017). Human machine interface in the Internet of Things (IoT). In *2017 12th System of Systems Engineering Conference (SoSE)* (pp. 1–6). <https://doi.org/10.1109/SYSOSE.2017.7994979>
- Number of mobile phone users worldwide 2013-2019. (n.d.). Retrieved October 17, 2017, from <https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/>
- Oliver, N., Horvitz, E., & Garg, A. (2002). Layered Representations for Human Activity Recognition. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces* (p. 3–). Washington, DC, USA: IEEE Computer Society. <https://doi.org/10.1109/ICMI.2002.1166960>
- Osuna, E., Freund, R., & Girosit, F. (1997). Training support vector machines: an application to face detection. In *Computer vision and pattern recognition, 1997. Proceedings., 1997 IEEE computer society conference on* (pp. 130–136). IEEE.
- Paing, T., Morroni, J., Dolgov, A., Shin, J., Brannan, J., Zane, R., & Popovic, Z. (2007). Wirelessly-Powered Wireless Sensor Platform. In *2007 European Conference on Wireless Technologies* (pp. 241–244). <https://doi.org/10.1109/ECWT.2007.4403991>
- Pal, M. (2005). Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1), 217–222.
- Palaniappan, S., & Awang, R. (2008). Intelligent heart disease prediction system using data mining techniques. In *Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on* (pp. 108–115). IEEE.
- Paradiso, J., Ablner, C., Hsiao, K., & Reynolds, M. (1997). The magic carpet: physical sensing for immersive environments. In *CHI'97 Extended Abstracts on Human Factors in Computing Systems* (pp. 277–278). ACM.

- Parasuraman, A. (2000). Technology Readiness Index (Tri) A Multiple-Item Scale to Measure Readiness to Embrace New Technologies. *Journal of Service Research*, 2(4), 307–320. <https://doi.org/10.1177/109467050024001>
- Park, S. K., Mukherjee, B., & Zhao, Z. (2017). Construction of environmental risk score beyond standard linear models using machine learning methods: application to metal mixtures, oxidative stress and cardiovascular disease in NHANES. *ENVIRONMENTAL HEALTH*, 16.
- Peppers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), 45–77.
- Pham, C., Diep, N. N., & Phuong, T. M. (2017). e-Shoes: Smart shoes for unobtrusive human activity recognition. In *2017 9th International Conference on Knowledge and Systems Engineering (KSE)* (pp. 269–274). <https://doi.org/10.1109/KSE.2017.8119470>
- Polatkan, G., Jafarpour, S., Brasoveanu, A., Hughes, S., & Daubechies, I. (2009). Detection of forgery in paintings using supervised learning. In *2009 16th IEEE International Conference on Image Processing (ICIP)* (pp. 2921–2924). <https://doi.org/10.1109/ICIP.2009.5413338>
- Quinlan, J. R., Morris, G., Jackson, J., & O'Connell, P. (1993). *C4.5: programs for machine learning*. San Mateo, California : Morgan Kaufmann Publishers, 1993.
- Rajasekaran, R. G., Manikandaraj, S., & Kamaleshwar, R. (2017). Implementation of Machine Learning Algorithm for predicting user behavior and smart energy management. In *2017 International Conference on Data Management, Analytics and Innovation (ICDMAI)* (pp. 24–30). <https://doi.org/10.1109/ICDMAI.2017.8073480>
- Rajput, A., Aharwal, R. P., Dubey, M., Saxena, S., & Raghuvanshi, M. (2011). J48 and JRIP rules for e-governance data. *International Journal of Computer Science and Security (IJCSS)*, 5(2), 201.
- Raun, N. F. (2016). Smart environment using internet of things(IOTS) - a review. In *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (pp. 1–6). <https://doi.org/10.1109/IEMCON.2016.7746313>
- Ren, J., Guo, H., Xu, C., & Zhang, Y. (2017). Serving at the Edge: A Scalable IoT Architecture Based on Transparent Computing. *IEEE Network*, 31(5), 96–105. <https://doi.org/10.1109/MNET.2017.1700030>
- Rodriguez-Galiano, V. F., Ghimire, B., Rogan, J., Chica-Olmo, M., & Rigol-Sanchez, J. P. (2012). An assessment of the effectiveness of a random

- forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67, 93–104.
- Rogers, Y. (2006). Moving on from Weiser's Vision of Calm Computing: Engaging UbiComp Experiences. In *UbiComp 2006: Ubiquitous Computing* (pp. 404–421). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/11853565\\_24](https://doi.org/10.1007/11853565_24)
- Rothberg, M. B., Arora, A., Hermann, J., Kleppel, R., Marie, P. S., & Visintainer, P. (2010). Phantom vibration syndrome among medical staff: a cross sectional survey. *BMJ*, 341, c6914. <https://doi.org/10.1136/bmj.c6914>
- Salim, F., & Haque, U. (2015). Urban computing in the wild: A survey on large scale participation and citizen engagement with ubiquitous computing, cyber physical systems, and Internet of Things. *International Journal of Human - Computer Studies*, 81, 31–48. <https://doi.org/10.1016/j.ijhcs.2015.03.003>
- Sample, A. P., Yeager, D. J., Smith, J. R., Powledge, P. S., & Mamishev, A. V. (2006). Energy Harvesting in RFID Systems. In *2006 International Conference on Actual Problems of Electron Devices Engineering* (pp. 445–449). <https://doi.org/10.1109/APEDE.2006.307453>
- Satyanarayanan, M. (2001). Pervasive computing: vision and challenges. *IEEE Personal Communications*, 8(4), 10–17. <https://doi.org/10.1109/98.943998>
- Savio, D., & Ludwig, T. (2007). Smart Carpet: A Footstep Tracking Interface. In *21st International Conference on Advanced Information Networking and Applications Workshops, 2007, AINAW '07* (Vol. 2, pp. 754–760). <https://doi.org/10.1109/AINAW.2007.338>
- Schatten, M., Ševa, J., & Tomičić, I. (2016). A roadmap for scalable agent organizations in the Internet of Everything. *The Journal of Systems & Software*, 115, 31–41. <https://doi.org/10.1016/j.jss.2016.01.022>
- Serra, R., Knittel, D., Croce, P. D., & Peres, R. (2016). Activity Recognition With Smart Polymer Floor Sensor: Application to Human Footstep Recognition. *IEEE Sensors Journal*, 16(14), 5757–5775. <https://doi.org/10.1109/JSEN.2016.2554360>
- Shin, D. (2014). A socio-technical framework for Internet-of-Things design: A human-centered design for the Internet of Things. *Telematics and Informatics*, 31, 519–531. <https://doi.org/10.1016/j.tele.2014.02.003>
- Simon, H. A. (1996). *The Sciences of the Artificial*. MIT Press.
- Song, Y., Yau, S. S., Yu, R., Zhang, X., & Xue, G. (2017). An Approach to QoS-based Task Distribution in Edge Computing Networks for IoT Applications. In *2017 IEEE International Conference on Edge*

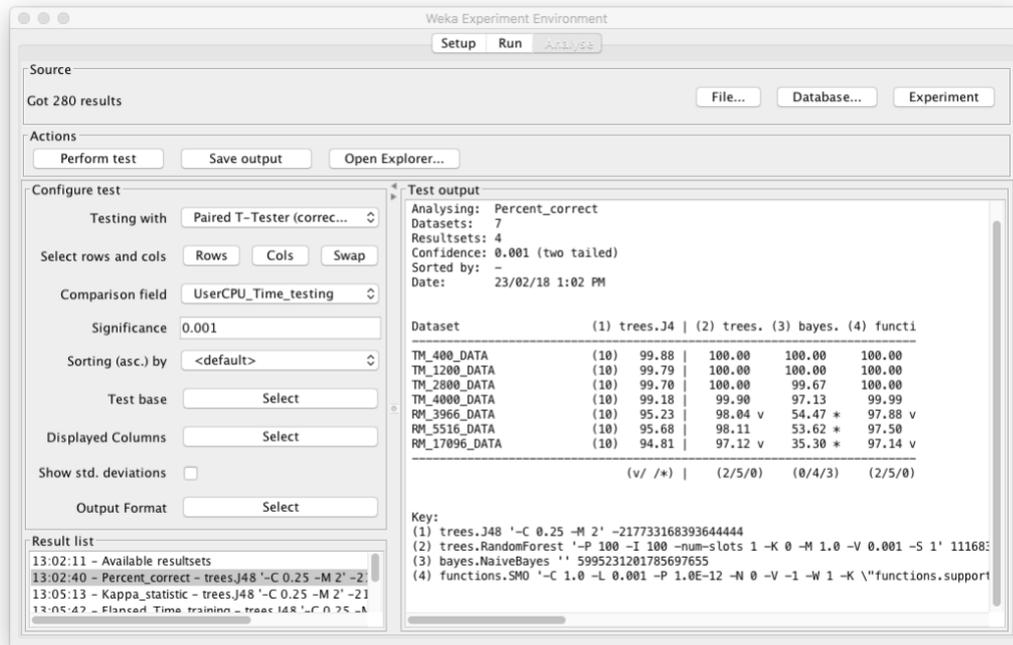
- Computing (EDGE)* (pp. 32–39).  
<https://doi.org/10.1109/IEEE.EDGE.2017.50>
- Spencer, G. A. (2016). Big Data: More than Just Big and More than Just Data. *Frontiers of Health Services Management*, 32(4), 27–33.
- Stecca, M., Moiso, C., Fornasa, M., Baglietto, P., & Maresca, M. (2015). A Platform for Smart Object Virtualization and Composition. *IEEE INTERNET OF THINGS JOURNAL*, 2(6), 604–613.
- Strategy, ITU & Unit, & Policy. (2005). ITU Internet Reports 2005: The internet of things. *Geneva: International Telecommunication Union (ITU)*.
- Strode, D. E., & Chard, S. M. (2014). A proposal for using design science in small-scale postgraduate research projects in information technology. In *2014 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)* (pp. 242–245).  
<https://doi.org/10.1109/TALE.2014.7062633>
- Sun, L., Zhang, D., Li, B., Guo, B., & Li, S. (2010). Activity Recognition on an Accelerometer Embedded Mobile Phone with Varying Positions and Orientations. In *Ubiquitous Intelligence and Computing* (pp. 548–562). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-16355-5\\_42](https://doi.org/10.1007/978-3-642-16355-5_42)
- Sung, J., Lopez, T. S., & Kim, D. (2007). The EPC sensor network for RFID and WSN integration infrastructure. In *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on* (pp. 618–621). IEEE.
- Sztyler, T., Stuckenschmidt, H., & Petrich, W. (2017). Position-aware activity recognition with wearable devices. *Pervasive and Mobile Computing*, 38(Part 2), 281–295.  
<https://doi.org/10.1016/j.pmcj.2017.01.008>
- Tanaka, O., Ryu, T., Hayashida, A., Moshnyaga, V. g., & Hashimoto, K. (2015). *A Smart Carpet Design for Monitoring People with Dementia* (Vol. 1089). Springer Verlag. [https://doi.org/10.1007/978-3-319-08422-0\\_92](https://doi.org/10.1007/978-3-319-08422-0_92)
- Tapia, E. M., Intille, S. S., & Larson, K. (2004). Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In *Pervasive Computing* (pp. 158–175). Springer, Berlin, Heidelberg.  
[https://doi.org/10.1007/978-3-540-24646-6\\_10](https://doi.org/10.1007/978-3-540-24646-6_10)
- Tolcha, Y., & Kim, D. (2016). Distributed event-based resource-oriented EPCglobal middleware using GPC. In *RFID Technology and Applications (RFID-TA), 2016 IEEE International Conference on* (pp. 29–34). IEEE.

