Occupant-Centric Energy Management for Small Commercial Buildings

R E ANDE

PhD 2020

Occupant-Centric Energy Management for Small Commercial Buildings

RUTH ANDE

A thesis submitted in partial fulfilment of the requirements of
Manchester Metropolitan University
for the degree of Doctor of Philosophy

Department of Engineering
Manchester Metropolitan University

2020

Declaration of Authorship

I, Ruth ANDE, declare that this thesis titled, "Occupant-Centric Energy Management for Small Commercial Buildings" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date:

_____

Abstract

As the UK strives to reduce its impact on the environment, small and medium sized enterprises (SMEs) face significant energy reduction barriers which include high costs, the lack of expertise and significant time limitations. Many energy management systems (EMS) do exist but they are largely inaccessible to SMEs because they generally fit into three categories: being complex and expensive; affordable but requiring expertise to fit and manage; or affordable but overly simple and ineffective. Therefore, this thesis focuses on the development of on an holistic occupant-centric EMS to overcome the limitations of existing solutions to enable SMEs to overcome the barriers they have experienced. The principle of the occupant-centric EMS is to improve the temporal match between building occupants and energy consuming systems. To meet this principle, a number of enabling technologies are utilised including, Internet of Things (IoT), wireless sensor networks (WSN) and machine learning (ML).

The major contributions of this work include the development of

- a WSN simulation tool
- a methodology to analyse different network deployment techniques
- creation of a large labelled multimodal data set
- a single mode and multimode ML architecture which is designed and deployed on a constrained edge-based system to utilise binary classification to determine occupancy
- a holistic low cost occupant-centric EMS which automates a significant reduction of energy consumption within small commercial buildings

A number of node placement algorithms are developed to assess existing WSN deployment techniques that are utilised for unobtrusive, privacy protecting IoT data capture. The most suitable technique is determined to be the sensor grid which uses 44% of the hardware of other deployments and demonstrates an accuracy of 81% for occupancy monitoring.

To further improve the performance of occupancy monitoring, an edge-based ML model which analyses thermal image data is designed and

implemented demonstrating more than 96% accuracy in an office environment. To improve the performance in a wider range of environments, the ML model is extended to enable simultaneous analysis of the IoT multimodal building data. This model achieves the same performance in the office but demonstrates a 15% improvement in sensitivity and 31% in precision in another environment. The utilisation of additional low cost sensors and data fusion techniques enable an increase in building coverage from 78% to 100%, whilst maintaining the quantity of IoT nodes. The completed developed occupant-centric IoT-based EMS costs less than a fifth of existing comparable systems. The experimental evaluation results demonstrate more than 10% reduction in total building energy consumption whilst maintaining a comfortable working environment.

Acknowledgements

First and foremost I want to give all glory to God, without whom I would not have made it here. You have given me the knowledge and the strength to make this journey, so I want to praise and thank you.

I am extremely grateful to my supervisors, Prof. Bamidele Adebisi and Prof. Mohammad Hammoudeh for their invaluable advice, continuous support and patience throughout my studies. Prof. Adebisi, thank you for believing in me and pushing me. Prof. Hammoudeh, thank you for the many occasions you sat and listened to my ideas before helping me to find a focus.

I would like to thank RAA Technology Solutions Limited for funding this research. Also, thank you to the team at RAA, particularly Segun Popoola and Fatai Kassim who were influential in shaping my research methods and critiquing my results. I would like to thank my university colleagues in JDE033 for the time spent together in the office and our many lunches! Thank you to all my friends and family who offered your support by lending an ear, sending messages of well wishes, sweet treats and food deliveries.

To my incredible family,

Akin, Tosin and Dara

I want to thank you so much. You all sacrificed so much to enable me to reach this point and I could not have done it without your support.

Akin, I want to say thank you for your constant belief in me, daily encouragement and immeasurable patience as you carried so much on my behalf. I love you so much!

Tosin and Dara, thank you for bearing with me when I had no energy left. Thank you for your regular excitement when I tried to share the mysteries of my research with you. Thank you for all of the weekends and evenings that you let me study. I love you both to the moon and back!

Contents

Abstract **ii**

Acknowledgements **iv**

List of Figures **x**

List of Tables **xiv**

1 Introduction **1**
1.1 Research Motivation 1
1.2 Research Aims 2
1.3 Research Objectives 2
1.4 Key Contributions 3
1.5 Preliminaries 4
1.5.1 Internet of Things 4
1.5.2 Energy Consumption and Climate Change 5
1.6 Thesis Organisation 6
1.7 List of Publications 7
1.7.1 Book Chapters 7
1.7.2 Journal Papers 8
1.7.3 Conference Papers 9

2 Concepts and Literature Review **10**
2.1 Introduction 10
2.2 Building Monitoring 10
2.2.1 Occupant-centric EMS 12
2.3 Energy Management in Small Commercial Buildings 13
2.3.1 Small Commercial Building User Requirements 14
2.3.2 Energy Efficiency Barriers for SMEs 16
2.3.3 Existing Research on SME Energy Efficiency 17
2.3.4 Existing Commercial Solutions for SME Energy Management 18

2.4 Energy Management Enabling Technologies 19
2.4.1 Wireless Sensor Networks 19
2.4.1.1 Existing WSN Deployment Techniques 20
2.4.2 Machine Learning 21
2.4.2.1 Deep Learning Techniques 22
2.4.2.2 Deep Neural Network for Occupancy Detection 23
2.4.2.3 Existing IoT-based Occupancy Detection with Applied ML 24
2.5 Smart Building Mapping and Simulation Tools 27
2.5.1 Existing WSN Mapping and Simulation Tools 28
2.5.2 Proposed WSN Mapping and Simulation Tool 29
2.6 Chapter Summary 29

3 MIoTs **31**
3.1 Overview of Smart Building Mapping and Simulation Tool . . 31
3.2 MIoTs User Interface 33
3.3 MIoTs Sensors 36
3.4 MIoTs WSN deployments 37
3.5 Methodology to Evaluate MIoTs Simulator 37
3.6 Evaluation of MIoTs Simulator 40
3.6.1 Position of the Nodes in the Network 40
3.6.2 Correlation Between the Real and Simulated Data . . . 42
3.6.3 Performance of Real and Simulated Data 45
3.7 Chapter Summary 46

4 WSN Deployments for Building Monitoring **48**
4.1 Introduction 48
4.2 Preliminaries of WSN Deployment Techniques 49
4.2.1 Strong and Weak Coverage 49
4.2.2 Existing Physical Deployment Techniques 50
4.3 Applying Packing Techniques to Develop Strong Coverage WSN Deployment Techniques 53
4.4 Node Density 54
4.4.1 Sensor Grid Node Density 54
4.4.2 Sensor Perimeters Node Density 55
4.4.3 Sensor Hexagons Node Density 61
4.5 Space Coverage 62
4.5.1 Sensor Grid Coverage 63
4.5.2 Sensor Hexagons Coverage 64

4.5.3 Node Density of Sensor Grid Versus Sensor Hexagons 65
4.6 Overlapping Node Coverage Deployment Techniques 67
4.6.1 Overlapping Sensor Grid 67
4.6.2 Overlapping Sensor Perimeters 68
4.6.3 Overlapping Sensor Hexagons 69
4.7 Evaluation of WSN Deployment Techniques 70
4.7.1 Evaluation 1: Node density, coverage and overlap . . . 72
4.7.2 Evaluation 2: Data collection 74
4.7.2.1 Evaluation of PIRS data 74
4.7.2.2 Evaluation of CO_2S data 74
4.7.2.3 Evaluation of IRS data 75
4.8 Chapter Summary 77

5 A DNN for Edge-based Occupancy Detection 79
5.1 Introduction 79
5.2 DeNNOTE Architecture 81
5.3 Process of Occupancy Classification 82
5.3.1 System Hardware and Software 83
5.3.2 Data Capture 84
5.3.2.1 Monitored environment 84
5.3.2.2 Data capture method 84
5.3.2.3 Data set creation 85
5.3.3 Data Preprocessing 86
5.3.4 Optimisation of DNN Model 86
5.4 Performance Evaluation of DNN Model 87
5.4.1 Evaluation of Data Sets 88
5.4.2 Evaluation of DNN's Generalisation Ability 89
5.4.3 Evaluation of DNN's Hyperparameters 91
5.5 Deployment of DNN using WSN Techniques 93
5.5.1 Evaluation of Deployment of Nodes with Separate DNN 95
5.5.2 Evaluation of Deployment of Nodes with Integrated DNN 100
5.6 Chapter Summary 104

6 IoT Node Optimisation 106
6.1 Introduction 106
6.2 Reconfiguration of Node Sensors 106
6.3 Multimode DNN 108
6.3.1 DNN-2: Single-branch Multimode DNN Architecture . 109

6.3.2 DNN-3: Multi-branch Multimode DNN Architecture . 111
6.4 Evaluation of Reconfigured Node Sensors 115
6.4.1 Node Sensors: Single PIRS 115
6.4.2 Node Sensors: Single IRS 116
6.4.3 Node Sensors: Combine Single PIRS and Single IRS . . 117
6.4.4 Node Sensors: Combine Single PIRS and Two IRS . . . 119
6.4.5 Node Sensors: Combine Four PIRS and Two IRS 120
6.5 WSN Deployments to Evaluate the Reconfigured Node Sensors 123
6.6 Evaluation of the Multimode DNNs 125
6.7 Chapter Summary 130

7 Occupant-centric IoT-based Energy Management System 131
7.1 Introduction 131
7.2 OcCEMS Overview 132
7.3 OpenEMS Integration Platform 134
7.4 Energy Consumption Baseline 135
7.4.1 Baseline Model 136
7.5 Range of DeNNOTE Nodes 138
7.6 Deployment of OcCEMS 139
7.7 Evaluation of Baseline Model 140
7.8 Evaluation of OcCEMS Deployment 142
7.8.1 Evaluation of OcCEMS Hardware Cost 144
7.8.2 Evaluation of OcCEMS Simulated Performance 145
7.9 Chapter Summary 147

8 Conclusion and Recommendations 149
8.1 Conclusion 149
8.2 Limitations of Work 151
8.3 Further Work 152

A Appendix 153
A.1 Published Work 153
A.1.1 Journal Paper 153
A.1.2 Conference Paper 198

Bibliography 207

List of Figures

1.1 Conceptual organisation of Thesis 8

2.1 Energy consumption based enterprise size and building size . 15
2.2 Components of Wireless Sensor Node 20

3.1 Process of designing and evaluating simulation tool 32
3.2 MIoTs Code: Defining building layout 34
3.3 MIoTs Code: Defining sensor network 34
3.4 MIoTs Code: User defined parameters 35
3.5 Default sensor coverage volume 36
3.6 Rectangular IRS coverage volume 36
3.7 PIRS output file 37
3.8 CO_2S output file 37
3.9 Ground-truth room data output file 37
3.10 PIRS deployment 38
3.11 CO_2S deployment 38
3.12 IRS deployment 38
3.13 Combined deployment 38
3.14 Room 1: simulated CO_2 data 41
3.15 Office: real CO_2 data 41
3.16 Room 2: simulated CO_2 data 42
3.17 Kitchen: real CO_2 data 42
3.18 Room 3: simulated CO_2 data 43
3.19 Meeting room: real CO_2 data 43
3.20 Simulated negative image 45
3.21 Real negative image 45
3.22 Simulated negative image 45
3.23 Real negative image 45
3.24 Simulated positive image 45
3.25 Real positive image 45

4.1 2D sensor coverage area 51

4.2 Overview of six proposed sensor network deployment techniques . . . 51
4.3 Node coverage based on being positioned on building ceiling 52
4.4 Pictorial view of sensor perimeter deployment in different polygon shaped spaces . . . 58
4.5 Spacing of nodes in sensor hexagons . . . 61
4.6 Spacing of nodes in sensor hexagons . . . 64
4.7 Spacing of nodes in overlapping sensor grid and sensor hexagons deployments . . . 67
4.8 Horizontal view of 4 WSN deployments illustrating regions of node coverage . . . 72
4.9 Graphs showing the detected CO_2 level and the number of occupants for each WSN deployment . . . 75
4.10 Graphs showing the actual number of occupants compared with the detected number for each WSN deployment . . . 77

5.1 System hardware . . . 83
5.2 Overview of the DeNNOTE occupancy detection process . . . 84
5.3 Comparison of model performance achieved with Data.i1, i2, i3 and Data.i5 . . . 89
5.4 Test performance of DNN based on 2 training epochs and varied activation functions using office test data . . . 94
5.5 Test performance of DNN based on 2 training epochs and varied activation functions and meeting room test data . . . 94
5.6 Test performance of DNN based on 2 training epochs and varied activation functions and kitchen test data . . . 95
5.7 WSN architecture showing separate data capture and data processing . . . 96
5.8 Simulation process illustrating creation, capture, fusion and processing of sensor data based on separate data capture and processing . . . 97
5.9 Room maps (top) and corresponding 4 fused IR images (bottom) for each WSN deployment . . . 99
5.10 Graph showing the performance of the DNN based on processing data fused from 4 sensor nodes, comparing 4 WSN deployments . . . 100
5.11 WSN architecture showing integrated data capture and data processing . . . 101

5.12 Simulation process illustrating creation, capture and processing of sensor data based on integrated data capture and processing . . . 102
5.13 Graph showing the performance of the DNN based on processing data locally on sensor node, comparing 4 WSN deployments . . . 103

6.1 DNN-2: Single-branch Multimode DNN Block Diagram . . . 110
6.2 DNN-3: Multi-branch Multimode DNN Block Diagram . . . 113
6.3 3D coverage of a single PIRS . . . 116
6.4 3D coverage of a single IRS . . . 117
6.5 3D coverage of a single PIRS and IRS . . . 118
6.6 3D coverage area of a single PIRS and two IRS . . . 119
6.7 3D side view of a single PIRS angled at -55° along zy axis . . . 121
6.8 3D coverage of four PIRS and two IRS (side view) . . . 121
6.9 3D coverage of four PIRS and two IRS (top view) . . . 122
6.10 Floor maps and simulated reconfigured node deployments for 3 small commercial buildings . . . 125
6.11 Test performance of single and multimode DNNs using Office test data . . . 127
6.12 Test performance of single and multimode DNNs using Meeting room test data . . . 127
6.13 Test performance of single and multimode DNNs using Kitchen test data . . . 128

7.1 Graphical representation of the OcCEMS system . . . 132
7.2 OcCEMS Process . . . 133
7.3 Mock-up of OcCEMS homepage built on OpenEMS UI . . . 135
7.4 Ground (left) and first (right) floor baseline building model . . 137
7.5 Gas consumption simulated by 3 iterations of the energy consumption baseline model . . . 141
7.6 Electricity consumption simulated by 3 iterations of the energy consumption baseline model . . . 141
7.7 Floor map and simulated OcCEMS deployment in ground floor of Manchester-based SME . . . 143
7.8 Floor map and simulated OcCEMS deployment in first floor of Manchester-based SME . . . 144
7.9 The actual gas consumption compared to the simulated gas consumption of Model3 and OcCEMS . . . 146

7.10 The actual electricity consumption compared to simulated electricity consumption of Model3 and OcCEMS 146

List of Tables

2.1 Overview of existing work utilising ML to optimise IRS data for occupancy detection 25
2.2 Performance comparison with existing systems 27

3.1 Calculated sensor Cartesian co-ordinates 40
3.2 Performance comparison of simulated and real data processed by DNN 46

4.1 Comparison of sensor network features based on deployment techniques 73
4.2 Comparison of the performance of sensor data based on deployment techniques 77

5.1 Generalisation ability of Model.i5 90
5.2 Test performance of DNN based on varied epochs, Sigmoid and Sigmoid activation functions and office test data 92
5.3 Test performance of DNN based on varied epochs, Sigmoid and Sigmoid activation functions and meeting room test data 92
5.4 Test performance of DNN based on varied epochs, Sigmoid and Sigmoid activation functions and kitchen test data 92
5.5 Test performance of DNN demonstrating highest generalisation ability (Sigmoid activation function and 2 training epoch) 93
5.6 Number of actual occupants compared to detected occupants 98
5.7 Comparing 4 WSN deployments; each comprising groups of 4 sensor nodes transmitting data to separate sink node with DNN 104
5.8 Comparing 4 WSN deployments; each comprising DeNNOTE nodes with both sensors and integrated DNN 104

6.1 Comparison of node density based on different sensor combinations 125
6.2 Comparison of Performance of DNN-1, DNN-2 and DNN-3 with Office data (%) 126

6.3 Comparison of Performance of DNN-1, DNN-2 and DNN-3 with Meeting room data (%) 126
6.4 Comparison of Performance of DNN-1, DNN-2 and DNN-3 with Kitchen data (%) 126

7.1 Cost of DeNNOTE node components 139
7.2 DeNNOTE node specifications and cost 139
7.3 Key of DeNNOTE nodes for a real-life deployment on the ground floor 145
7.4 Key of DeNNOTE nodes for a real-life deployment on the first floor 145

List of Abbreviations

2D, 3D Two dimension, three dimension

°C Degrees Celsius, a unit measure of temperature

AI Artificial intelligence

BASEE Boosting access for SMEs to energy efficiency

BEIS Business, Energy and Industrial Strategy

BMS Building management system

CAD Consumer access device

CO_2S Carbon dioxide sensor

CO_2 Carbon dioxide

DEC Display energy certification

DeNNOTE **De**ep **N**eural **N**etwork **O**ccupancy detection **T**hermal **E**dge-based system

DNN Deep neural network

EMS Energy management system

ESM Energy saving measures

FOV Field of view

GHG Greenhouse gas

HSE The Health & Safety Executive

HVAC Heating, ventilation and air conditioning

ICT Information communication technology

IoT Internet of Things

IRS Infrared sensor

kWh kilowatt-hour, a unit of energy

LS Light sensor

lux Luminous flux per unit area, a unit of illuminance

MIoTs Smart building **M**apping **IoT** **s**imulation tool

MLP Multi-layer perceptron

ML Machine learning

MM Multimode

MTOE Million tonnes of oil equivalent

NN Neural network

OcCEMS **Oc**cupant-**C**entric IoT-based **EMS**

OpenEMS Open Source EMS integration platform

PIRS Passive infrared sensor

POE Post occupancy evaluation

RFID Radio frequency identification

SC Specific challenges

SMET2 Smart metering equipment technical specification version 2

SME Small and medium sized enterprise

SM Single-mode

SN Sensor network

SO Specific objectives

UI User interface

WSN Wireless sensor network

Nomenclature

$\mathbb{R}$	Real numbers.
α	Accuracy.
$\beta(\cdot)$	Flatten function.
$\bigcap$	Logical AND.
$\bigcup$	Logical OR.
$\bigoplus$	Concatenation of multiple values.
η	Neurons in each dense hidden layer
$\lambda, \lambda_l, \lambda_l o, \lambda_w, \lambda_w o$	Constant to determine whether to place a sensor overlapping the space perimeter, length, overlapping length, width, overlapping width.
$\lceil \cdot \rceil$	The absolute value.
$\mathbb{N}$	Natural numbers.
$\overline{x_{\mathrm{N}}}$	Mean value of numeric data $\mathbf{x_N}$.
ψ	First-order stochastic gradient descent function named Adam.
$\sigma_1(\cdot), \sigma_2(\cdot), ...{}_4(\cdot)$	Activation function for dense hidden layers 1, 2... 4.
$\mathbf{X_N}$	Numeric data where $\mathbf{X_N} \in \mathbb{R}^{a \times j}$.
$\mathbf{b}_1, \mathbf{b}_2....{}_4$	Bias vector for hidden layers 1, 2... 4.
$\mathbf{c}_c$	Concatenated layer.

DNN-1 Single-mode DNN.

DNN-2 Multimode single branch DNN.

DNN-3 Multimode multi branch DNN.

DNN-S2e Optimised DNN model.

$\mathbf{f}$ Flatten layer.

$\mathbf{h}_1, \mathbf{h}_{2....4}$ Hidden layer 1, 2... 4.

OcCEMS-EP OcCEMS EnergyPlus model.

$\mathbf{W}^{'}$ New weight matrix.

$\mathbf{W}_1, \mathbf{W}_{2....4}$ Weight matrix for hidden layers layers 1, 2... 4.

$\mathbf{x_{Cc}}$ Combined CO_2data captured by the node.

$\mathbf{x_{Ic}}$ Combined IR data captured by the node.

$\mathbf{X_I}$ Image data where $\mathbf{X_I} \in \mathbb{R}^{a \times c \times d \times e}$.

$\mathbf{x_N}$ Single sample of standarised numeric input data.

$\mathbf{x_{Pc}}$ Combined PIR data captured by the node.

$\mathbf{x}$ Single sample of data.

$\mathbf{y}$ Array of ground-truth data labels.

θ Sparse categorical cross-entropy loss function.

$\tilde{\mathbf{y}}$ Array of labels predicted by DNN model.

$\{\cdot\}$ The respective shape is transposed into the corresponding tessellating rectangle.

A Area.

a	Batch size of IR image data.
a, b, c	Dimensions of a right angle triangle.
A_c	Combined node coverage area.
A_C, A_H, A_R, A_S, A_T	Area of circle, rectangle, square or triangle.
$A_I, A_{I,1}$	Sensor coverage area of a single IRS.
$A_P, A_{P,1}$	Sensor coverage area of a single PIRS.
A_t	Tessellating node coverage area.
A_{CO_2}	Sensor coverage area a single CO_2S.
$A_{t\{P,1\}}$	Tessellating node coverage area of a single PIRS.
B, H	Base and height dimensions of a triangle.
C	Circumference of a circular shaped space or room.
c	IR image width.
d	IR image height.
e	IR image channels.
f	Field of view of sensor or node.
f_I	Field of view of an IRS.
f_P	Field of view of a PIRS.
f_{Il}	Length-way's field of view of IRS.
f_{Iw}	Width-way's field of view of IRS.
g	Image size, where $g = c \times d \times e$.
h	Height of node in relation to the floor.

i Incremental index.

j Number of single dimension numeric data inputs.

k Number of sensor perimeters.

L DNN Loss score.

l, l_o Space or room length, overlapping length.

m Number of sensors within a single node.

n Number of samples in a batch of IR image data.

$N_{\text{s-x}}, N_{\text{s-y}}$ Node spacing along the x or y axis.

N_{sHo} Node spacing within the overlapping hexagon deployment.

N_{s} Node spacing.

P Sensor perimeter.

p Current sensor perimeter.

q Concatenated data size, where $q = g + j$.

R Radius of a circle.

r Sensor or node range radius.

r_c Sensor communication range.

S Node density.

s Standard deviation of numeric data $\mathbf{x_N}$.

S_G, S_H, S_P Node density based on a sensor grid, sensor hexagons or sensor perimeters deployment.

$S_{CG}, S_{PG}, S_{RG}, S_{SG}$ Node density in a circular, polygon, rectangular or square shaped space based on a sensor grid deployment.

$S_{CP}, S_{PP}, S_{RP}, S_{SP}$ Node density in a circular, polygon, rectangular or square shaped space based on a sensor perimeters deployment.

S_{Go}, S_{Ho}, S_{Po} Node density based on an overlapping sensor grid, sensor hexagons or sensor perimeters deployment.

S_{PH}, S_{RH}, S_{SH} Node density in a polygon, rectangular or square shaped space based on a sensor hexagons deployment.

S_{Rl}, S_{Rw} Sensor densities along the length and width of the sensor perimeters.

$SC, SC_{\text{SG}}, SC_{\text{SH}}, SC_{\text{SP}}$ Space coverage, space coverage of a sensor grid, sensor hexagons or sensor perimeters deployment.

u DNN epochs.

w, w_o Space or room width, overlapping width.

x The current edge in a sensor perimeter.

X_{total} Total number of edges in a sensor perimeter.

Chapter 1

Introduction

1.1 Research Motivation

Worldwide, energy consumption is rising and buildings are responsible for a significant proportion of this consumption. According to the Department for Business, Energy and Industrial Strategy (BEIS), in 2018 the UK's energy consumption was 143 million tonnes of oil equivalent (MTOE), where 31.2% was consumed by the industry and service sector [1]. All buildings consumed 40.3% of this total, with commercial buildings responsible for 19.2% of the total. In 2015, there were 5.38M UK businesses, 99% of which were small and medium-sized enterprises (SMEs) [2], which were responsible for 48.4% of the commercial energy consumption.

Research shows that long term energy reductions can be achieved through energy saving initiatives that include consumption monitoring and providing feedback data to users [3]. Studies of SMEs [4] [5] [6] show that the barriers to uptake of energy reduction initiatives include high costs, significant time commitments and lack of expertise. Many commercial building energy monitoring and reporting systems do exist, but these systems are largely inaccessible to SMEs because they generally fit into three categories; being complex and expensive, affordable but requiring expertise to fit and manage, or affordable but overly simple and ineffective. In the 2008 Climate Change Act, the UK committed to an 80% reduction in its carbon emissions by 2050 with respect to pre-1990 levels. Further to this, in June 2019, the UK committed to "at least 100%" [7] reduction of GHG emissions by 2050 compared to pre-1990 levels. If the UK is to meet these ambitious targets, significant steps must be taken to enable more SMEs to overcome the barriers they have faced in reducing their energy consumption, so they may become part of the solution. The

occupant-centric Internet of Things (IoT) based energy management system (EMS) that is proposed and developed in this thesis can make energy reductions accessible to SMEs, enabling individual enterprises to achieve a reduction of more than 10%.

1.2 Research Aims

The aim of this research is to develop an occupant-centric IoT-based EMS, which will enable SMEs that operate within small buildings to automate a reduction in their energy consumption whilst still maintaining a comfortable working environment[1].

1.3 Research Objectives

In developing an occupant-centric IoT-enabled EMS specific to small commercial buildings, a number of specific challenges (SC) were identified:

1. SC1: To understand the barriers that SMEs face when trying to reduce their energy consumption.
2. SC2: To determine a route to enable these barriers to be overcome.
3. SC3: To define a methodology to unobtrusively monitor a building, its occupants and their energy consumption without causing disruption, privacy or security issues.
4. SC4: To outline a methodology to automate energy reductions whilst maintaining a comfortable environment.
5. SC5: To evaluate the performance of the energy reduction methodology with respect to overcoming barriers and enabling a reduction in energy consumption.

As such, the following specific objectives (SO) will meet these challenges and the aim of this thesis:

1. SO1: Study existing literature to gain a clear understanding of the energy usage of SMEs that operate from small commercial buildings and the existing energy-saving solutions.

[1]Comfortable working environment refers to the internal properties of the building including thermal comfort, air quality, light levels and acoustic comfort.

2. SO2: Design an energy management system which will enable SMEs to reduce energy consumption whilst overcoming the identified barriers.
3. SO3: Assess existing sensor deployment techniques and multimodal data fusion techniques utilised for building monitoring.
4. SO4: Define the requirements for an unobtrusive, accurate and low-cost EMS based on the outcomes of SO1, SO2 and SO3.
5. SO5: Develop and evaluate an EMS in terms of its ability to reduce energy consumption whilst maintaining a comfortable working environment using the system requirements defined in SO4.

1.4 Key Contributions

The key contributions of this thesis were:

1. In Chapter 3, a smart building mapping and simulation tool is developed and evaluated. This contribution also includes a journal paper 'MIoTs: A smart building mapping and simulation tool', which has been submitted to the Elsevier Journal of Simulation Modelling Practice and Theory.
2. In Chapter 4, the optimal WSN deployment technique is identified to meet the cost, layout and coverage requirements of a small commercial building. This contribution also includes a journal paper 'Optimisation of Wireless Sensor Network Deployments for Occupancy Detection', which has been submitted to the Elsevier Journal of Building Engineering.
3. In Chapter 5 and 6, an edge-based deep neural network (DNN) is defined and extended to enable the accurate determination of occupancy. This contribution also includes a journal paper 'A Deep Neural Network for an Edge-Based Occupancy Detection System', which has been submitted to the IEEE Journal; IEEE Transactions on Industrial Informatics, Special Section on Advanced Collaborative Technologies for Artificial Intelligence of Things submission.
4. As part of Chapter 5, a large labelled multimodal building monitoring data set is created. The data set has been made publicly available on the Manchester Metropolitan e-space research repository and is accessible at:

- URL https://e-space.mmu.ac.uk/id/eprint/627856
- or DOI https://doi.org/10.23634/MMUDR.00627856

The multimodal data set is captured in a number of different internal environments and includes infrared thermal data, passive infrared data, CO_2 data and temperature data. The data is captured at rate of once per minute.

5. In Chapter 5 and 6, a multimodal IoT sensor node is developed and deployed as part of a WSN to achieve coverage of a large area, monitoring both occupancy and building conditions. This contribution also includes a conference paper 'Energy Monitoring Solution for SMEs', which has been published in the Proceedings of the International Conference on Sustainable in Energy and Buildings in Croatia, September 2020. Sustainability in Energy and Buildings Journal, Volume 7.
6. In Chapter 7, using the EnergyPlus energy simulation tool a baseline data generation model is developed. This model includes building's construction material and building usage patterns, such that the the model significantly reduces the building's energy monitoring lead time by accurately predicting the baseline energy consumption.
7. In Chapter 7 an holistic occupant-centric IoT-based EMS for small commercial buildings is developed, enabling SMEs to overcome the barriers they have faced in reducing their energy consumption whilst maintaining a comfortable environment.

1.5 Preliminaries

In this section, the concepts of the IoT and worldwide energy consumption are introduced as preliminary concepts to this work.

1.5.1 Internet of Things

The IoT is a network of everyday items, connected together through the Internet. The purpose of IoT is to improve quality of life by enabling the best response to an environmental change [8], [9]. This purpose is achieved by an IoT endpoint monitoring its environment to enable, assist with or automate a response to a change in this environment [10], [11]. An IoT endpoint

commonly referred to as the Thing, Device or Node, can be anything that includes the technological components to enable the Thing to connect to the Internet. An IoT end user can be a human, machine, or a combination [12]. IoT is not a specific device or technology, instead, it is the inter-working of different technologies, which enables the connectivity of many *Things*.

IoT consumer applications such as smartwatches, smart heating and voice-activated personal assistance, are now common. But IoT has also expanded to many other areas such as: commercial IoT products which include monitoring systems; Industrial IoT (IIoT) which enable the automation of many industrial processes; medical IoT (MIoT) such as applications used within medical procedures or biometric implants; Military IoT (IoMT) which includes drones and combat-based wearables; and Infrastructure-based IoT which includes smart city infrastructure for buildings, energy and transportation. This is not an exhaustive list and IoT is currently evolving with the addition of artificial intelligence (AI), machine learning (ML), augmented reality (AR) and virtual reality (VR), in applications from entertainment and infrastructure to environment management. In the next section, the topic of worldwide energy consumption will be discussed, followed by Chapter 2 which will introduce how technology, specifically IoT and AI can be used to help to reduce energy consumption.

1.5.2 Energy Consumption and Climate Change

Energy is a foundational commodity in modern society. To enable modern society to function and improve, we need reliable, secure, sustainable, affordable energy supply and distribution networks. In contrast to this requirement, it is widely recognised that our main energy source, fossils fuels, are limited. At present, fossil fuels, account for 81.2% of the worldwide energy supply [13]. The current known supplies of oil, gas and coal are only predicted to continue to meet energy requirements until approximately 2090. Worldwide energy consumption is increasing, particularly in emerging markets. Since the 1970s the energy demands of Asia and Africa have increased seven fold [13]. Nations are aware that unless significant changes are made to our energy usage, global energy requirements will continue to increase [13]. In addition to diminishing supplies, the use of these energy sources are causing catastrophic climate change [14].

In 2008, the Climate Change Act was passed in the UK. This committed the UK to reduce greenhouse gas (GHG) emissions by 80% by 2050, compared to 1990 levels. In 2015, 120 nations signed the Paris Agreement. The purpose of this agreement was to "strengthen the global response to the threat of climate change by keeping a global temperature rise this century well below 2 degrees Celsius above pre-industrial levels and to pursue efforts to limit the temperature increase even further to 1.5 degrees Celsius" [15]. The agreement requires nations to submit their targets and to increase their efforts towards this aim. In 2019, the UK increased its legal commitment from its 2008 commitment, to reach net-zero GHG emissions by 2050. In some countries, including the UK, the steps taken towards meeting these commitments are beginning to take effect. The UK's energy consumption dropped by 1% from 2018 to 2019 [16]. However, a recent assessment by a leading government think tank regarding the UK's ability to meet these targets is damning [17]. The assessment is summarised as "a lack of coordinated policies, constant changes of direction, a failure to gain public consent for measures and too little engineering expertise and delivery capability has left the UK well off track to meet its target" [18].

According to a recent progress report from the Committee on Climate Change to the UK government, their number one investment recommendation was "Low-carbon retrofits and buildings that are fit for the future" [19]. The report also highlights significant opportunities to (1) lead a move towards positive lower-carbon behaviours in the UK workforce and (2) target science and innovation funding [19]. Significant amounts of research suggest that technology can be utilised in the climate change fight [19]–[22]. One study suggests that information communication technology (ICT) will reduce GHG emissions by 7.8 Gigatonnes of CO_2 equivalent; 15% of global 2020 GHG emissions [22]. In Chapter 2, energy consumption in buildings will be considered in more detail, before introducing how IoT technology can be utilised as a part of the solution to target global warming.

1.6 Thesis Organisation

The remaining sections of this thesis are organised as follows:

Chapter 2 details the motivation for focusing this research on energy reduction within small commercial buildings. It also introduces the enabling

technologies, WSNs, which are utilised to deploy IoT hardware and AI, which will be used to enhance the performance of the IoT endpoints.

Chapter 3 details the specification and evaluation of a smart building mapping and simulation tool which is referred to as MIoTs. MIoTs is evaluated against real building data, such that the tool is used in a number of chapters to map sensor deployments and generate building data. In Chapter 4, MIoTs is used to assess the WSN deployments, in Chapter 5 to evaluate the deployment of the edge-based occupancy detection system, in Chapter 6 to assess the reconfigured node hardware and Chapter 7 to model and evaluate the occupant-centric EMS.

Chapter 4 presents a number of WSN deployment techniques, including algorithms to determine IoT node positions, node density and node coverage. The deployment techniques are assessed in terms of their cost, space coverage and the ability of sensors within these deployments to detect/count occupants and monitor building conditions.

In Chapter 5, an edge-based occupancy detection system is developed and evaluated. This system utilises AI to improve the performance of the IoT node deployments that were presented in Chapter 4. The system is evaluated in terms of its ability to accurately detect the presence of occupants within a monitored region. Also, the generalisation ability of the AI techniques are assessed in a number of different building environments.

In Chapter 6, the IoT node hardware proposed in Chapter 4 and developed in Chapter 5 is reconfigured to achieve a larger coverage area. Also, the AI techniques developed in Chapter 5 are expanded to process multiple heterogeneous inputs simultaneously.

In Chapter 7, the contributions of Chapters 4 to 6 are combined to create an occupant-centric IoT-based EMS. As part of this system, a baseline model is developed and a range of IoT nodes with different specifications are proposed. The EMS is evaluated based on it cost and ability to reduce energy consumption.

Chapter 8, the conclusions of this thesis are discussed, limitations are highlighted and areas for further work are suggested.

Fig. 1.1 shows the organisation of this thesis based on the development of concepts.