- Trabelsi, D., Mohammed, S., Chamroukhi, F., Oukhellou, L., & Amirat, Y. (2013). An unsupervised approach for automatic activity recognition based on hidden Markov model regression. *IEEE Transactions on Automation Science and Engineering*, *10*(3), 829–835.
- Turaga, P., Chellappa, R., Subrahmanian, V. S., & Udreă, O. (2008). Machine Recognition of Human Activities: A Survey. *IEEE Transactions on Circuits and Systems for Video Technology*, *18*(11), 1473–1488. <https://doi.org/10.1109/TCSVT.2008.2005594>
- Uğur Yavuz, S., Bonetti, R., & Cohen, N. (2017). Designing The “Next” Smart Objects Together With Children. *Design Journal*, *20*, S3789–S3800. <https://doi.org/10.1080/14606925.2017.1352882>
- Vaishnavi, V., & Kuechler, W. (2015). *Design Science Research Methods and Patterns : Innovating Information and Communication Technology* (Vol. Second edition). Boca Raton: CRC Press.
- van Kranenburg, R. (2008). *A critique of ambient technology and the all-seeing network of RFID*. Insitute of Network Cultures.
- Vaughan, J., Green, P., Salter, M., Grieve, B., & Ozanyan, K. (2017). *Floor Sensors of Animal Weight and Gait for Precision Livestock Farming*.
- Velloso, E., Schmidt, D., Alexander, J., Gellersen, H., & Bulling, A. (2015). The Feet in Human--Computer Interaction: A Survey of Foot-Based Interaction. *ACM Computing Surveys*, *48*(2), 21:1-21:35. <https://doi.org/10.1145/2816455>
- Venable, J., Pries-Heje, J., & Baskerville, R. (2014). FEDS: a Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, *25*(1), 77–89. <https://doi.org/10.1057/ejis.2014.36>
- Verame, J. K. M., Costanza, E., & Ramchurn, S. D. (2016). The Effect of Displaying System Confidence Information on the Usage of Autonomous Systems for Non-specialist Applications: A Lab Study. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 4908–4920). New York, NY, USA: ACM. <https://doi.org/10.1145/2858036.2858369>
- Vrigkas, M., Nikou, C., & Kakadiaris, I. A. (2015). A Review of Human Activity Recognition Methods. *Frontiers in Robotics and AI*, *2*. <https://doi.org/10.3389/frobt.2015.00028>
- Wang, C., Bi, Z., & Xu, L. D. (2014). IoT and Cloud Computing in Automation of Assembly Modeling Systems. *IEEE Transactions on Industrial Informatics*, *10*(2), 1426–1434. <https://doi.org/10.1109/TII.2014.2300346>
- Wang, J., Chen, Y., Hao, S., Peng, X., & Hu, L. (2017). Deep Learning for Sensor-based Activity Recognition: A Survey.