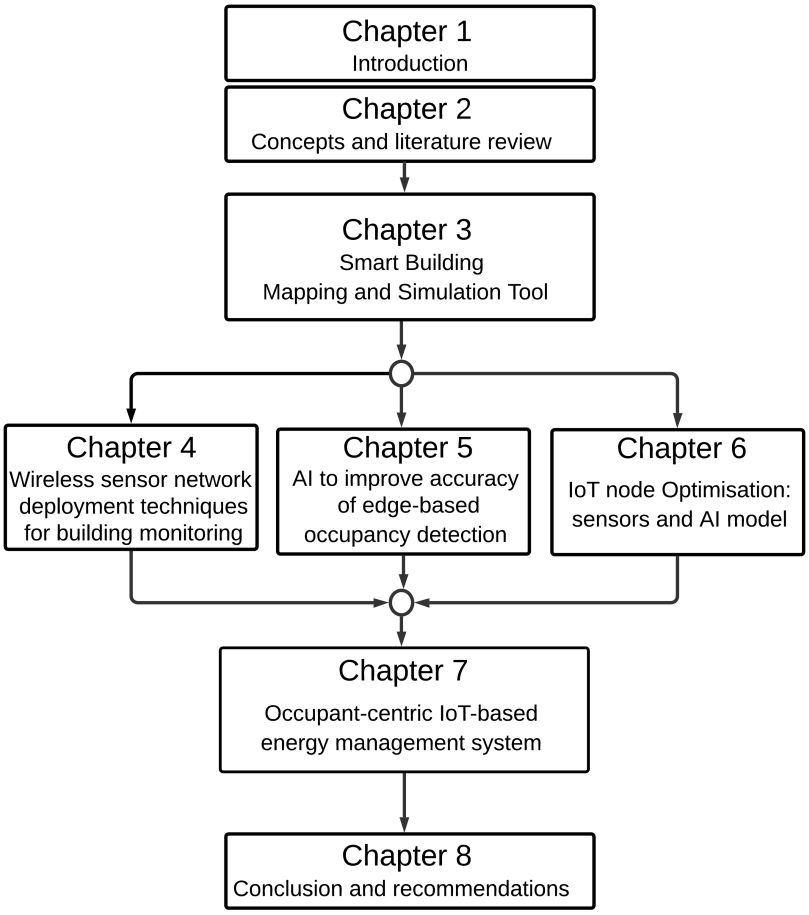


FIGURE 1.1: Conceptual organisation of Thesis

1.7 List of Publications

1.7.1 Book Chapters

1. Popoola, S. I., **Ande R.**, Kassim, B. F., Adebisi, B. "Deep Bidirectional Gated Recurrent Unit for IoT Botnet Detection in Smart Homes", Machine Learning and Data Mining for Emerging Trends in Cyber Dynamics. Springer Verlag, UK, 2020

1.7.2 Journal Papers

1. **Ande, R.**, Adebisi, B., Hammoudeh, M. & Saleem, J. "Internet of Things: Evolution and Technologies from a Security Perspective", Elsevier Sustainable Cities and Society Journal, Vol. 54, pp.101728-101743
2. Ikpehai, A., Adebisi, B., **Ande, R.**, Rabie, K.M., Anoh, K., Hammoudeh,

M., Gacanin, H. and Mbanaso, U.M., 2018. "Low-power Wide Area Network Technologies for Internet-of-things: A Comparative Review", IEEE Internet of Things Journal, Vol. 6(2), pp.2225-2240.

3. Popoola, S., **Ande, R.**, Kassim, F., & Adebisi, B., Hammoudeh, "SMOTE-DRNN: A Deep Learning Algorithm for Botnet Detection in the Internet of Things", IEEE Internet of Things Journal, 2021. (under review)

1.7.3 Conference Papers

1. **Ande, R.**, Ikpehai, A., Adebisi, B., & Hammoudeh, M. "Energy Monitoring Solution for SMEs", In Proceedings of International Conference on Sustainability in Energy and Buildings, Croatia, September 2020, Vol. 7, Sustainability in Energy and Buildings Journal.
2. Saleem, J., **Ande, R.**, Hammoudeh, M., Raza, U. & Adebisi, B., "IoT Standardisation: Challenges, Perspectives and Solution", In Proceedings of the 2nd International Conference on Future Networks and Distributed Systems, June 2018, pp. 1-9.
3. Saleem, J., **Ande, R.**, Adebisi, B. & Hammoudeh, M. "A State of the Art Survey-Impact of Cyber Attacks on SME's", In Proceedings of the International Conference on Future Networks and Distributed Systems, July, 2017

Chapter 2

Concepts and Literature Review

2.1 Introduction

Global energy consumption within buildings rose by more than 8% between 2010 and 2019 [14]. According to research by BEIS, in 2018, the total UK energy consumption was very large at 143 MTOE. Where the domestic sector consumed 28.9%, transport: 39.9%, industry and service sector combined: 31.2% [1]. Within the UK, buildings are responsible for 40.3% of all energy usage, with commercial buildings consuming 49.4% [1]. Considering solely commercial building consumption, 40.9% was consumed for space heating, lighting, water processing and appliances, with a further 33.2% consumed for industrial processes [1]. This means commercial space heating, lighting, water processing and appliances[1] account for 8.1% of the UK's national energy consumption.

In the subsequent sections, a number of concepts, the related issues and the existing work will be introduced. These concepts include the use of IoT-based building management systems to reduce energy consumption, energy management within small commercial buildings and the enabling technologies for building energy management.

2.2 Building Monitoring

Research into existing IoT systems has demonstrated that IoT is an ideal technology for monitoring buildings [23], [24]. Commercial buildings, in particular, have many requirements which can be monitored and managed by IoT-based systems. Such systems are commonly known as building

[1]This includes laptop and desktop computers and office equipment such as kettles, fridges and printers.

management systems (BMS). These requirements include occupant comfort, building usability, building security and energy consumption [23]. Due to the huge and complex nature of energy consumption within commercial buildings, it is an ideal sector for technical solutions which incorporate IoT technology [23]. Such systems are referred to as EMS, which can be a standalone system or a subset of a larger BMS.

There is already significant research into IoT-based EMS[2] for commercial buildings [25], with research demonstrating these systems can reduce energy consumption in a single building by up to 42% [24]. Though there are some gaps in this research, particularly research that is focused on small commercial buildings. Also, some existing systems have oversimplified the relationship between building users[3] and energy consumption, whilst others systems are complex and too expensive for many buildings.

An example of an oversimplified EMS includes the use of passive infrared sensors (PIRS), commonly referred to as motion sensors, to indicate room occupancy [26]. If there are no detected occupants the EMS will automatically turn-off of energy-consuming systems such as lights and heating. PIRS detect moving occupants but are unable to detect stationary occupants [27]. In instances where building occupants are stationary for long periods[4], the selected sensors may not detect them. Additional examples include sensors whose data are analysed using ML to indicate occupancy [28]. If the accuracy of these systems is low, again occupants may be undetected or may be falsely detected when they are not present. Both examples can lead to a high rate of false responses. These false responses can cause systems to be wrongly turned off when they are required, or respectively, turned on when they are not required. Repeated false negatives, i.e. not detecting occupants, can lead to frustration for building occupants [29], who may respond by overriding the EMS causing a rise in energy consumption. Repeated false positives, i.e. indicating occupants are present when they are not can cause the unnecessary activation of systems, also causing a rise in energy consumption.

In comparison, there are many examples of complex and expensive systems designed specifically for large commercial buildings [30], [31]. The cost and complexity of these systems mean they are unsuitable for a huge proportion

[2]Such systems will simply be referred to as EMS from this point forward.
[3]Building users or building occupant are the people who access a building.
[4]For example, carrying out sedentary work based at a work desk.

of commercial buildings, particularly small commercial buildings which will be introduced in Section 2.3. Even though small commercial buildings make up a huge proportion of all commercial buildings, they are largely ignored in energy reduction legislation and incentives [23], [25], which creates a significant gap.

2.2.1 Occupant-centric EMS

The basis of an occupant-centric EMS is to improve the temporal match between the presence of building occupants and when energy-consuming systems are active[5] [32], [33]. Occupant-centric EMS carry out three main monitoring tasks: detecting occupants, monitoring energy usage and monitoring the conditions inside the building. Energy monitoring includes monitoring the amounts of energy that are consumed and its end use[6]. By monitoring conditions inside a building, an EMS can vary settings to create conditions that are comfortable for occupants. These conditions are referred to as occupant comfort levels and include thermal comfort, air quality, light levels and acoustic levels; though this is not an exhaustive list. For an office building, the Health and Safety Executive (HSE) recommend the temperature range is 20°C to 26 °C, humidity 30% to 60%, CO_2levels below 1000 ppm, light levels of 500 lux and acoustic levels of 30 dB to 45 dB [34]–[36].

A comprehensive work by Akkaya et al. [37] surveyed existing occupancy monitoring techniques based on accuracy, cost, intrusiveness and privacy. Occupancy monitoring systems are generally split based on their detection technology, for example establishing occupant levels using communication technologies including WiFi [38] or RFID [39], establishing occupant levels using sensors including phone-based sensors [32], image-based sensors [40] and IoT-based sensors [41]. Some of these technologies create issues for commercial systems. Many communication based systems require user agreement [42] and some demonstrate low performance when occupant levels are low [24], though others, such as RFID-based systems are highly accurate, but have high upfront costs [39]. Phone-based systems require user agreement and create privacy issues [43], [44]. Image-based systems are expensive and create privacy issues [45].

[5]Simply put, this means only turning systems on when there are people present.

[6]Examples of end-use include lighting, space heating, appliances and industrial processes.

In contrast, IoT-based sensor systems vary significantly in performance, but they can be low cost and accurate [41]. Data fusion techniques have also been used to improve the performance of sensor-based systems, enabling occupancy accuracy around 80% [46]. The costs and logistics of sensor-based EMS vary significantly, dependant upon the type of sensors, their quantity and the building environment [47]. Another benefit of sensor-based occupancy systems is the data collected to determine occupancy can also be used to monitor other building data. The use of existing building infrastructures can further reduce system costs [37]. IoT-based sensors which have been utilised for occupancy and building monitoring include temperature, humidity, light, CO_2 [48], sound sensors, PIRS, infrared sensors [49] and Doppler sensors [50].

Based on their cost, achievable performance and adaptability, this thesis will study the application of sensors as part of an IoT-based EMS, which will monitor occupancy and building conditions to reduce energy consumption within commercial buildings.

2.3 Energy Management in Small Commercial Buildings

Earlier in this Chapter, the level of energy consumption in commercial buildings was summarised. In 2015, there were $5.38M$ UK businesses, 99% of which were SMEs [2], responsible for consuming 48.4% of commercial energy. In the UK there are $1.57M$ commercial buildings, 92% of which are small premises that consume a third of all commercial energy. Of these small premises, 65% are older buildings [51]. Small premises are defined as buildings less than $1000m^2$, which is equivalent to $11 - 13$ average-sized homes [52] and older buildings are defined as pre-1990 constructions. Older buildings are renowned for high levels of energy inefficiencies, due to low levels of insulation, inefficient glazing, heating and lighting. Research has shown that across the board, the energy consumption of commercial buildings could be reduced by up to 25% through the application of no cost and low-cost energy-saving measures [53]. This reduction could be more than 45% based on deep retrofit energy saving measures [53]. If this level of energy reduction could be achieved across all small commercial buildings this would reduce the national energy consumption by $1.8\% - 3.2\%$, which is a significant step towards the net zero target [7].

Most of the existing work on EMS is focused on domestic EMS or large, complex commercial EMS [23], [54]. The specification of an EMS for a small commercial building differs significantly from a domestic system or system developed for a large commercial building. Firstly, domestic systems are not suitable for commercial buildings since they are developed with different privacy specifications which would not meet employee privacy protection law [55]. Secondly, large enterprises have very different requirements than the SMEs that typically inhabit small commercial buildings. Unlike large enterprises, SMEs are not required to monitor or report their energy usage or their level of energy efficiency. Therefore, as a result of limited research [56], limited government support [23], lack of incentives, lack of legal requirements and the significant barriers that these SMEs face [57], the uptake of energy reduction technologies is much lower in small commercial buildings than large commercial buildings.

2.3.1 Small Commercial Building User Requirements

To facilitate energy reduction in small commercial buildings, the requirements of these buildings and their users must be studied. Generally, an SME is defined as an enterprise with $0 - 249$ employees. For specific sectors including retail, office and hospitality, the inhabitants of small commercial buildings are generally SMEs. This is illustrated in Fig. 2.1, created from data collected in [16]. The figure shows the energy consumption of enterprises based on employee numbers and building size. Energy consumed by micro ($0 - 9$ employees) and small enterprises ($1 - 49$ employees) is shaded with red dots. The energy consumed by small buildings less than $500m^2$ is marked with a red dotted line and square markers. The energy consumed by medium-size enterprises ($50 - 249$ employees) is shaded with green horizontal lines. The energy consumed by small to medium sized buildings $500 - 999m^2$ is marked with a green dashed line and circular markers. Finally, the energy consumed by large enterprises is shaded with a purple checked pattern and the energy consumed by large buildings over $1000m^2$ is marked with a purple dashed line with diamond markers.

The figure illustrates that for the retail and office sectors, the energy consumption based on building size directly correlates with the size of the enterprise, i.e. energy consumption in small buildings corresponds with that of small enterprises; similarly with medium sized buildings

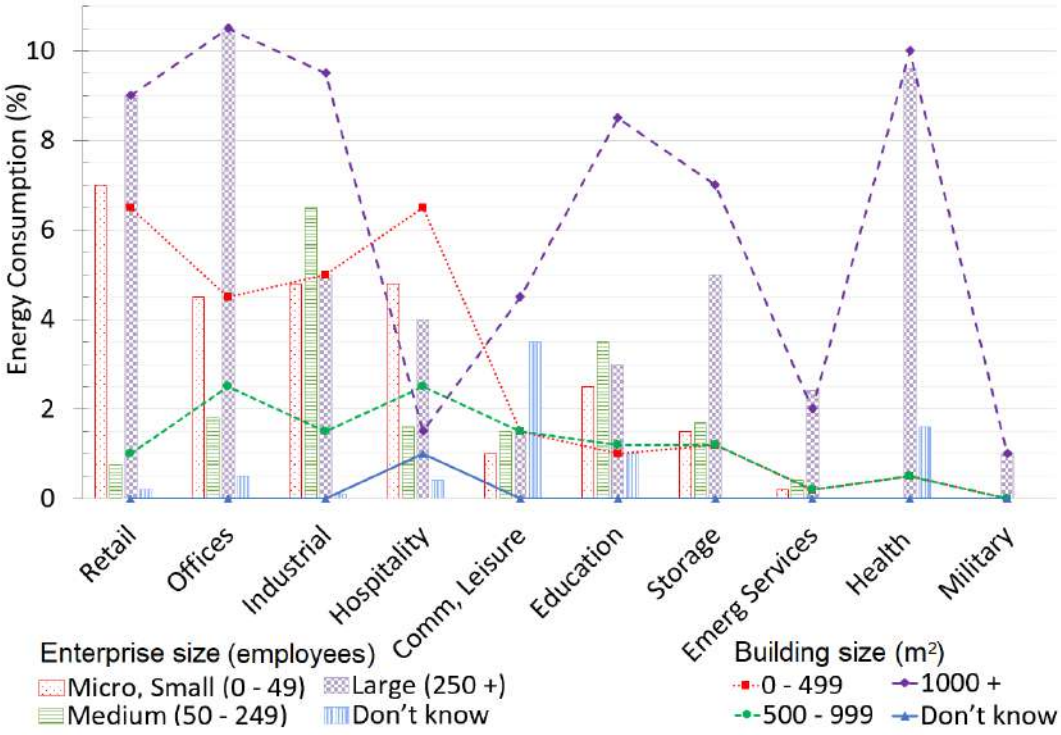


FIGURE 2.1: Energy consumption based enterprise size and building size

corresponding with medium sized enterprises; and large buildings corresponding with large enterprises. This relationship demonstrates that for the retail and office sector, small enterprises occupy small commercial buildings. For the industrial sector, the energy consumption of small buildings also corresponds to the energy consumption of small enterprises, though as enterprise size increases to medium and large, the energy consumption no longer directly correlates with the building size. This change in relationship is due to the energy consumption of industrial enterprises being linked with both the enterprise size and the energy consumption of the industrial processes. Additionally industrial building size is linked with enterprise size and the space required for industrial equipment [16].

Similarly, for the leisure sector, there is a correlation between the energy consumption of small and medium sized enterprises and that of small and medium buildings, but not between large enterprises and large buildings. For a significant number of large leisure buildings e.g. theatres and art galleries, this difference is a result of energy consumption being heavily dependant upon visitor numbers rather than the enterprise size. The research in this thesis will focus on SMEs within the retail and office sector

since it has been established that SMEs in these sectors inhabit small commercial buildings.

SMEs are significantly different from large enterprises and have very different needs and requirements. SMEs are sensitive to operational costs, of which building rent or mortgage costs are a significant proportion, averaging between $2 - 20\%$ of gross income [58]. The HSE 1992 Workplace Regulations [59] defines a minimum floor space of $3.67 - 4.58m^2$ per employee. So, a micro enterprise could occupy a building as small as $3.67 - 33m^2$, plus space for facilities. The Total Office Cost Survey 2019 [60] values an individual work area at £$5,408 - £18,988$ annually[7], which is calculated based on an individual area 2.5 times larger than HSE requirements. Using HSE regulations the surveyed buildings would instead cost £$2,130 - £7,478$ per work area. Research found that SMEs select commercial buildings based on minimal spatial requirements and minimal costs [58]. Based on these findings, this thesis will focus on small commercial building occupied by retail, office and hospitality-based SMEs.

Surveys show that unlike larger enterprises, 89% of UK SMEs do not have a centralised heating, ventilation and air conditioning (HVAC) system [61]. Similarly 80% of SMEs have a boiler and radiator based heating system that is operated on a schedule. The output of boiler-based heating systems is varied to meet occupant comfort levels, but are not varied dependant upon the number of occupants. For the main source of ventilation, 56% of SMEs rely on natural ventilation which is created by opening windows and doors [61]. Interestingly, a large number of existing occupancy based EMS [28], [39], [40], [43], [45], [47], [49], [50], [62]–[65] are developed for buildings with centralised HVAC systems. In contrast to a boiler-based system, these EMS vary the output of the HVAC system based on the number of detected occupants and occupant comfort levels.

2.3.2 Energy Efficiency Barriers for SMEs

The energy consumption of SMEs is not regulated like that of large enterprises. As a result, there is a limited amount of research which has studied energy efficiency within SMEs, referenced studies include [4], [5], [57], [66], [67]. Additionally, the clear lack of government policy focus has led to low expectations with regards how much effect SMEs can have

[7] This values are based on work spaces in Norwich and London.

towards net zero targets. An example of this is "the UK's calculation of expected energy savings from non-domestic energy efficiency policies includes no specific contribution from SMEs" [56]. In comparison to this position, the majority of SMEs want to reduce their energy consumption [61], [67], with $70 - 80\%$ saying they are taking steps towards this [68]. Though a large proportion of SMEs cite significant barriers as the reason for not doing more [56]. There are a large number of barriers that are highlighted by these surveys, though the number one barrier that SMEs cite is cost. Cost barriers include both the upfront costs and lifetime costs that are experienced when SMEs access expertise, retrofitting solutions, or installing specific technologies [56], [67]. Across multiple surveys SMEs list the next most significant barriers as a lack of expertise within their enterprise and time constraints of the individual that is tasked with improving energy efficiency, since improving energy efficiency is generally additional to their main roles within the enterprise [56], [69].