- Want, R. (2006). An introduction to RFID technology. *IEEE Pervasive Computing*, 5(1), 25–33. <https://doi.org/10.1109/MPRV.2006.2>
- Weber, R. M. (2016). Internet of Things Becomes Next Big Thing. *Journal of Financial Service Professionals*, 70(6), 43–46.
- Weiser, M. (1991). The Computer for the 21st Century. *Scientific American*, 265(3), 94.
- Weiser, M., Gold, R., & Brown, J. S. (1999). The origins of ubiquitous computing research at PARC in the late 1980s. *IBM Systems Journal*, 38(4), 693–696. <https://doi.org/10.1147/sj.384.0693>
- Welbourne, E., Battle, L., Cole, G., Gould, K., Rector, K., Raymer, S., ... Borriello, G. (2009). Building the Internet of Things Using RFID: The RFID Ecosystem Experience. *IEEE Internet Computing*, 13(3), 48–55. <https://doi.org/10.1109/MIC.2009.52>
- What Is Big Data? - Gartner IT Glossary - Big Data. (n.d.). Retrieved December 4, 2017, from <https://www.gartner.com/it-glossary/big-data/>
- Widodo, A., & Yang, B.-S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21(6), 2560–2574.
- Williams, J. L. (2016). Privacy in the age of the Internet of things. *Human Rights*, (4), 14.
- Wilson, C., Hargreaves, T., & Hauxwell-Baldwin, R. (2017). Benefits and risks of smart home technologies. *Energy Policy*, 103(Supplement C), 72–83. <https://doi.org/10.1016/j.enpol.2016.12.047>
- Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.
- Wu, C. L., Tseng, Y. S., & Fu, L. C. (2013). Spatio-temporal Feature Enhanced Semi-supervised Adaptation for Activity Recognition in IoT-Based Context-Aware Smart Homes. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing* (pp. 460–467). <https://doi.org/10.1109/GreenCom-iThings-CPSCom.2013.94>
- Y.2060 : Overview of the Internet of things. (n.d.). Retrieved October 22, 2017, from <https://www.itu.int/rec/T-REC-Y.2060-201206-I>
- Yan-e, D. (2011). Design of Intelligent Agriculture Management Information System Based on IoT. In *2011 International Conference on Intelligent Computation Technology and Automation (ICICTA)* (Vol. 1, pp. 1045–1049). <https://doi.org/10.1109/ICICTA.2011.262>

- Yu, C. R., Wu, C. L., Lu, C. H., & Fu, L. C. (2006). Human Localization via Multi-Cameras and Floor Sensors in Smart Home. In *2006 IEEE International Conference on Systems, Man and Cybernetics* (Vol. 5, pp. 3822–3827). <https://doi.org/10.1109/ICSMC.2006.384726>
- Zarkami, R. (2011). Application of classification trees-J48 to model the presence of roach (*Rutilus rutilus*) in rivers. *Caspian Journal of Environmental Sciences*, *9*(2), 189–198.

## 8 APPENDIX A: WEKA RESULTS



```

Scheme:      weka.classifiers.trees.J48 -U -M 2
Relation:    TM_2800_DATA
Instances:   2800
Attributes:  3970
              [list of attributes omitted]
Test mode:   split 60.0% train, remainder test
  
```

=== Classifier model (full training set) ===

J48 unpruned tree

```

node1144 <= 0
|
| node846 <= 0
| |
| | node388 <= 0: STANDING (350.0)
| | node388 > 0: AWAY (350.0)
| |
| | node846 > 0
| | |
| | | node2311 <= 0: RELAXED (350.0)
| | | node2311 > 0: NEUTRAL (350.0)
| |
| | node1144 > 0
| | |
| | | node3324 <= 217: LEFT (350.0)
| | | node3324 > 217
| | | |
| | | | node2610 <= 203
| | | | |
| | | | | node3854 <= 201: TYPING (350.0)
| | | | | node3854 > 201: MOUSE (350.0)
| | | |
| | | | node2610 > 203: RIGHT (350.0)
  
```

```

Number of Leaves :    8
Size of the tree :   15
  
```

=== Confusion Matrix ===

	a	b	c	d	e	f	g	h	<-- classified as
1407	22	117	144	174	0	44	0	0	a = NEUTRAL
42	1088	93	28	43	70	7	0	0	b = RELAXED
0	0	906	127	0	0	0	0	0	c = TYPING
220	7	444	1445	51	14	0	0	0	d = MOUSE
1	2	2	2	81	0	0	0	0	e = LEFT
0	0	0	0	0	35	0	0	0	f = RIGHT
0	0	0	1	0	0	69	11	0	g = STANDING
0	0	0	0	0	0	2	139	0	h = AWAY

```

Scheme:      weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:    TM_2800_DATA
Instances:   2800
Attributes:  3970
              [list of attributes omitted]
Test mode:   split 60.0% train, remainder test
  