2.3.3 Existing Research on SME Energy Efficiency

A number of policy-based research surveys include a comprehensive survey by Fawcett et al. [56] which highlights the distinctive characteristics of SMEs, the significant potential for energy savings and the current gaps in government policy. Freshner et al. [4] undertook 280 SME energy audits, evaluated them and based on these results, developed a process to make the audits more flexible, enabling greater engagement with SMEs. Fleiter et al. [5] used data from energy audits to propose solutions to overcome SME energy efficiency barriers. Bunse et al. [6] studied the gap between energy reduction measures and their implementation in SMEs. Marquez et al. [67] created a detailed report which reviewed the barriers to energy efficiency for existing Australian buildings and proposed steps to overcome these barriers. These existing works each add to the knowledge in this field, and some [4], [56], [67] make strong recommendations that could inform and enable changes in government legislation.

EMS surveys include Jia et al. [70] that surveyed different aspects of EMS including occupancy, data acquisition and modeling techniques. Akkaya et al. [37] carried out a very comprehensive survey of energy monitoring and occupancy monitoring within smart buildings. Sulistyanto et al. [71] carried out a study of different IoT-based monitoring systems that could be employed to support traditional energy auditing.

There are a number of SME-focused developmental works, though they are all limited in how effective they are since none of these work tackle the barriers that SMEs face. These works include Cosgrove et al. [72] who developed an SME monitoring and targeting plan based on utility bill analysis. Dugay et al. [73] developed a system to reduce energy consumption within micro enterprises by reducing phantom loads. Heilala et al. [74] proposed a monitoring system specifically for SMEs within manufacturing. Johnson et al. [75] and Yang et al [76] proposed systems to monitor energy usage and present the consumption data to users in real-time. Luna et al. [77] proposed a sensor for monitoring building energy consumption. Al-Hassan et al. [78] proposed smart sockets to monitor energy consumption.

2.3.4 Existing Commercial Solutions for SME Energy Management

Existing commercial EMS cost between £20 − £50 per meter squared [79], [80]. A number of lower cost systems are available, such as the Beringar IoT Building Resource and Occupancy Tracking Module [81] costing £450 per module for the hardware, with an annual subscription of £350 per module for software-as-a-service data reporting. It is a plug-and-play system which does not require significant setup time or energy management expertise. The low starting costs and ease of use make Beringar's solution suitable for SMEs. Though it is noted that each hardware module has a coverage area up to $50m^2$ and subscription costs are relatively high, particularly if multiple modules are required. The Pressac Smart Monitoring hardware module [82] costs £750 each. The Smart Citizen Starter Hardware kit [83] costs £100. Though these systems are low cost, both require additional expertise to install, configure and manage, meaning the SME might need to pay installation and support. The Pressac kits include a WiFi, Ethernet and LTE gateway but do not include software for data reporting, instead they are compatible with a number of commercial cloud platforms. The Smart Citizen kits are Open Source and compatible with a range of Open Source reporting tools. Alternate low cost commercial systems which can be used to automate the control of lighting and HVAC systems include PIR motion-based systems [84], though these systems are generally overly simple and ineffective [26], [85].

Research has shown that energy saving initiatives can achieve long term energy reductions [3]. Feedback from surveyed SMEs [5], [6], [25], [61]

shows that barriers to the implementation of energy reductions include high costs, significant time commitments and lack of expertise. Many research-based and commercial EMS do exist but are inaccessible to SMEs because they generally fit into three categories: being complex and expensive; affordable but requiring expertise to fit and manage; or affordable but overly simple and ineffective. Therefore, this thesis will focus on an holistic solution to overcome the barriers that are highlighted above. The enabling technologies which will be utilised are introduced in Section 2.4.

2.4 Energy Management Enabling Technologies

The complexity and high cost of building retrofitting for improved energy efficiency can be overwhelming for many SMEs. As such, IoT, WSNs and ML will be utilised to develop a cost effective and accurate EMS to assist such enterprises to achieve and maintain long term energy reductions. WSNs and ML are introduced in this section.

2.4.1 Wireless Sensor Networks

A WSN is a network of spatially distributed sensor nodes that function together to communicate the data that is captured in their environment through wireless communication links [86]. WSNs were initially developed for environments unsuitable for wired networks. But due to the demand for greater network flexibility, research has paved the way for a huge development in WSN technology. As a result, WSNs are now chosen for their adaptability, flexibility and scalability, resulting in WSN being deployed in environments where wired deployments would be costly, impractical or difficult [86].

A WSN node can be an IoT endpoint[8]. A WSN node is illustrated in Fig. 2.2 and consists of three components: power unit, sensor or actuator and communication unit. A sensor node can also include a processor and storage. WSN nodes are generally constrained devices with limited power, sensing capabilities, processing power, storage or communication capabilities.

[8]It is important to differentiate, an IoT endpoint does not need to be a WSN node, since an IoT endpoint can connect via wire or wirelessly.

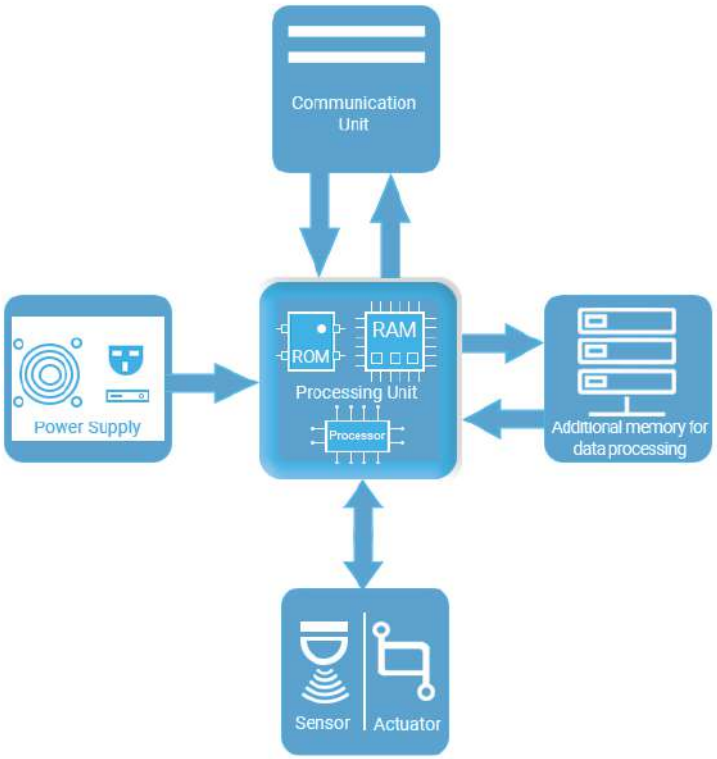


FIGURE 2.2: Components of Wireless Sensor Node

2.4.1.1 Existing WSN Deployment Techniques

There is a significant amount of research in the area of WSNs, according to one survey, it is one of the most researched technologies of the last decade [86]. The majority of work in this field is concerned with network topology [87], [88], energy conservation [89], [90], routing protocols [91], [92], data collection [90], [93] data fusion [94], [95] and network costs [96]. Significantly less work is focused on the physical placement of sensors. However, as WSN are utilised in more applications, particularly for IoT and Industrial IoT (IIoT) applications, efficient and effective deployment techniques are required.

Existing work that is focused on WSN deployments including sensor positions, sensor density or area coverage are discussed below. This includes a survey by Plata-Chaves et al. [97], which reviews existing work focused on optimising multimode WSN. Mini et al. [98] studied the problem of M-coverage, such that each sensor is connected to 'M' number of other sensors. Mini et al. [98] also proposed a solution to use the minimum number of sensors to achieve M-coverage. Cheng et al. [99] developed an

algorithm to control the movement of mobiles sensors to create a sensor barrier and achieve coverage between two external points. Hammoudeh et al. [100] developed a method to evaluate a WSN border crossing detection system. This included determining the node density required to achieve barrier coverage. They also proposed a WSN routing protocol. Zhoe et al. [101] studied the deployment of mobile sensors into an existing WSN to achieve improved network performance and proposed an algorithm to achieve the maximum area coverage using the minimum number of sensors.

There are a number of existing works that utilise a virtual pattern to assist with optimal positioning of nodes. This includes a study by Chakrabarty et al. [102] that proposed an outdoor surveillance-based grid placement technique, positioning multimode sensors with differing coverage areas. Chakrabarty et al. [102] also studied deployment costs. Similarly, Dhillon et al. [103] proposed a number of algorithms that utilised a grid to determine optimal sensor positions whilst trying to achieve optimal coverage in an external environment. Nazarzehi et al. [104] proposed an algorithm to deploy and manage the movement of mobile sensors to enable the rapid exploration of a 3D outdoor environment to find a target. The algorithm utilised a grid system to manage the sensors. In comparison, Al-Turjman et al. [105] proposed a hexagon-based WSN deployment algorithm and modelled its energy consumption based on the Zigbee communication protocol.

It is noted that the majority of literature studying the physical position of WSN nodes are focused on external environments since indoor WSN are generally WiFi based and well understood. The current work, which is concerned with determining the optimal WSN deployment, based on the minimum cost and maximum level of coverage to enable accurate monitoring of an internal building environment. Additionally, the existing literature does not compare network features such as total coverage or coverage overlap. Based on this gap, Chapter 3 will study existing WSN deployment techniques to determine the optimal technique for IoT-based occupant and building monitoring. It will also propose algorithms to determine the node density and area coverage of each technique.

2.4.2 Machine Learning

IoT was introduced in Section 1.5.1 as a collection of Things which can collect data about their environment, process this data, automate a response

and transmit data through the Internet. IoT can be further optimised by integrating it with AI, which is the technique of training machines to perform human tasks [106]. ML is a subset of AI and can be defined as the process of training a machine how to learn from the data it is given. The integration of ML enables an IoT network to apply advanced data analytics to garner valuable insights from its data [107]. In short, ML algorithms and techniques can significantly enhance the performance of an IoT system to analyse and learn from the data it has captured [108]; detecting patterns or anomalies in IoT data. ML is 20 times faster than other analysis techniques and delivers a significantly higher level of accuracy [107]. Within this research it was found that some ML algorithms require a significant amount of processing power both during training and deployment, which may not be compatible with resource constrained IoT endpoints. As such, ML techniques need to be developed specifically for edge-based IoT endpoints. A successful edge-based approach can significantly reduce the amount of data that needs to be transmitted around a network [109], which in turn can reduce system costs and improve latency, responsiveness and system scalability [110].

2.4.2.1 Deep Learning Techniques

There are a number of ML techniques and in the current state-of-the-art occupancy-based EMS, deep learning (DL) techniques demonstrate a good level of performance. Within DL, there are a number of techniques to build a DL model. These models include DNN, Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). Unlike shallow neural networks, deep learning models have two or more hidden layers which helps the network to learn the hierarchical representations of data using multiple levels of abstraction [111]. This architecture is scalable, i.e. it maintains efficient classification performance even as the volume of data becomes large without overfitting to the training data. When comparing DL architectures, the complexity of DNN is lower than other DL architectures such as CNN and RNN. Therefore, DNN models requires minimal data pre-processing, shorter training time, lesser computation, and smaller memory space. These characteristics make the DNN comparatively more well suited to development on a constrained edge-based system. Also, DNN increases detection accuracy and reduces false alarm rate.

This thesis will focus on a feed forward DNN deep learning model. Data that

is fed into the DNN model will move from the input layer towards the output layer without going in the opposite direction, whilst the loss calculation is propagated backwards to allow weights to be altered. Due to selecting a feed forward DNN, this model will require less processing power than some of the alternative models. A DNN is a supervised learning model, meaning it requires ground-truth labels in the training stage. The DNN model is trained to learn the discriminative features in data. When the DNN model is given unseen data, it will classify them into the predefined output categories.

A DNN comprises multiples layers, where each layer consists of a number of neurons. Each neuron receives its input from the previous layer and performs a single computation. The layers of neurons together implement a complex nonlinear computation [112]. The DNN uses the training data to define weight and matrix bias of each neuron based on the relationship between the input features and the output labels. The structure of the DNN, i.e. the number of hidden layers, nodes in each hidden layer, learning rate, batch size and activation function are specified by the DNN's hyperparameters [113]. These hyperparameters can be varied to create an optimal DNN model [114]. koutsoukas et al. [115] investigate which hyperparameters to vary, including the number of hidden layers, activation functions and the number of epochs over which the model is trained. Techniques from this study [115] will inform the optimisation of the DNN that is developed.

2.4.2.2 Deep Neural Network for Occupancy Detection

As part of this thesis, an ML model will be developed to process the IoT building data to quickly and accurately determine whether occupants are present. One measure of the model's performance is its generalisation ability, which is it ability to correctly categorise unseen data [116]. Throughout this thesis, generalisation ability will refer the model's ability to accurately interpret new and unseen data which has been captured in new but similar building environments compared to the training environment. Both the training and test environments must share similar characteristics such as types of heating systems, temperature ranges and similar occupant usage patterns and behaviours. For example, two similar environments could include Offices A and B. In both environments, the occupants are sat at desks, displaying minimal movement and dissipating low levels of energy. An example of different environments could be Office A, Office C

and Gymnasium D. Where Office A is an SME office with a boiler-based heating system and open-able windows for ventilation. Office C is a large enterprise office with a centralised HVAC system. Gymnasium D, is both a totally different environment and its occupant display different usage patterns. As such, A, C and D are considered too different to measure the model's generalisation ability.

For the proposed DNN model to demonstrate a high generalisation ability, it will perform well when trained with data captured in Office A and tested with unseen data captured in both Office A and B, without the need to retrained with data captured in B. The benefit of a model demonstrating a high generalisation ability within similar environments is, the trained model can be replicated and deployed across multiple building environments with a high level of confidence that the replica will also perform well. A model demonstrating a low generalisation ability would need to be retrained for each deployment environment, which is a complex and time consuming process.

2.4.2.3 Existing IoT-based Occupancy Detection with Applied ML

There is a significant amount of literature in this field, but based on the gaps that were highlighted in Section 2.3, the review will focus on the use of ML to optimise the performance of infrared sensor (IRS) based occupancy detection systems. IRS can accurately detect stationary and moving heat sources and there are a number of existing occupancy detection systems that utilise IRS. These systems incorporate a range of techniques to prevent the systems from wrongly identifying stationary, non-occupant based heat sources, for example a heater or computer, including data fusion [40], image processing (excluding ML techniques) [28], [40], [49], [50], [64], [65], ML [28], [40], [49], [50], [65], or a combination. The state-of-the-art in the area of AI-enabled occupancy detection systems that use thermal data are summarised in Table 2.1 and their performance is summarised in Table 2.2.

Existing work includes a highly referenced 2013 study [28], in which the ThermoSense system was developed. Beltran et al. work compared the performance of 3 ML algorithms, trained with data from an IRS and PIRS and deployed the system to control building HVAC systems. Tyndall et al. [49], further developed the ThermoSense study and considered the temporal effect of collecting data more frequently. Similarly, Cao et al. [40] studied the performance of an artificial neural network (ANN) model,

TABLE 2.1: Overview of existing work utilising ML to optimise IRS data for occupancy detection

Ref.	AI technique	Image processing	Data size	Sensor	No. of test envir	Max No. of occup
[28]	Compared KNN, ANN and LR	Background removal and feature extraction	24 hours	IRS, PIRS	1	3
[49]	Compared KNN, ANN, LR, K*, C4.5, SVM and 0-R	Background removal and feature extraction	-	IRS, PIRS	1	3
[40]	ANN	HOG feature extraction	3727	Fused IRS & camera	-	-
[50]	Compared DT and DNN	Subtracted mean room temperature	1000	DRS, IRS	1	1
[65]	No ML	Compared 2 image processing algorithms	923	IRS	1	3
[65]	Compared LR, SVM, KNN, DT, RF	6 step image processing	3273	IRS	1	3

Abbreviations: K-nearest neighbour (KNN), artificial neural network (ANN), linear regression (LR), K-star (K), support vector machine (SVM), decision tree (DT), random forest (RF), infrared sensor (IRS), Doppler radar sensor (DRS), passive infrared sensor (PIRS)

trained with fused photographic and IR data. Mikkilineni et al. [64] did not utilise ML, instead they demonstrated the performance of the blob detection image processing, applied to data collected by multiple IRS. In addition, they proposed an optimal sensor position algorithm. Abedi et al. [50] compared the performance of a DNN trained with data from a Doppler radar sensor and IRS. Singh et al. [65] compared the performance of image processing algorithms applied to data collected by multiple IRS, installed in a range of positions. Singh et al. also compared the performance of a number of ML algorithms to predict the activity of each occupant.

The limitations of the existing work include [40] which fuses thermal and image data, meaning it introduces privacy issues, such that it is not suitable for a commercial environment. The other works [28], [49], [50], [64], [65] are

developed and evaluated within a single environment, with very low occupant numbers (a maximum of three). All of the studies used small data sets ($1000 - 3700$ samples) and apply high computation image processing techniques. Image pre-processing and ML algorithms need to be considered in terms of the costs and benefits. Where costs include monetary, processing power, storage and power consumption, compared to the benefit of high performance. Two of the existing studies [50], [65] required ICT work stations to execute the image pre-processing and model training, which significantly increases system costs in comparison to a system which is developed and deployed on a single edge device.

None of the state of the art were tested in multiple environments; each of the existing studies train and test their classification model in a single environment and do not demonstrate the generalisation ability of their model to operate within multiple similar environments [28], [40], [49], [50], [65]. This is true even in [28], where the ML model is deployed across multiple buildings. It seems Beltran et al. [28] assume that their model functions optimally in all of the deployment environments. Their work does not demonstrate multi-environment generalisation, so it can not be verified that the HVAC system is only switched off at the appropriate times, i.e. when there are no occupants or occupant comfort level have been met[9].

The performance of the state-of-the-art is shown in Table 2.2. The systems that were developed as edge-based systems are [28], [40], [49] which achieve precision of 77% to 82% and accuracy up to 96%. Two other systems [50] and [65] achieve higher performance, but their classification models have high computation requirements and were developed on high-end workstations. To overcome the limitations and the gaps highlighted in the state-of-the-art, this work will focus on developing a DNN model which is low cost and demonstrates optimal performance. Linked with the limitations on SMEs which may include not having access to high specification computing equipment, the system will be both developed on an edge-based system and deployed on an edge-based system, it will achieve a high generalisation ability which will enable to it demonstrate a high confidence regarding its performance in multiple environments.

[9] It should be noted that it is relatively easy to reduce energy consumption by turning off HVAC systems, instead the aim is to reduce energy consumption by turning HVAC off when it is not required.