```

=== Classifier model (full training set) ===

J48 pruned tree

```

node1144 <= 0
|
| node846 <= 0
| |
| | node388 <= 0: STANDING (350.0)
| | node388 > 0: AWAY (350.0)
| |
| | node846 > 0
| | |
| | | node2311 <= 0: RELAXED (350.0)
| | | node2311 > 0: NEUTRAL (350.0)
| |
| | node1144 > 0
| | |
| | | node3324 <= 217: LEFT (350.0)
| | | node3324 > 217
| | | |
| | | | node2610 <= 203
| | | | |
| | | | | node3854 <= 201: TYPING (350.0)
| | | | | node3854 > 201: MOUSE (350.0)
| | | |
| | | | node2610 > 203: RIGHT (350.0)
  
```

```

Number of Leaves :    8
Size of the tree :   15
  
```

```
node544 > 205
| node436 <= 699: STANDING (23.0)
| | node436 > 699
| | | node305 <= 0
| | | | node684 <= 389: AWAY (282.0)
| | | | node684 > 389
| | | | | node482 <= 0: AWAY (4.0)
| | | | | node482 > 0: STANDING (6.0/1.0)
| | | node305 > 0: STANDING (4.0)
```

Number of Leaves : 316  
Size of the tree : 631

Time taken to build model: 319.33 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.11 seconds

=== Summary ===

Correctly Classified Instances	6430	94.0333 %
Incorrectly Classified Instances	408	5.9667 %
Kappa statistic	0.9211	
Mean absolute error	0.0157	
Root mean squared error	0.1203	
Relative absolute error	8.317 %	
Root relative squared error	39.1038 %	
Total Number of Instances	6838	

Number of Leaves : 12  
Size of the tree : 23

Time taken to build model: 7.75 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.41 seconds

=== Summary ===

Correctly Classified Instances	1582	98.875 %
Incorrectly Classified Instances	18	1.125 %
Kappa statistic	0.9871	
Mean absolute error	0.0032	
Root mean squared error	0.0531	
Relative absolute error	1.4592 %	
Root relative squared error	16.0556 %	
Total Number of Instances	1600	

```
node814 > 244
| node967 <= 1065
| | node1068 <= 344: STANDING (36.0)
| | | node1068 > 344
| | | | node1156 <= 0: AWAY (50.0/1.0)
| | | | node1156 > 0: STANDING (8.0)
| | node967 > 1065: NEUTRAL (7.0)
```

Number of Leaves : 54  
Size of the tree : 107

Time taken to build model: 22.42 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.43 seconds

=== Summary ===

Correctly Classified Instances	1974	99.4458 %
Incorrectly Classified Instances	11	0.5542 %
Kappa statistic	0.9907	
Mean absolute error	0.0027	
Root mean squared error	0.0369	
Relative absolute error	1.8268 %	
Root relative squared error	13.5596 %	
Total Number of Instances	1985	

```
node544 > 205
| node436 <= 699: STANDING (23.0)
| | node436 > 699
| | | node305 <= 0
| | | | node684 <= 389: AWAY (282.0)
| | | | node684 > 389
| | | | | node482 <= 0: AWAY (4.0)
| | | | | node482 > 0: STANDING (6.0/1.0)
| | | node305 > 0: STANDING (4.0)
```

Number of Leaves : 288  
Size of the tree : 575

Time taken to build model: 318.43 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.11 seconds

=== Summary ===

Correctly Classified Instances	6435	94.1065 %
Incorrectly Classified Instances	403	5.8935 %
Kappa statistic	0.9221	
Mean absolute error	0.0161	
Root mean squared error	0.1195	
Relative absolute error	8.4906 %	
Root relative squared error	38.8699 %	
Total Number of Instances	6838	

Number of Leaves : 10  
Size of the tree : 19

Time taken to build model: 6.83 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.39 seconds

=== Summary ===

Correctly Classified Instances	1586	99.125 %
Incorrectly Classified Instances	14	0.875 %
Kappa statistic	0.99	
Mean absolute error	0.0028	
Root mean squared error	0.0463	
Relative absolute error	1.2947 %	
Root relative squared error	14.0061 %	
Total Number of Instances	1600	

```
node814 > 244
| node967 <= 1065
| | node1068 <= 344: STANDING (36.0)
| | | node1068 > 344
| | | | node1156 <= 0: AWAY (50.0/1.0)
| | | | node1156 > 0: STANDING (8.0)
| | node967 > 1065: NEUTRAL (7.0)
```

Number of Leaves : 34  
Size of the tree : 67

Time taken to build model: 24.48 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.42 seconds

=== Summary ===

Correctly Classified Instances	1933	97.3804 %
Incorrectly Classified Instances	52	2.6196 %
Kappa statistic	0.9554	
Mean absolute error	0.0122	
Root mean squared error	0.0781	
Relative absolute error	8.2134 %	
Root relative squared error	28.6539 %	
Total Number of Instances	1985	

## Appendix A: Weka Results

```

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-c
Number of kernel evaluations: 3224 (92.244% cached)

Time taken to build model: 6.62 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.52 seconds

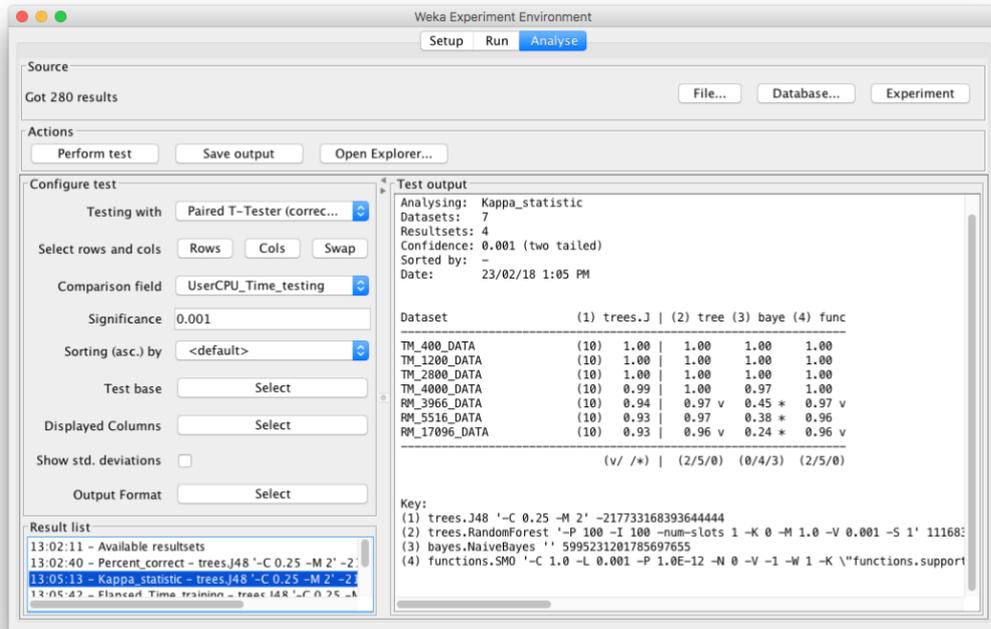
=== Summary ===
Correctly Classified Instances      1985      100 %
Incorrectly Classified Instances    0          0 %
Kappa statistic                    1
Mean absolute error                0.0058
Root mean squared error            0.0227
Relative absolute error             3.9048 %
Root relative squared error        8.329 %
Total Number of Instances          1985

Time taken to build model: 10.97 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.66 seconds

=== Summary ===
Correctly Classified Instances      1975      99.4962 %
Incorrectly Classified Instances    10        0.5038 %
Kappa statistic                    0.9915
Mean absolute error                0.1875
Root mean squared error            0.2913
Relative absolute error            125.9944 %
Root relative squared error        106.912 %
Total Number of Instances          1985
    
```



```

Time taken to build model: 4.28 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 3.43 seconds

=== Summary ===
Correctly Classified Instances      1262      79.5712 %
Incorrectly Classified Instances    324      20.4288 %
Kappa statistic                    0.7476
Mean absolute error                0.0512
Root mean squared error            0.2255
Relative absolute error            26.2666 %
Root relative squared error        72.3535 %
Total Number of Instances          1586

Time taken to build model: 33.13 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 15.18 seconds

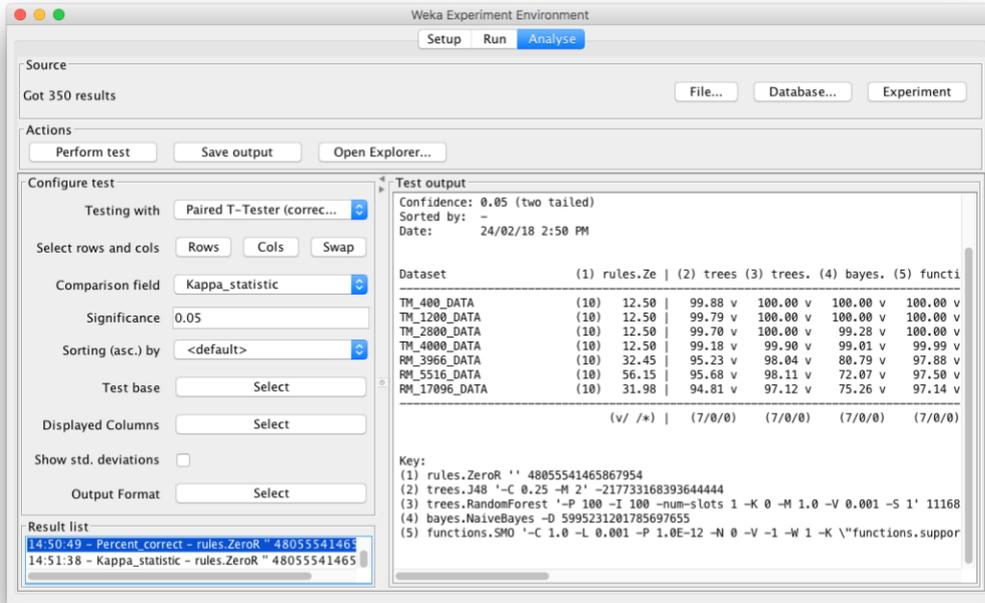
=== Summary ===
Correctly Classified Instances      5170      75.6069 %
Incorrectly Classified Instances   1668      24.3931 %
Kappa statistic                    0.6879
Mean absolute error                0.061
Root mean squared error            0.2466
Relative absolute error            32.246 %
Root relative squared error        80.1694 %
Total Number of Instances          6838
    
```