TABLE 2.2: Performance comparison with existing systems

System	Classifier	Accuracy (%)	Precision (%)	System hardware
[28]	ANN	-	77.14	Raspberry Pi and Arduino
[49]	K*	-	82.56	Raspberry Pi and Arduino
[40]	ANN	96.00	-	Raspberry Pi and Arduino
[50]	DNN	99.96	-	Work station
[65]	CCA	100	-	Work station

Abbreviations: artificial neural network (ANN), K-star (K), deep neural network (DNN), connected component algorithm (CCA)

2.5 Smart Building Mapping and Simulation Tools

The design, deployment and evaluation of a building monitoring WSN is expensive, complex and time consuming [86], [117]. Additionally, due to Covid-19, there is currently a significant impact on normal commercial building usage which adds further complexity to this research.

The design process includes visiting a building to study its layout and meeting with building users to understand their requirements before defining the network specifications [117]. Network costs are a function of the hardware costs, installation costs, ongoing management costs, maintenance costs and the cost of energy consumed by each node [102], [118]. Hardware costs include nodes and communication components, where node costs can vary significantly from around £30[10] [119], to hundreds or even thousands of pounds each. Installation costs vary significantly dependant upon size and nature of the area to be monitored, the number of nodes, the deployment technique and installation expertise. The ongoing network management costs are dependant upon on how frequently the network needs maintenance and the level of maintenance, for example replacing batteries, or replacing node components such as sensors or processors. Additionally, installation can be further complicated in difficult to access or dangerous environments, for example a power station or manufacturing site.

[10]Based on a Raspberry Pi 0 W and a number of low cost sensors.

Simulation tools can achieve a similar result compared real-life deployments when evaluating WSN [117]. Due to the costs and complexities described above, simulating the WSN deployments within this thesis is the most appropriate method of system development, analysis and evaluation. This methodology will reduce the design time and costs related to hardware, installation, management and maintenance [117], [120], [121]. It will also enable multiple deployments to be compared in a single building and for deployments to be evaluated in buildings which are not currently available for real-life deployments [122].

2.5.1 Existing WSN Mapping and Simulation Tools

There is a significant number of WSN simulation tools [120]. Examples include J-Sim [123], which can be used to simulate sensors to monitor a specific condition, but is very complex to operate and has a significant computational overheads. Another tool SENS [124] is limited since it can only be used to simulate and monitor sound data. Some tools are technology specific, including COOJA [125], which simulates Contiki nodes, or TOSSIM [126] and Prowler [127] which both simulate TinyOS nodes. Many tools focus heavily on a single aspect of the network, for example CNET [128] is focused on the transport and network layers. Similarly, SensorSim [129] is focused on the network protocol stack. According to a recent survey, many of the existing tools include unrealistic, incomplete or inaccurate simulation models [120].

Existing sensor network positioning tools include SOU, which was developed to enable the monitoring of internal temperatures and reduce the measurement uncertainty in large sporting venues [130]. The tool was built to help HVAC engineers to determine the optimal number and position of temperature sensors to improve the performance of the HVAC system. Wu et al. [131] developed a sensor placement optimisation tool for bridge structures to determine the quantity, type and position of sensors along the structure. The tool was built upon an existing 3D simulation tool and utilised strain gauge and accelerometer sensor APIs to building the sensor models. A commercial 3D tool, SPOT, was developed by Persistent Sentinel [132]. This tool can simulate radar or sonar sensor networks deployed in external environments, assessing the effect of terrain or obstacles on the network's coverage and performance. There has also been limited attempts to model and simulate indoor environments,

including [133], which focused on modelling radio propagation and communication protocols.

A number of CCTV-based tools focus on the physical position of image-based sensors and the resulting coverage. This includes a 3D CCTV deployment tool developed by Mitsubishi [134] which creates layout plans for different environments. Similar Commercial CCTV positioning tools include JVSG CCTV Design Software [135], VideoCAD [136] and D-Link Surveillance Floor Planner [137]. These tool are all limited to positioning and do not include sensor models or simulate monitoring data.

To the best of the author's knowledge, there are currently no Open Source smart building mapping and simulation tools that meet the two requirements of this research. The first requirement is to determine sensors positions based on building layout and deployment pattern. The second requirement is to simulate the corresponding sensor data based on occupant movement in the modelled building. Due to this gap, MIoTs a smart building mapping and simulation tool is proposed.

2.5.2 Proposed WSN Mapping and Simulation Tool

The MIoTs tool will generate a number of outputs which include a 2D model of a building environment/layout, 2D placement of sensors in the modelled environment, simulation of the movement of occupants around the environment and resulting data captured by a range of deployed multimodal sensor models. The development and evaluation of MIoTs is detailed in Chapter 3. The development life cycle of MIoTs was defined in [117]. The tool was evaluated using the process established for SOU [130] which included comparing the simulated data against data collected in a number of real deployments. MIoTs mapping and simulation tool is used throughout this thesis to simulate the deployment of a variety of WSNs. Following the assessment of this thesis, the code for MIoTs will be made publicly available.

2.6 Chapter Summary

In this Chapter, the motivation for the thesis, background theory, enabling technologies and existing literature are introduced. Due to the environmental impact of GHG, nations are taking action to reduce energy

consumption. More than 7% of all energy consumed within the UK is consumed in small commercial buildings and the literature review highlights that these buildings are mostly ignored by government legislation, government policy and academic research. Further to this, the literature review highlights that SMEs, who are main the users of these buildings experience significant barriers trying to reduce their energy consumption. The motivation for this thesis is to enable SMEs overcome these barriers and reduce their energy consumption. To achieve this, an EMS is proposed in later chapters. Existing EMS which have been developed for domestic buildings or larger commercial buildings are unsuitable for small commercial buildings due to significantly different requirements. The gaps in the state-of-the-art EMS developed specifically for small commercial buildings include systems designed to manage heating systems which are not typical to small commercial buildings; being developed on expensive specialised workstations; based on very small occupant numbers and only being tested in a single indoor environment.

In this chapter three of the enabling technologies; WSN, ML and simulation tools were introduced. In the following chapters a simulation tool is developed and evaluated, enabling it to be utilised throughout this research for system development and assessment. This tool enables a significant reduction in the research cost and complexity. Next, existing techniques to model an indoor sensor network with wireless communication protocols are studied and analysed. Additionally, DNN-based data analysis is developed and integrated into IoT nodes to improve occupancy prediction based on real-time building data. These contributions are combined to create an occupant-centric energy management system.

In the following chapter the simulation tool is introduced.

Chapter 3

Smart Building Mapping and Simulation Tool

3.1 Overview of Smart Building Mapping and Simulation Tool

The smart building, floor plan **M**apping **IoT** **s**imulation tool, referred to as MIoTs, has been developed. MIoTs simulates a building that is monitored by a WSN deployed throughout the building based on a number of predefined sensor node deployment patterns that are detailed in Chapter 3. It also simulates occupants throughout the building, and models the behaviour of a range of multimodal sensors to capture the building data.

MIoTs was developed and evaluated using the process illustrated in Fig. 3.1, similar to that defined in [117]. The sensor nodes and the WSN deployments are based on the contribution of Chapter 3. MIoTs has been developed in Python using a number of libraries including Numpy, SciPy, Matplot, and Pillow and implements a number of purpose-built classes which include `Room`, `Person`, `Sensor_network` and `Measure`. The `Room` class defines the size of the building, internal rooms and the initial room conditions such as the concentration of CO_2 and the level of ventilation. The `Sensor_network` class defines the physical attributes of the sensor network including the physical deployment position and coverage range and of the three different types of sensors. The `Measure` class defines the behaviour of the three different sensors and the creation of the corresponding sensor measurement files. The `Person` class defines the effect of a person in the simulation, including the effect of the individual's breathing on the number of CO_2 particles, the heat map of the individual and the movement of the individual in a room, and in

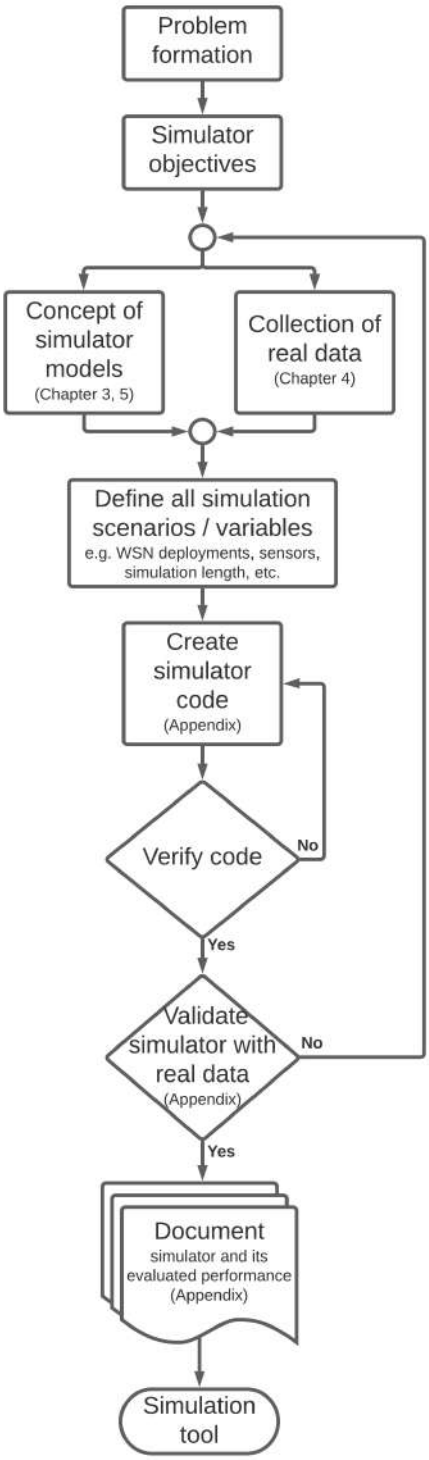


FIGURE 3.1: Process of designing and evaluating simulation tool

or out of a room. The user can define the duration of time to simulate over, the size and layout the building and each of the internal rooms, the rate of ventilation, including the choice of natural or mechanical ventilation, the WSN deployment pattern, which types of sensors to deploy, the

characteristics of the sensors and the frequency of sensor data collection. Regarding building occupants, the user can vary the behaviour of the people in the building including in and out times, for example 8am and 5pm, the maximum number of people and the probability that the people will move around in a room, in and out of the rooms, or in and out of the building.

For each simulation, MIoTs creates a number of output files including:

- A single 2D image file showing the building floor layout and the sensor network deployment, including each sensor's position and coverage area.
- Two .csv files containing ground truth information which is updated at each measurement time interval. The first file maintains a count of the number of people in each room and the second file contains the Cartesian co-ordinates of each person in the building.
- A .csv file for each different type of sensor. This file includes the Cartesian co-ordinates and coverage range of all of the sensors. It is also updated at each measurement interval with the data these sensors collect.
- If IRS are deployed, for each IRS, a black and white thermal image file is created at each measurement interval. These image files are contain a timestamp and sensor ID,

MIoTs is defined in Sections 3.2 to 3.4 and evaluated in Section 3.5. The modelling accuracy of MIoTs is evaluated by comparing the simulated data against real data collected in a number of environments.

3.2 MIoTs User Interface

The user runs the simulation directly from the MIoTs python code. Many aspects of the simulation can be defined by the user, including the building, the sensor network, the number of building occupants and frequency of their movement. Fig. 3.2 shows an example of the code to create a building containing three rooms, including the position and size of the individual rooms. Fig. 3.3 shows the code to create a sensor network, including the sensor types, their specifications and deployment pattern.

```
### BUILDING WITH 3 ROOMS
room = []
r=Room([0,0,15,15],[0,2,2,0])
room.append(r)
r=Room([0,0,15,15,],[3,6,6,3])
room.append(r)
r=Room([20,20,30,30],[0,3,3,0])
room.append(r)
r=Room([20,20,30,30],[4,6,6,4])
room.append(r)
# The building boundaries
builiding=Room([0,0,12,12],[0,10,10,0]) # The building is not appended to the room list
```

FIGURE 3.2: MIoTs Code: Defining building layout

```
# Creation of the sensor network, packing_scheme='square' or 'hexagonal', overlapped = 'True' or 'False'
CO2_sensor_network = Sensor_network(sensor_type='CO2', sensor_range=5, packing_scheme='hexagonal', overlapped=False)
for i in range(0, len(room)-1):
    CO2_sensor_network.deploy(room, i)
for i in range(0, len(room)-1):
    room[i].draw_on_ax_sensor_CO2(ax)

PIR_sensor_network = Sensor_network(sensor_type='PIR', sensor_range=5, packing_scheme='hexagonal', overlapped=False)
for i in range(0, len(room)-1):
    PIR_sensor_network.deploy(room, i)
for i in range(0, len(room)-1):
    room[i].draw_on_ax_sensor_PIR(ax)

IR_sensor_network = Sensor_network(sensor_type='IR', sensor_range=5, packing_scheme='square', overlapped=False)
for i in range(0, len(room)-1):
    IR_sensor_network.deploy(room, i)
for i in range(0, len(room)-1):
    room[i].draw_on_ax_sensor_IR(ax)
plt.show()
```

FIGURE 3.3: MIoTs Code: Defining sensor network

Fig. 3.4 shows the user defined parameters for the building occupants, defining their movements, the simulation duration and measurement frequency. The specification of each of these parameters are optional since they all have default values. The parameter definitions and default values are detailed below:

- *initial_n_people* (integer): Initial number of people inside building at simulation start. Defaults to 0.
- *n_people_min* (integer): Minimum number of the people in the building. Defaults to 0.
- *n_people_max* (integer): Maximum number of people in the building. Defaults to 200.
- *moving_in_room_prob* (float): The probability that a person moves around inside a room. Defaults to $0.5(50\%)$.
- *change_room_prob* (float): The probability that a person moves from one room to other room. Defaults to $0.05(5\%)$.
- *in_prob* (float): The probability that a person enters the building (at *in_time*). Defaults to $0.9(90\%)$.

- *out_prob* (float): The probability that a person leaves the building (at *out_time*). Defaults to $0.9(90\%)$.
- *in_time* (integer): Time from which occupants can enter the building. Defaults to $8(08:00)$.
- *out_time* (integer): Time at which occupants leave the building. Defaults to $16(16:00)$.
- *vent_min* (integer): Minimal level of ventilation. Defaults to 5 for natural ventilation, 60 for mechanical ventilation.
- *vent_max* (integer): Maximum level of ventilation. Defaults to 30 for natural ventilation, 80 for mechanical ventilation.
- *vent_mechanical* (integer): Value 0 defines natural ventilation, 1 defines mechanical ventilation. Defaults to 0.
- *n_in_at_time* (integer): The maximum number of people that can enter to the building at the same time. Defaults to 5.
- *n_out_at_time* (integer): The maximum number of people that can leave the building at the same time. Defaults to 5.
- *random_in_out_prob* (float): The probability of a person random entering or leaving the building, it is a random event, it can happen once per epoch. Defaults to $0.02(2\%)$.
- *n_measure_per_epoch* (integer): The number of measurements per epoch
- *n_epoch* (integer): Number of epochs per simulation, where an epoch is equivalent to 1 hour. Defaults to 48

```
class Simulation:
    def __init__(self,
                 initial_n_people=None,
                 n_people_min=0,
                 n_people_max=5,
                 moving_in_room_prob=0.5,
                 change_room_prob=0.05,
                 in_prob=0.9,
                 out_prob=0.9,
                 in_time=8,
                 out_time=16,
                 n_in_at_time=2,
                 n_out_at_time=2,
                 vent_min=5,
                 vent_max=30,
                 vent_mechanical=1,
                 random_in_out_prob=0.02,
                 n_measure_per_epoch=1,
                 n_epoch=48):
        ...
```

FIGURE 3.4: MIoTs Code: User defined parameters

3.3 MIoTs Sensors

The sensors that can be simulated include a CO_2S which measures the concentration of CO_2 particles in the air, returning number of CO_2 particles per million gas particles, denoted ppm. The next sensor is a PIRS which detects the movement of heat, allowing detection of moving individuals. This sensor returns a 1 when movement is detected, or a 0 when no movement is detected. The third sensor is an infrared sensor (IRS) which detects areas of heat, stationary or moving. For each reading the IRS creates a 32×24 pixel black and white image file. Within this image, areas that have the background temperature are shown with black pixels and areas that are warmer are shown with white pixels. For all three sensors, the file formats and the data that is simulated reflect the output of the real sensors. All of the nodes are positioned on the ceiling of the building, creating a 2D coverage area on the building floor. By default all of the sensors model a circular coverage area as illustrated in Fig. 3.5. There is also the option for the IRS to model a rectangular pyramid coverage volume, which results in a rectangular coverage area on the floor, like that of IRS MLX90640. The dimensions of the rectangular IRS coverage area is given in Eq. 3.1 and illustrated in Fig. 3.6.

$$\begin{aligned} l &= 2r \\ w &= 0.75l = 1.5r \end{aligned} \tag{3.1}$$

where l and w are the length and width of the IRS coverage area and r is its coverage range.

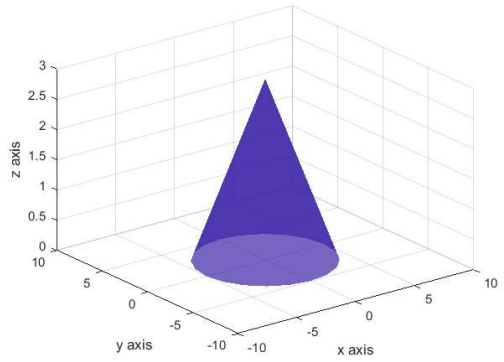


FIGURE 3.5: Default sensor coverage volume

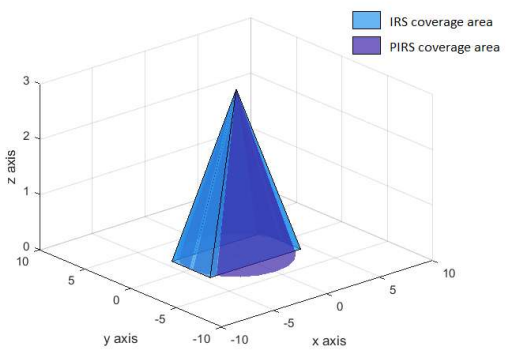


FIGURE 3.6: Rectangular IRS coverage volume

3.4 MIoTs WSN deployments

A number of WSN deployment patterns are available, these include a sensor grid, overlapping sensor grid, sensor hexagons and overlapping sensor hexagons deployment. Each of these deployments are defined in much more detail in Chapter 3. They are all organised deployment patterns that define the physical position and coverage range of the nodes. The sensor nodes are all assumed to have a communication range equal to or greater than their coverage range, also of the networks are based on simple 'q' coverage, such that each area of the building is monitored by a single node. Figs. 3.7 and 3.9 show the layout of the simulated output files and Figs. 3.20, 3.22 and 3.24, show some examples the simulated IR image files.

	B	C	D	E	F	G	H	I	J	K	L	M	N
1	PIR_sensor	room_num	x	y	range	00h00	00h02	00h04	00h06	00h08	00h10	00h12	00h14
2	1	0	4	4	4	0	0	0	0	0	0	0	0
3	2	0	4	10	4	0	0	0	0	0	0	0	0
4	3	1	8	3	4	0	0	0	0	0	0	0	0
5	4	2	8	7	4	0	0	0	0	0	0	0	0
6													

FIGURE 3.7: PIRS output file

	B	C	D	E	F	G	H	I	J	K	L	M	N
1	CO2_senso	room_num	x	y	range	00h00	00h02	00h04	00h06	00h08	00h10	00h12	00h14
2	1	0	4	4	4	790	790	790	790	790	790	790	790
3	2	0	4	10	4	790	790	790	790	790	790	790	790
4	3	1	8	3	4	690	690	690	690	690	690	690	690
5	4	2	8	7	4	760	760	760	760	760	760	760	760

FIGURE 3.8: CO_2S output file

	B	C	D	E	F	G	H	I	J	K	L	M	N
1	room_id	00h00	00h02	00h04	00h06	00h08	00h10	00h12	00h14	00h16	00h18	00h20	00h22
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0
4	2	0	0	0	0	0	0	0	0	0	0	0	0

FIGURE 3.9: Ground-truth room data output file

3.5 Methodology to Evaluate MIoTs Simulator

The simulator's behaviour will be validated by comparing its output data against real data, such that "the conceptual model is an accurate representation of the real system" [117]. The simulated building is based upon a small building with three separate rooms and $0 - 5$ occupants, whose movements are simulated over a period of 24 hours. Sensor and ground truth data is collected once per minute. Occupant in and out times

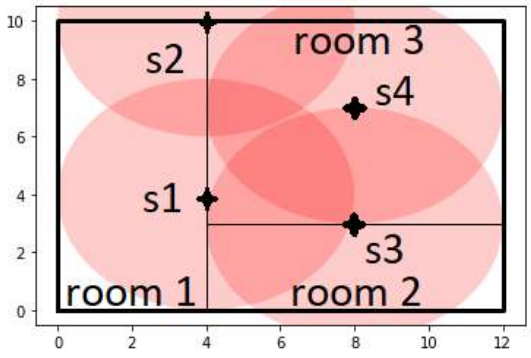


FIGURE 3.10: PIRS deployment

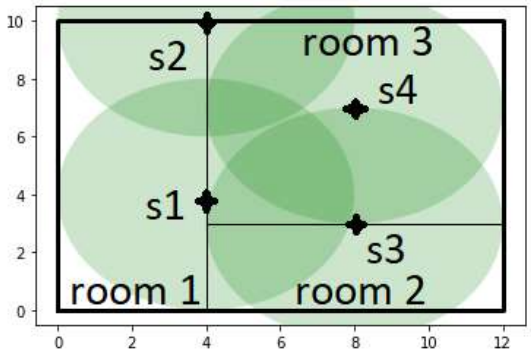


FIGURE 3.11: CO_2S deployment

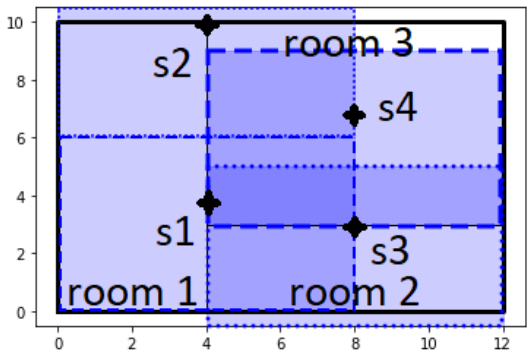


FIGURE 3.12: IRS deployment

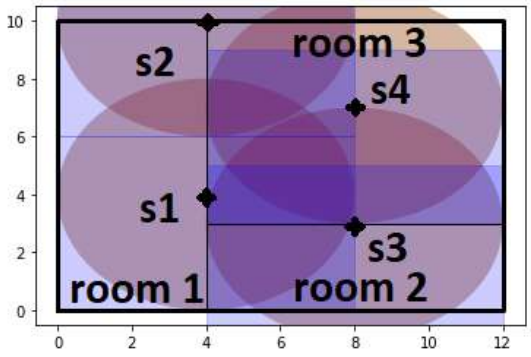


FIGURE 3.13: Combined deployment

are specified as typical working hours, $8:00-18:00$. Real data is collected by the DeNNOTE occupancy detection system which is proposed in Chapter 5. The DeNNOTE system is a single IoT node containing a Raspberry Pi, PIRS, IRS, CO_2S, temperature sensor and a DNN which processes the captured heterogeneous data to determine whether the monitored areas is occupied or not. Within MIoTs, there are three heterogeneous sensor types that can be simulated; PIRS, IRS and CO_2S. Each type of sensor can be given different parameters and deployed in different patterns within a single simulation. During the verification of MIoTs, all three types of sensors are grouped together into a single node, and specified with the same coverage range so that the simulated data resembles that collected by the DeNNOTE system. Room 1 is a simulation of the Office, room 2 is a simulation of the Kitchen and room 3 is a simulation of the Meeting room.

The steps to evaluate the MIoTs simulation tool include firstly, studying the physical deployment of the sensor nodes, secondly, the correlation between the real and simulated data and thirdly, a comparison of the performance that

is achieved when the real and simulated data are processed by the DNN.

Regarding the deployment of the sensor network, MIoTs creates an image which shows a 2D view of the deployment positions and coverage area of the sensor network. Also, the Cartesian co-ordinates of each sensor are given in the sensor output files. This sensor location information will be compared with real positional information. The PIRS and CO_2S are simulated with a conical coverage volume. MIoTs positions the PIRS on the ceiling of the building, which results in a corresponding circular coverage area on the building floor.

When multiple sensors with different coverage areas are deployed together in a single node, this work is concerned with the combined coverage area, i.e., if all of the sensors have the same range, the combined coverage area is assumed to be equal to the sensor that achieves the smallest coverage area, which in this instance is the circular coverage area of the PIRS or CO_2S. The sensor deployment in this verification simulation is based on the sensor grid technique with no overlap, as defined in Eq. 4.6. A significant number of low cost IoT sensors achieve a range of $4m$ based on being positioned at a ceiling height of $3m$ and having a FOV of 110, therefore all of the sensors are defined with this coverage range of $4m$. Based on room dimensions sensors may be placed on the room perimeter meaning there may some sensor coverage overlap.

Next, the correlation between the real and simulated data that is generated by the three types of sensors will be evaluated. For the CO_2S, there will be a visual comparison of the trends in the CO_2 levels due to the number of occupants in a room and whether they stay for a prolonged period. Similarly, for IRS, the real and simulated thermal images will be compared visually. The data generated by the PIRS will be evaluated based on the real and simulated data accuracy. The balanced accuracy (BA) is based on measuring the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) occurrences for the different PIRS. A TP indicates when the PIRS correctly detects the presence of an occupant by recording a '1', TN indicates when it correctly detects the absence of an occupant by recording a '0'. A FP indicates when the PIRS incorrectly indicates an occupant is present when they are not, wrongly recording '1', and similarly, a FN is when it incorrectly indicates the absence of occupants and records a '0'. Generally, PIRS have a low level of accuracy when determining occupant presence. This is because they do not detect occupants, but instead they

detect the movement of a heat source. This means if an occupant is relatively stationary, for example sitting at a desk working, they may not be detected causing the PIRS to return a high number of FN. Finally, all of the simulated data, including the CO_2 data, motion data and the IR thermal images are processed by the DeNNOTE DNN to determine whether occupants are present. For all iterations, the DNN is first trained with real data collected in the Office, then the performance of the simulated data will be compared against performance of data collected in the Office, Kitchen and the Meeting Room.

3.6 Evaluation of MIoTs Simulator

3.6.1 Position of the Nodes in the Network

Figs. 3.10 and 3.13 show three different versions of the MIoTs image file which shows the building outline and its three rooms, the positions and coverage areas of the PIRS, CO_2S and IRS. For clarity, the edge of each IRS coverage area is demarked with different types of dotted lines. The PIRS and CO_2S are shown to have circular coverage areas, in comparison, the IRS has a rectangular coverage area. The coverage of each sensor is only effective within the room where the sensor is located. The coverage overlap that is visible in Fig. 3.10 between PIRS $s1$ and $s2$ (and CO_2S $s1$ and $s2$ in Fig. 3.11) is due to the dimensions of room 1. PIRS $s1$ is positioned at Cartesian co-ordinates $(4, 4)$ and based on the grid deployment, PIRS $s2$ should be positioned at co-ordinates $(4, 12)$, but this is outside of the room, so to reduce the creation of unmonitored areas the grid deployment technique takes this into account and instead PIRS $s2$ is positioned on the room perimeter at co-ordinates $(4, 10)$. The simulated positions of each sensor, denoted $s1 - s4$, are shown as Cartesian co-ordinates in Table 3.1.

TABLE 3.1: Calculated sensor Cartesian co-ordinates

Sensor s1	Sensor s2	Sensor s3	Sensor s4
(4,4)	(4,10)	(8,3)	(8,7)

To evaluate if the simulated sensor positions are correct, the positions are calculated based on knowing the co-ordinates of each room and using the grid technique defined in Section 4.3. Based on the grid technique, each sensor is positioned within the centre of a grid square, where each square

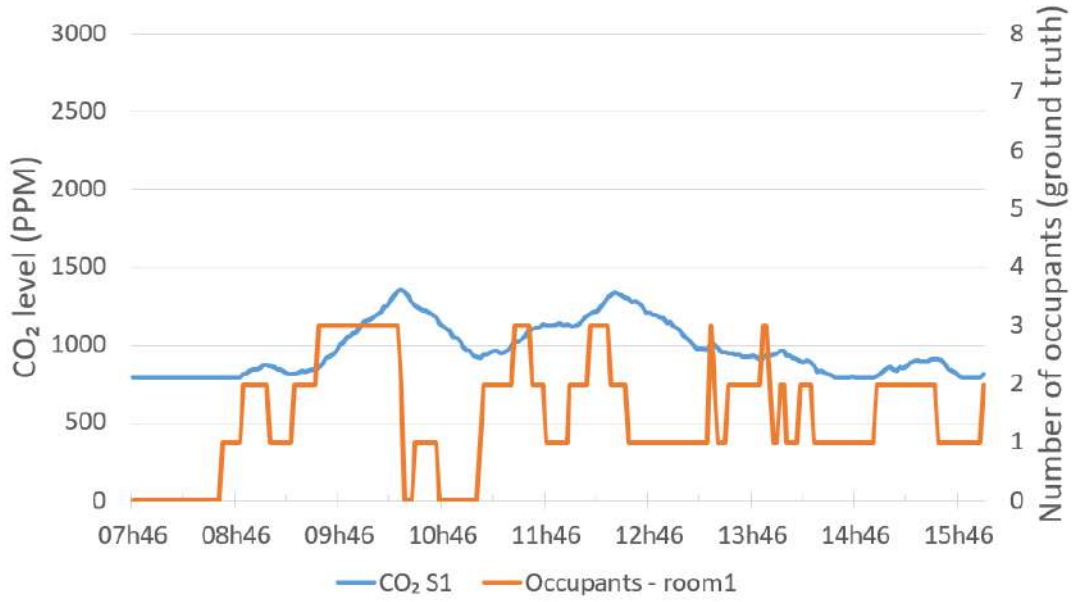


FIGURE 3.14: Room 1: simulated CO_2 data

FIGURE 3.15: Office: real CO_2 data

has a dimension $2r$ and each sensor has a range $r = 4$. Room 1 has x co-ordinates $[0, 0, 4, 4]$ and y co-ordinates $[0, 10, 10, 0]$. The calculated positions are shown in Table 3.1 and the simulated positions are shown in Figs. 3.10 and 3.13. By comparing both, it can be that the calculated and simulated sensor positions are identical.

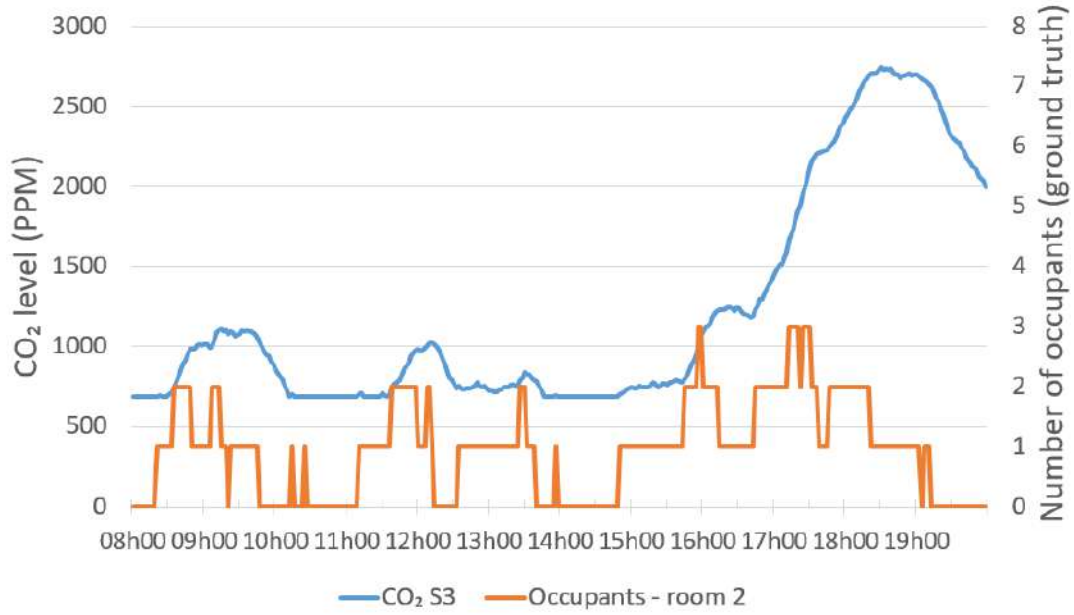


FIGURE 3.16: Room 2: simulated CO_2 data

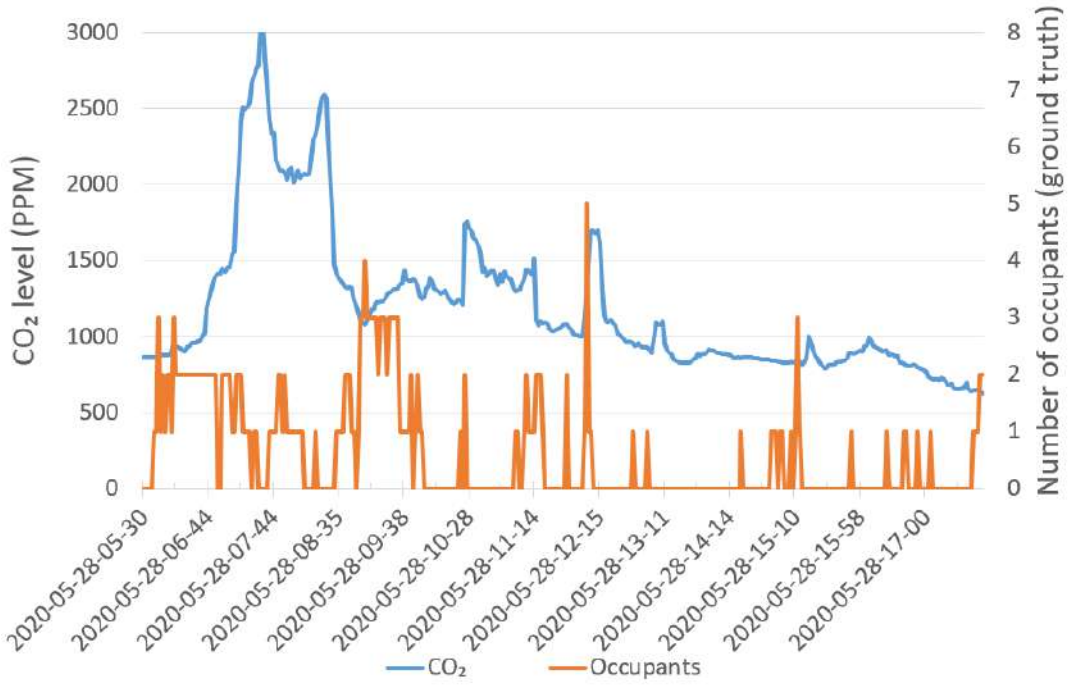


FIGURE 3.17: Kitchen: real CO_2 data

3.6.2 Correlation Between the Real and Simulated Data

Figs. 3.14 and 3.19 show the simulated and real CO_2 data within rooms and corresponding ground truth for the number of occupants in each room. MIoTs has been set with a natural rate of ventilation of $5 - 30\%$ to match the conditions in the Office, Kitchen and the Meeting room which have natural

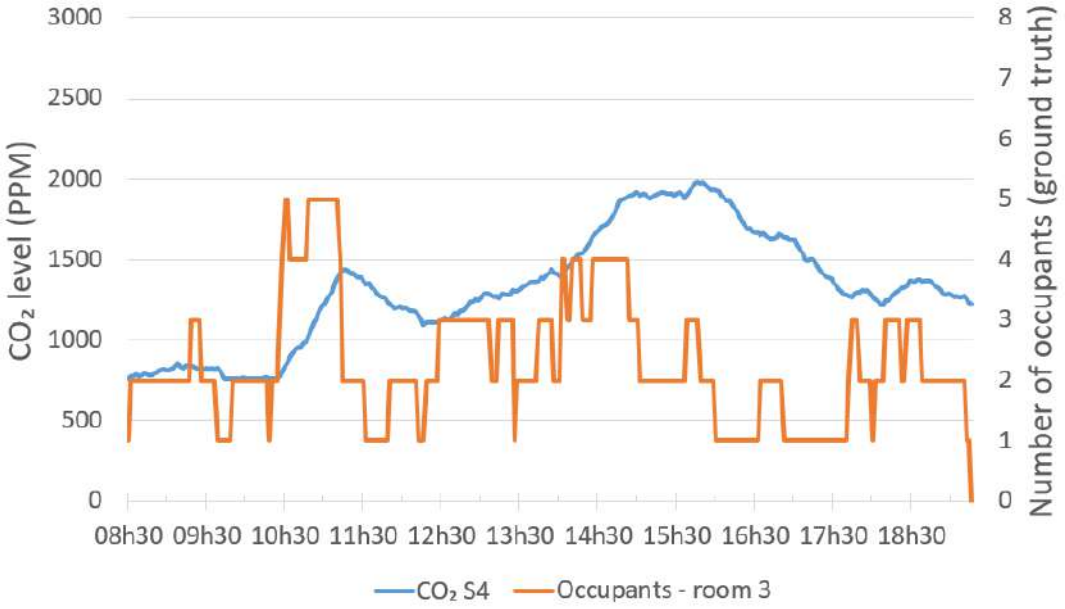


FIGURE 3.18: Room 3: simulated CO_2 data

FIGURE 3.19: Meeting room: real CO_2 data

ventilation not air conditioning. It can be seen that the variation in the CO_2levels based on occupancy is similar across both the real and simulated data. These similarities include the lag between occupant numbers changing, the rate at which the CO_2 levels increase when the number of occupants increase, or when occupants remain in the room for a prolonged period. In comparison, the level of CO_2decreases when the occupant

number is reduced or occupants vacate the room entirely, returning to $600 - 800$ppm when the room is vacated for 30 minutes or longer.