Dataset	(1) trees.J	(2) tree	(3) baye	(4) func
TM_400_DATA	(10) 0.00	0.00	0.00	0.00
TM_1200_DATA	(10) 0.01	0.00	0.00	0.00
TM_2800_DATA	(10) 0.00	0.00	0.01	0.00
TM_4000_DATA	(10) 0.00	0.00	0.00	0.00
RM_3966_DATA	(10) 0.06	0.02 *	0.37 v	0.02
RM_5516_DATA	(10) 0.03	0.01	0.30 v	0.02
RM_17096_DATA	(10) 0.03	0.01 *	0.25 v	0.01

(v/ /\*) | (0/5/2) (3/4/0) (0/7/0)

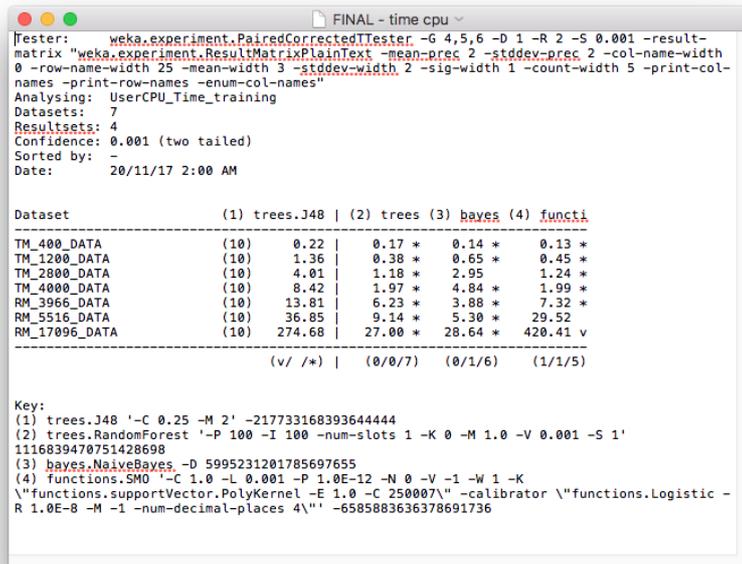
Dataset	(1) trees.J	(2) tree	(3) baye	(4) func
TM_400_DATA	(10) 0.00	0.00	0.00	0.00
TM_1200_DATA	(10) 0.00	0.00	0.00	0.00
TM_2800_DATA	(10) 0.00	0.00	0.00	0.00
TM_4000_DATA	(10) 0.00	0.00	0.00	0.00
RM_3966_DATA	(10) 0.02	0.01	0.04	0.01
RM_5516_DATA	(10) 0.04	0.02 *	0.10	0.02
RM_17096_DATA	(10) 0.01	0.01	0.05 v	0.01

(v/ /\*) | (0/6/1) (1/6/0) (0/7/0)



Dataset	(1) trees.J4	(2) trees	(3) bayes	(4) functi
TM_400_DATA	(10) 99.50	100.00	100.00	100.00
TM_1200_DATA	(10) 99.83	100.00	100.00	100.00
TM_2800_DATA	(10) 99.93	100.00	99.36	100.00
TM_4000_DATA	(10) 99.57	99.98	99.02	100.00
RM_3966_DATA	(10) 96.09	98.39 v	82.98 *	98.31 v
RM_5516_DATA	(10) 96.19	98.71 v	72.95 *	98.01
RM_17096_DATA	(10) 95.98	97.64 v	75.95 *	97.44 v

Summary statistics: (v/ /\*) | (3/4/0) (0/4/3) (2/5/0)



```

SVC options results
Tester: weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-
matrix "weka.experiment.ResultMatrixPlainText -mean-prec 2 -stddev-prec 2 -col-name-width
0 -row-name-width 25 -mean-width 3 -stddev-width 2 -sig-width 1 -count-width 5 -print-col-
names -print-row-names -enum-col-names"
Analysing: Elapsed_Time_testing
Datasets: 1
Resultsets: 4
Confidence: 0.05 (two tailed)
Sorted by: -
Date: 25/02/18 5:24 PM

Dataset (1) funcio | (2) func. (3) func. (4) funct
-----
RM_5516_DATA (4) 0.27 | 0.14 0.13 28.36 v
-----
(v/ /*) | (0/1/0) (0/1/0) (1/0/0)

Key:
(1) functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K
\"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -calibrator \"functions.Logistic -
R 1.0E-8 -M -1 -num-decimal-places 4\" -6585883636378691736
(2) functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 1 -V -1 -W 1 -K
\"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -calibrator \"functions.Logistic -
R 1.0E-8 -M -1 -num-decimal-places 4\" -6585883636378691736
(3) functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 2 -V -1 -W 1 -K
\"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -calibrator \"functions.Logistic -
R 1.0E-8 -M -1 -num-decimal-places 4\" -6585883636378691736
(4) functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 2 -V -1 -W 1 -K
\"functions.supportVector.NormalizedPolyKernel -E 2.0 -C 250007\" -calibrator
\"functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4\" -6585883636378691736
    
```

```

ewe
Tester: weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-
matrix "weka.experiment.ResultMatrixPlainText -mean-prec 2 -stddev-prec 2 -col-name-width
0 -row-name-width 25 -mean-width 3 -stddev-width 2 -sig-width 1 -count-width 5 -print-col-
names -print-row-names -enum-col-names"
Analysing: Elapsed_Time_training
Datasets: 1
Resultsets: 4
Confidence: 0.05 (two tailed)
Sorted by: -
Date: 25/02/18 8:16 PM

Dataset (1) function | (2) funct (3) funct (4) functi
-----
RM_5516_DATA (4) 23.96 | 23.41 25.44 104.75 v
-----
(v/ /*) | (0/1/0) (0/1/0) (1/0/0)

Key:
(1) functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K
\"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -calibrator \"functions.Logistic -
R 1.0E-8 -M -1 -num-decimal-places 4\" -6585883636378691736
(2) functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 1 -V -1 -W 1 -K
\"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -calibrator \"functions.Logistic -
R 1.0E-8 -M -1 -num-decimal-places 4\" -6585883636378691736
(3) functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 2 -V -1 -W 1 -K
\"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -calibrator \"functions.Logistic -
R 1.0E-8 -M -1 -num-decimal-places 4\" -6585883636378691736
(4) functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 2 -V -1 -W 1 -K
    
```

```

FINAL - time cpu testing
Tester: weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.001 -result-
matrix "weka.experiment.ResultMatrixPlainText -mean-prec 2 -stddev-prec 2 -col-name-width
0 -row-name-width 25 -mean-width 3 -stddev-width 2 -sig-width 1 -count-width 5 -print-col-
names -print-row-names -enum-col-names"
Analysing: UserCPU_Time_testing
Datasets: 7
Resultsets: 4
Confidence: 0.001 (two tailed)
Sorted by: -
Date: 20/11/17 2:00 AM

Dataset (1) trees.J | (2) tree (3) baye. (4) func
-----
TM_400_DATA (10) 0.00 | 0.00 v 0.09 v 0.00 v
TM_1200_DATA (10) 0.00 | 0.00 v 0.26 v 0.02 v
TM_2800_DATA (10) 0.00 | 0.01 v 0.62 v 0.04 v
TM_4000_DATA (10) 0.00 | 0.01 v 0.89 v 0.07 v
RM_3966_DATA (10) 0.00 | 0.03 v 0.88 v 0.05 v
RM_5516_DATA (10) 0.00 | 0.04 v 1.22 v 0.07 v
RM_17096_DATA (10) 0.01 | 0.10 v 3.87 v 0.36 v
-----
(v/ /*) | (7/0/0) (7/0/0) (7/0/0)

Key:
(1) trees.J48 '-C 0.25 -M 2' -21773316839364444
(2) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1'
1116839470751428698
(3) bayes.NaiveBayes -D 5995231201785697655
    
```

```

FINAL - serialized model size
Tester: weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.001 -result-
matrix "weka.experiment.ResultMatrixPlainText -mean-prec 2 -stddev-prec 2 -col-name-width
0 -row-name-width 25 -mean-width 2 -stddev-width 2 -sig-width 1 -count-width 5 -print-col-
names -print-row-names -enum-col-names"
Analysing: Serialized_Model_Size
Datasets: 7
Resultsets: 4
Confidence: 0.001 (two tailed)
Sorted by: -
Date: 20/11/18 1:56 AM

Dataset (1) trees.J48 '- | (2) trees.Rando (3) bayes.Naiv (4) functions.
-----
TM_400_DATA (10) 204565.00 | 355692.70 v 2987017.50 v 856795.20 v
TM_1200_DATA (10) 204566.00 | 494114.90 v 3093029.30 v 803025.00 v
TM_2800_DATA (10) 204566.00 | 801807.90 v 3548124.30 v 881950.20 v
TM_4000_DATA (10) 207102.80 | 1026173.30 v 3821126.50 v 980416.20 v
RM_3966_DATA (10) 240443.60 | 5199984.20 v 2881481.20 v 707916.20 v
RM_5516_DATA (10) 247752.00 | 6359232.30 v 2796980.00 v 657036.20 v
RM_17096_DATA (10) 364325.00 | 17227015.20 v 4526798.90 v 1090934.40 v
-----
(v/ /*) | (7/0/0) (7/0/0) (7/0/0)

Key:
(1) trees.J48 '-C 0.25 -M 2' -21773316839364444
(2) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1'
1116839470751428698
(3) bayes.NaiveBayes -D 5995231201785697655
    
```