The lag between occupants entering a room and CO_2 levels increasing is approximately 15 minutes in both the real and simulated data. When occupants remain in a room, in both the simulated and real data, it can be seen that the CO_2 levels continue to rise. In the smaller rooms, for example the Kitchen, or room 2, it can be seen that the CO_2 level does rise much more quickly and reach higher levels. When the levels of CO_2 in a room reach XX, the room can feel uncomfortable and stuffy. This generally causes occupants to take action to increase the level of ventilation in these rooms, the occupants response is likely to include opening doors or windows which can cause the $CO_2$to decrease more quickly, as can be seen in the Kitchen, around $8.35am$. It is noted that the simulated data is slightly smoother than the real data.

Figs. 3.20 and 3.25 show the black and white 32×24 pixel images that are created by the simulated and real IRS respectively. The images are labelled as either “negative" which indicates that there are no occupants within the monitored area or “positive" which indicates there are occupants. The images show areas of heat as white pixels within a black image. In these thermal images, a person appears as a well defined area of white that covers an area just under $1m^2$. When an occupant has recently left a room, they leave behind a small area of warm air causing a small thermal hot spot as shown in Fig. 3.23. This warmth dissipates over a few minutes, causing the thermal hot spot to also decrease in size until eventually it is gone, as shown in Fig. 3.21.

The performance of the PIRS data that was captured in a number of different real environments over a period of a month was assessed based on using the data to indicate the presence of occupants. The PIRS demonstrated BA of 69.26%. This rate is relatively low because PIRS only detect occupants when they move, unlike IRS which detects stationary or moving occupants. The simulated data for the PIRS consistently demonstrated a similar level of performance to the real sensor. For the data simulated in room 1, the PIRS achieved a BA of 69.88%, room 2 of 69.64%, and room 3 of 72.03%. The slight increase in performance of the PIRS within room 3 was a result of increasing the rate of movement within and in/out of a room for room 3. This resulted in 45 movements of occupants in or out of the room over 24 hours, compared with room 1 which had 29, and room 2 which had 38. The MIoTs setting that

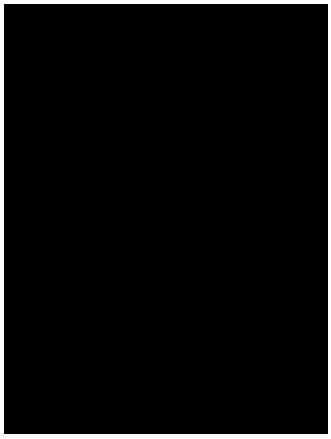


FIGURE 3.20: Simulated negative image

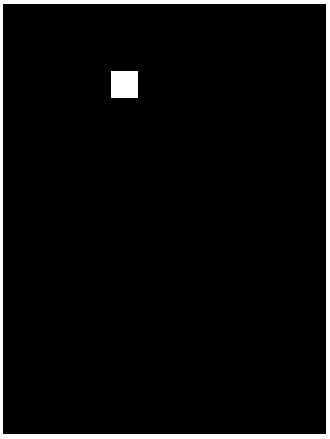


FIGURE 3.22: Simulated negative image

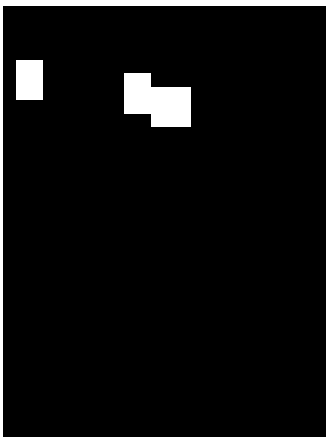


FIGURE 3.24: Simulated positive image

FIGURE 3.21: Real negative image

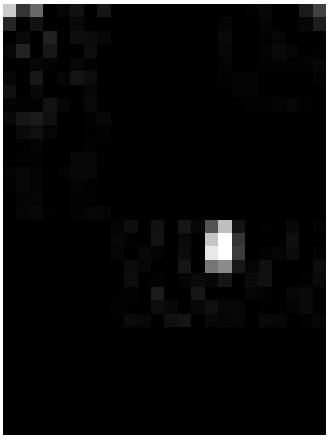


FIGURE 3.23: Real negative image

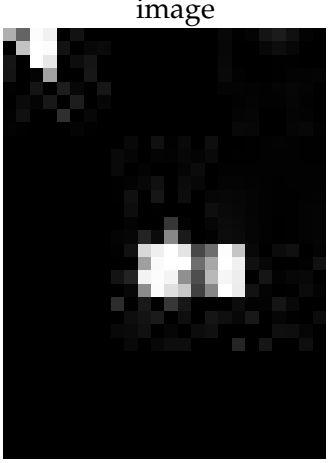


FIGURE 3.25: Real positive image

adjusts the level of occupant movement for each room was selected to mimic the level of occupant movement within the corresponding real rooms.

3.6.3 Performance of Real and Simulated Data

The simulated data collected at each time instance is processed by the DeNNOTE DNN to determine whether occupants are present. The DNN has first been trained with data collected in the Office and evaluated with data collected in the Office, Kitchen and Meeting room.

Table 3.2 shows the performance of the simulated and real data, where the data collected in the Office and Meeting room achieves a high AUC over 90%. Similarly, the simulated data for the corresponding rooms, room 1

TABLE 3.2: Performance comparison of simulated and real data processed by DNN

Simulated Environ	DNN processing simulated data: AUC (%)	Real Environ	DNN processing real data: AUC (%)
room 1	98.39	Office	95.65
room 2	94.70	Meeting	92.59
room 3	85.64	Kitchen	72.21

and room 2, also returned a high AUC. Though the simulated data performed very slightly better, achieving approximately 2% better performance. It is expected that the simulated data will perform slightly better than the real data due to the simulated data containing less noise. This is noted in the CO_2which is slightly smoother than the real data, in the simulated IR images, the heat images are slightly crisper around the edges. For real data collected in the Kitchen, the performance that was achieved was much lower than the other rooms, around 70%. This reduced performance is due to the food warming equipment which distorted the thermal images and caused a higher rate of FP. Room 3 was simulated to reflect the Kitchen, particularly with respect to the room size, rate of ventilation, number of occupants and level of occupant movement, but the simulation did not include the food warming equipment. This explains why the simulated data collected for room 1 achieved a lower performance than rooms 2 and 3, but significantly better than the performance of data collected in the Kitchen.

3.7 Chapter Summary

Due to costs and complexities involved in deploying a WSN, a smart building mapping and simulation tool was developed. This tool is written in Python and enables a 2D WSN deployment to mapped and the sensor data to be simulated. The functionality of the tool is introduced along with its user interface, user-defined parameters and the sensors whose behaviour are simulated. To enable the simulation tool be used in throughout this research, it was evaluated using real building monitoring data. The simulated sensor data was compared to data that was collected in three real environments. For all three sensors, the simulated data was very similar to that generated in real life. MIoTs does include the generation of some sensor

noise, but it was less than the noise exhibited in the real data. As a result, the simulated data is smoother and achieves a slightly higher rate of accuracy, about 2%, than the corresponding real life data. Due to these strong evaluation results, the MIoTs smart building simulation tool is used in the evaluation of other areas of this thesis.

In the following chapter a number of existing techniques for the deploying of sensor networks are introduced and assessed in terms of hardware cost and the coverage of an indoor environment.

Chapter 4

Wireless Sensor Network Deployments for Building Monitoring

4.1 Introduction

Within commercial buildings, occupant monitoring and comfort level monitoring can be used to inform the actions of an EMS. Historically, EMS used a combination of pre-programmed building usage models and data collected by a small number of sensors to predict this data. Due to out-of-date models and limited sensor data, this often resulted in inaccurate information, which resulted in ineffective EMS.

Instead, multimodal building data such as temperature, light levels and occupancy data can be accurately measured in real-time, or near real-time, by IoT-based distributed WSN [102]. An IoT-based WSN is comprised of tactically positioned multimode IoT sensors which utilise data fusion techniques to combine the different data streams. The physical positioning of the sensors is significant in relation to the area coverage that is achieved [102]. There are two main types of coverage; area and target coverage. Area coverage is the monitoring of an area or region of interest. In comparison, target coverage is the monitoring of a specific mobile or stationary target within a given area [138]. Inside a building, a target may be a high-value object, for example, artefacts displayed inside a museum. Area coverage can be broken down further into weak and strong coverage [99], [139]. Weak coverage is defined as an area which is monitored by sensors but also includes regions with coverage gaps. An area of weak coverage can, in theory, be crossed without detection though it is difficult to cross

since the gaps are not easily identifiable. An example of weak coverage is deploying sensors only across windows or doorways or randomly within a space. Strong coverage is defined as an area where sensors are used to monitor the entire area, such that the area can not be crossed without detection. An example of strong coverage is monitoring all of a building's floor space. In environments where a high performance is required, target, weak and strong coverage techniques can be combined. Different types of deployments require different amounts of hardware including but not limited to sensors, processing units, communication units and power supplies. This means that different sensor network deployments can vary significantly in performance and cost.

The contributions of this chapter is the calculation and comparison of sensor density and space coverage for a number of existing WSN deployments.

In this chapter, the problem of occupancy detection and building monitoring will be studied through six different strong coverage sensor deployment techniques. Equations and algorithms are proposed in Section 4.4 and Section 4.5 to determine the sensor density and coverage respectively. In Section 4.6, two deployments that include sensor coverage overlap are examined. In Section 4.7, using simulated building data, the deployment techniques are analysed and evaluated to determine which strong coverage sensor deployment technique can deliver the optimal wireless sensor network, to achieve accurate building monitoring data at low cost.

4.2 Preliminaries of WSN Deployment Techniques

4.2.1 Strong and Weak Coverage

The combination of weak and strong coverage deployments are regularly used to monitor commercial buildings. In such deployments, weak coverage is commonly deployed in areas that are only accessed for short periods, including doorways, corridors and public toilets. Motion detection sensors, e.g. PIR or cameras, may be used to detect an occupant entering the area. This event may be used to automate building services, for example opening the door or controlling the light level. In the same building, a strong coverage deployment may be used in areas where occupants dwell or there are multiple entrances and exits, for example, public spaces, cafes

or work areas. Strong coverage might be achieved using RF, gas, noise, temperature, heat and humidity sensors or a combination. These sensors can detect occupants, monitor comfort levels and automate the control comfort levels when occupants are present and automate turning down or off systems once an area is vacated.

4.2.2 Existing Physical Deployment Techniques

Packing is the mathematical method of filling a given area with the maximum number of circles, without any overlap and minimal vacant areas [140]–[142]. This method is commonly used in logistics to fit the maximum number of solid cylindrical containers, for example food tins, into a rectangular shipping container.

There are two well-accepted methods, square and hexagonal packing. For square packing, the area to be packed is divided into virtual squares, each identical in size, vertically and horizontally aligned. Next a circle is placed inside each square of the grid with its centre also being the centre of the square. The dimensions of the squares are $(2R)^2$, where R is the radius of the circles to be packed. The circle's perimeter meets the perimeter of the square at the middle of each of the squares' vertices. This square packing technique is commonly applied as a two dimensional (2D) view of indoor space monitoring [102], [103] which will be referred to as the sensor grid deployment, where the floor is divided into virtual squares which will each be monitored by a single sensor node. A sensor node is comprised of m sensors with a sensing radius range of r and a horizontal cumulative horizontal angle of 360°. When positioned on the ceiling directly above the centre of each square, these nodes create a spherical cone-shaped coverage volume, such that it results in a 2D node coverage area on the floor. This 2D node coverage area is given in Eqs. 4.1 to 4.3 and illustrated in Fig. 4.1. The sensor grid technique is illustrated in Fig.4.2 as deployment (a).

Hexagonal packing is an alternative method of packing circles into a rectangular space [140]–[142] and it begins in a similar way to square packing; the first column of circles are placed along the perimeter on the y-axis of the space. The next column of circles are moved vertically down by distance r with respect to the first column and positioned beside the first column, as close as possible. The third column is shifted back up distance r, so as to be horizontally in-line with the first column. These steps are repeated until the space is filled. This is referred to as hexagonal packing

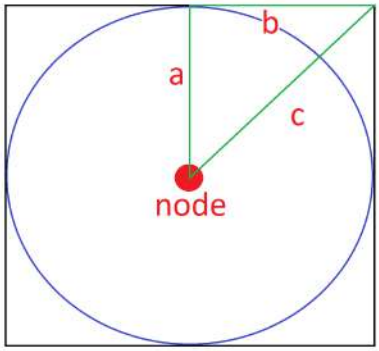


FIGURE 4.1: 2D sensor coverage area

$$l_S = 2a \tag{4.1}$$

$$A_S = (l_S)^2 = (2a)^2 \tag{4.2}$$

$$a = r$$

$$\therefore A_S = 4r^2 \tag{4.3}$$

since each circle will touch six other circles or the edge of the container, creating repeating hexagonal patterns. This hexagonal packing technique can also be utilised in sensor deployments and will be referred to as sensor hexagons. Similar to the sensor grid, the sensor nodes are placed on the ceiling in a hexagonal pattern which create 2D node coverage circles positioned in a hexagon pattern on the floor below, illustrated as deployment (c) in Fig. 4.2.

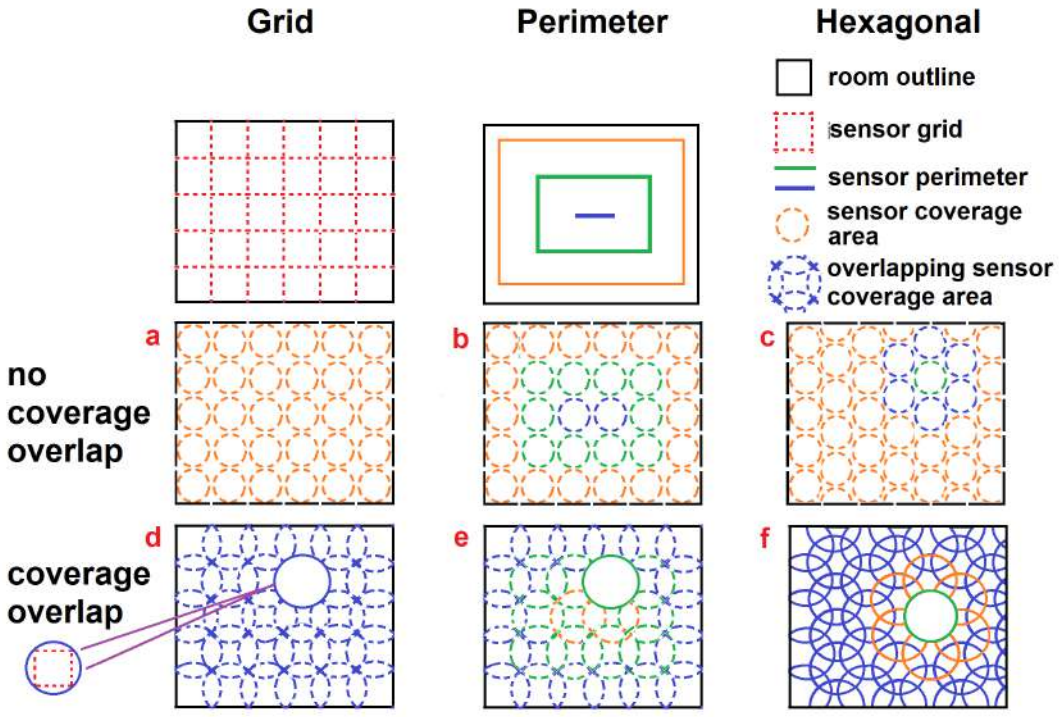


FIGURE 4.2: Overview of six proposed sensor network deployment techniques

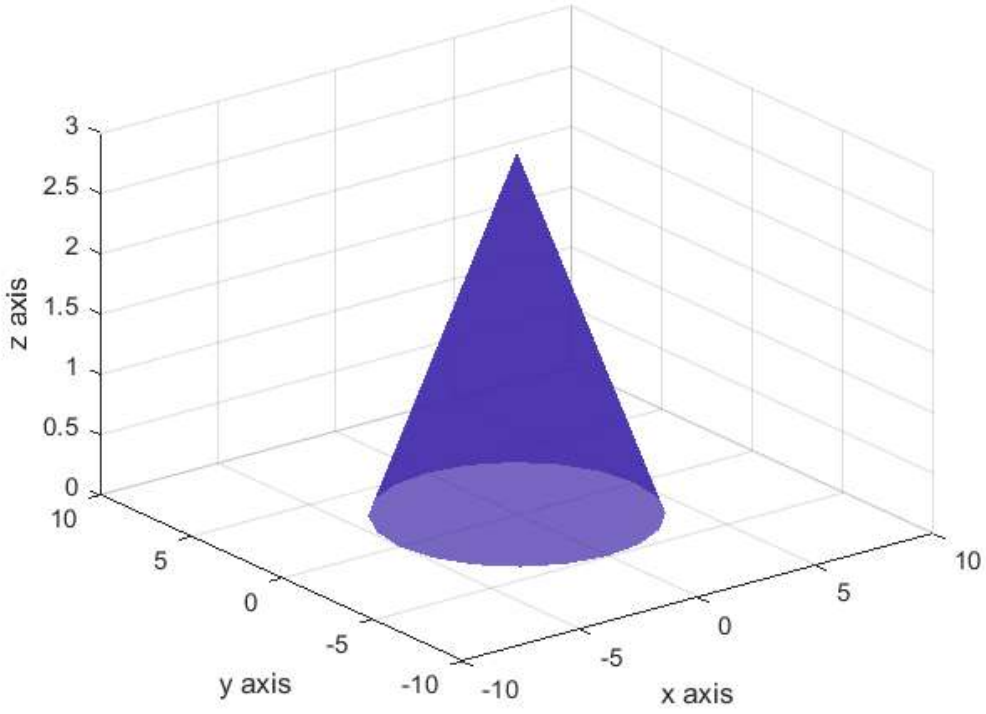


FIGURE 4.3: Node coverage based on being positioned on building ceiling

In a single sensing node there may be one or more sensors. The coverage area and range of each sensor are defined by their specifications. Some sensors have a 360° sensing angle, otherwise known as a field of view (FOV). In comparison, other sensors, for example, PIRS typically have a FOV between $90 - 270°$. To overcome these differences, it will be assumed that each sensor, or a number of sensors in a single node that are monitoring the same attribute, will together create a conical coverage area. Such that the conical apex is positioned at the sensor node and the circular coverage area will be positioned on the perpendicular surface. Throughout this work, the sensor will be positioned on the ceiling and the corresponding coverage circle will be positioned on the floor, as shown in Fig. 4.3. The coverage range of sensors can vary significantly and when multimode sensors are combined in a single node, it will be assumed that the node's coverage range is equal to that of the sensor with the minimum sensing range.

4.3 Applying Packing Techniques to Develop Strong Coverage WSN Deployment Techniques

The proposed deployment techniques are based on the square packing and hexagonal packing techniques. The focus will be on the physical positioning of sensors and the resulting area coverage, rather than logical positioning or connectivity. The proposed deployments are based on environment monitoring within internal spaces. Therefore, simple coverage, such that each building area is covered by one or more sensors, is considered sufficient, instead of more complex solutions such as k-coverage or q-coverage [98]. Additionally, it can be assumed that the areas being monitored are sufficiently small that all sensors have a communication range, r_c, that is equal to, or greater than the sensing range radius r, such that $r_c \geq r$. This means each sensor can transmit packets to other sensors or directly to the base station [143].

This work proposes a second variation of the sensor grid technique, using multiple virtual perimeters rather than a grid. The perimeters are created perpendicular to the exterior walls of the space and nodes are positioned along them, as shown in Fig. 4.2, deployment (b). This technique will be referred to as sensor perimeters. These three identified techniques define the diameter of the node coverage circles as equal to the length and width of the original grid square, which results in a deployment technique without any node-to-node coverage overlap. In comparison, the next three proposed techniques do have node coverage overlap. The coverage overlap occurs by changing the ratio between the coverage circle and grid square, such that the circle encompasses the virtual grid square and the square is the maximum size that can fit wholly inside the circle. When this new ratio is applied to the sensor grid, sensor perimeters and sensor hexagons, the results are three new deployment techniques: namely the 'overlapping sensor grid', shown in Fig. 4.2, deployment (d); 'overlapping sensor perimeters', shown in Fig. 4.2, deployment (e); and 'overlapping sensor hexagons', shown in Fig. 4.2, deployment (f). All six of the sensor deployment techniques will be discussed further in the following sections.

4.4 Node Density

4.4.1 Sensor Grid Node Density

Within regularly shaped area e.g. internal rooms, specifically square and rectangular spaces, the application of the sensor grid and the sensor perimeters technique will create identical sensor deployments. The two techniques may create different deployments in more complex-shaped areas or areas with obstacles. The sensor perimeters technique offers greater flexibility, compared to the sensor grid, to deploy a reduced number of sensors. For example, deploying a single sensor perimeter as a weak barrier, rather than achieving strong coverage throughout the whole area. As such, a reduced number of sensor perimeters could be deployed instead of the k perimeters that are required for strong coverage. In this scenario, the additional sensors which would have been deployed up to perimeter k could instead be deployed elsewhere, reducing the WSN cost and complexity.

The steps to determine the sensor density for a square or rectangular shaped space deployed with the sensor grid technique are shown below, followed by Eqs. 4.4 to 4.6.

- determine constant λ, which defines whether to place an additional sensor overlapping the space perimeter
- determine the number of sensors, based on space length, width, λ and sensor range radius.

$$\begin{aligned} &\textbf{if } (2ir \geq l > (2ir - r)) \textbf{ then} \\ &\quad \lambda = 0 \\ &\textbf{else} \\ &\quad \lambda = (2ir - l) \end{aligned} \tag{4.4}$$

$$S_{SG} = \; m \left\lceil \frac{l + \lambda}{2r} \right\rceil^2 \tag{4.5}$$

$$S_{RG} = \; m \left\lceil \frac{l + \lambda_{\mathrm{l}}}{2r} \right\rceil \left\lceil \frac{w + \lambda_{\mathrm{w}}}{2r} \right\rceil \tag{4.6}$$

where S_{SG} is sensor density when deploying a sensor grid in a square-shaped space, S_{RG} is sensor density when deploying a sensor grid in a rectangular-shaped space, m is the number of sensors in each node, $\lceil \; \rceil$ the absolute value of the figure inside, l and w are the space length and width, r is the node range radius, λ, λ_{l} and λ_{w} are constants.

The node density of some irregularly shaped spaces, for example, a space which is a right-angled polygon is comprised of one or more rectangular areas, which means such a space can be split into the minimum number of rectangles and Eq. 4.6 can be applied, with an additional step. The additional step is to deduct any nodes which are deployed in overlapping areas, so as to not count them twice. The sensor density of this deployment is given as Eq. 4.7.

$$S_{PG} = \; m \sum_{i=1}^{i=shapes} \left\lceil \frac{l_{\mathrm{i}} + \lambda_{\mathrm{li}}}{2r} \right\rceil \left\lceil \frac{w_{\mathrm{i}} + \lambda_{\mathrm{wi}}}{2r} \right\rceil - m \sum_{j=1}^{j=overlaps} \left\lceil \frac{lo_{\mathrm{j}} + \lambda_{\mathrm{loj}}}{2r} \right\rceil \left\lceil \frac{wo_{\mathrm{j}} + \lambda_{\mathrm{woj}}}{2r} \right\rceil \tag{4.7}$$

where S_{PG} is the sensor density when deploying a sensor grid in a polygon-shaped space, l_{i} and w_{i} are the length and width of shape i, λ_{li} and λ_{wi} are the constants for shape i's length and width, lo_{j} and wo_{j} are the length and width of overlapping shape j, λ_{loj} and λ_{woj} are the constants for shape j, *shapes* is the (minimum) number of rectangles the space can be split into, *overlaps* is the number of overlapping areas.

If the shape of the space gets more complex, or nodes need to re-positioned to avoid obstacles, the application of Eq. 4.7 also gets more complex, requiring a new calculation for each sub-shape. Instead, for such spaces, the sensor perimeters technique is more appropriate since this technique determines the number of nodes in each perimeter, one perimeter at a time, irrespective of the size or shape of the rest of the space.

4.4.2 Sensor Perimeters Node Density

The sensor perimeters technique involves the creation of sensor perimeters which will consist of multiple nodes, whose coverage area will meet that of

neighbouring nodes, resulting in an unbroken perimeter of sensor coverage. The sensor perimeters denoted as P_{i}, are created by starting at Cartesian coordinates (0,0) of the space perimeter, for example, the interior walls, then moving a distance equal to the node sensor range, r and starting to create a virtual perimeter parallel to the interior wall, this virtual perimeter stops distance r from the next wall or an obstacle. Then, turning parallel to the next section of the interior wall (or obstacle), the next segment of the perimeter is created. These steps are repeated for all interior edges of the space until a virtual closed perimeter is created. Along this new perimeter, P_1, nodes are then positioned, starting at co-ordinates (0+r,0) and separated by distance referred to as 'node spacing', denoted N_{s}. To create an unbroken perimeter, the node spacing should be a minimum distance r and a maximum distance of $2r$. If the node spacing is equal to the distance r, there is some overlap of node coverage areas, as illustrated in Fig. 4.2, deployment (e). This deployment is referred to as overlapping sensor perimeters. If the node spacing is equal to distance $2r$, the next node's coverage area meets that of the previous node without any overlap, as illustrated in Fig. 4.2, deployment (b). This deployment is referred to as sensor perimeters.

Next, staying parallel to the first sensor perimeter P_1, but moving distance N_{s} towards the centre of the space, a second sensor perimeter, denoted P_2, is created. If distance N_{s} from the previous sensor perimeter is beyond the centre of the space, no further perimeters will be created. If the second perimeter is created, then nodes are positioned along it. This process is repeated, creating a total of k sensing perimeters, the inner-most sensing perimeter will be denoted P_{k}. The k perimeter, or partial perimeter, may be positioned right at the centre of the space. Eq. 4.8 is the number of perimeters. The sensor density for sensor perimeters deployed in a square-shaped space is given by Eq. 4.9.

$$k = \left\lceil \frac{l - 2r}{2N_{\mathrm{s}}} \right\rceil, \; for\, k \in \mathbb{N} \tag{4.8}$$

$$S_{SP} = 4m \sum_{p=1}^{p=k} \left\lceil \frac{l - 2r(2p-1) + \lambda}{2r} \right\rceil \tag{4.9}$$

$$\textbf{for}\, l == (2r(2k-1) + \lambda) : \; S_{SP} = S_{SP} + m$$

where k is the number of sensor perimeters, N_s is the node spacing, S_{SP} is the sensor density when deploying a sensor perimeter in a square-shaped space, p is the current sensor perimeter. Note the additional line of Eq. 4.9, such that if the length of the space is equal to $(2r(2k-1)+\lambda)$, an additional node is placed at the centre of the space.

An equation can be derived for other regularly shaped internal spaces, including rectangular, circular and right-angled polygons. The node density of a rectangular space is given in Eqs. 4.10 and 4.13.

$$S_{\mathrm{Rl}} = m \sum_{p=1}^{p=k} \left\lceil \frac{l - 2r(2p-1) + \lambda_{\mathrm{l}}}{2r} \right\rceil \tag{4.10}$$
$$\textbf{for}\, l == (2r(2k-1)+\lambda_{\mathrm{l}}) : S_{Rl} = S_{Rl} + m$$

$$S_{\mathrm{Rw}} = m \sum_{p=1}^{p=k} \left\lceil \frac{w - 2r(2p-1) + \lambda_{\mathrm{w}}}{2r} \right\rceil \tag{4.11}$$
$$\textbf{for}\, w == (2r(2k-1)+\lambda_{\mathrm{w}}) : S_{Rw} = S_{Rw} + m$$

$$S_{\mathrm{RP}} = 2S_{\mathrm{Rl}} + 2S_{\mathrm{Rw}} \tag{4.12}$$

$$\therefore S_{\mathrm{RP}} = 2m \left(\sum_{p=1}^{p=k} \left\lceil \frac{l - 2r(2p-1) + \lambda_{\mathrm{l}}}{2r} \right\rceil + \sum_{p=1}^{p=k} \left\lceil \frac{w - 2r(2p-1) + \lambda_{\mathrm{w}}}{2r} \right\rceil \right) \tag{4.13}$$
$$\textbf{for}\, l == (2r(2k-1)+\lambda_{\mathrm{l}}) : S_{\mathrm{RP}} = S_{\mathrm{RP}} + m$$
$$\textbf{for}\, w == (2r(2k-1)+\lambda_{\mathrm{w}}) : S_{\mathrm{RP}} = S_{\mathrm{RP}} + m$$

where S_{Rl} and S_{Rw} are the sensor densities along the length and width of the sensor perimeters respectively and S_{RP} is the sensor density in a rectangular-shaped space, based on a sensor perimeters deployment and p is the current perimeter.

The sensor density in a circular space, S_{CP}, is given as Eq. 4.14.

$$S_{\mathrm{CP}} = m \sum_{p=1}^{p=k} \left\lceil \frac{2\pi R - 2r(2p-1) + \lambda_{\mathrm{c}}}{2r} \right\rceil$$
$$\textbf{for}\, 2\pi R == (2r(2k-1) + \lambda_{\mathrm{c}}) : S_{\mathrm{CP}} = S_{\mathrm{CP}} + m \tag{4.14}$$

where S_{CP} is the sensor density when deploying a sensor perimeter in a circular shaped space, C is the circumference of the circular space, R is the radius of the circular space.

The sensor density for a right-angled polygon-shaped space is given as Eq. 4.15. Three examples of a perimeter deployment are illustrated in Fig. 4.4, where not all of the perimeters are complete closed perimeters like those created in the previous spaces. In this figure, different coloured squares denote the $(2r)^2$ grid squares, into which each node is placed. Each colour/shape is used to highlight a different perimeter. The numbers inside the squares denote the number of nodes placed along each edge of the perimeter, the numbers are not otherwise significant.

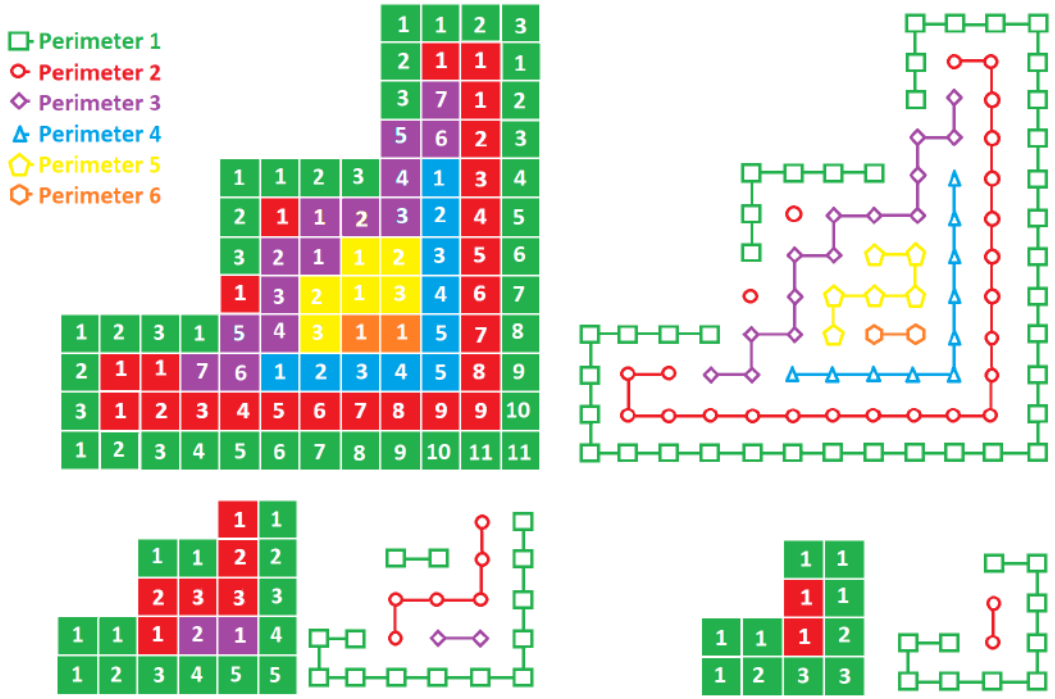


FIGURE 4.4: Pictorial view of sensor perimeter deployment in different polygon shaped spaces

$$S_{\mathrm{PP}} = m \sum_{p=1}^{p=k} \left(\sum_{x=1}^{x=X_{\mathrm{total}}} \left\lceil \frac{l_{\mathrm{x}} - 2r(2p-1) + \lambda_{\mathrm{x}}}{2r} \right\rceil \right) \tag{4.15}$$
$$\textbf{for } l_{\mathrm{x}} == (2r(2k-1) + \lambda) : S_{\mathrm{PP}} = S_{\mathrm{PP}} + m$$

where S_{PP} is the sensor density when deploying a sensor perimeter in a polygon-shaped space, x is the current edge, X_{total} is the total number edges in the space, p is the current perimeter, l_{x} is the length of edge x and λ_{x} is the constant based on length of x.

To determine sensor density for the sensor perimeters deployment in spaces that are right-angled polygons, Algorithms 1 and 3, are proposed.

Algorithm 1: Determine lambdaVector

Input: *lengthVector*, *r*
Result: *lambdaVector*, *vectorSize*
Initialisation: *loopVector*, *lambdaVector* = [ ]
Begin:
vectorSize = **Size**(*lengthVector*)
for *element=0* **to** *vectorSize -1* **do**
  element = *element* + 1
  l = *lengthVector*[*element*]
  $i = 1$
  while *loopVector*[*element*] = 0 **do**
    if $2ir \geq l > 2ir - r$ **then**
      lambdaVector[*element*] = 0
      loopVector[*element*] = 1
    end
    else if $2ir - r \geq l > 2ir - 2r$ **then**
      lambdaVector[*element*] = $2ir - l$
      loopVector[*element*] = 1
    end
    else
      $i = i + 1$
    end
  end
end
return *lambdaVector*, *vectorSize*

Algorithm 2: Determine the longest dimension in lengthVector

Input: $lengthVector$, $vectorSize$
Result: $lengthMax$
Initialisation: $lengthMax = 0$
Begin:
for $i = 1$ **to** $vectorSize - 1$ **do**
  if $lengthMax < lengthVector[i]$ **then**
    $lengthMax = lengthVector[i]$
  end
  $i = i + 1$
end
return $lengthMax$

Algorithm 3: Determine total number of nodes

Input: $lengthVector$, r, $lengthMax$, $vectorSize$, $lamdaVector$, m
Result: S_{PP}
Initialisation: $S_{PP} = S_x = 0$
Begin:
$k = \frac{lengthMax - 2r}{4r}$
for $p = 1$ **to** k **do**
  $S_p = 0$
  for $x = 0$ **to** $vectorSize - 1$ **do**
    $l_x = lengthVector[x]$
    $\lambda_x = lambdaVector[x]$
    $S_x = \left\lceil \frac{l_x - 2r(2p-1) + \lambda_x}{2r} \right\rceil$
    $S_p = S_p + S_x$
    if $l_x == 2r(2p-1) + \lambda_x$ **then**
      $S_p = S_p + 1$
    end
    $x = x + 1$
  end
  $S_{PP} = S_{PP} + S_p$
  $p = p + 1$
end
$S_{PP} = m\, S_{PP}$
return S_{PP}

4.4.3 Sensor Hexagons Node Density

The alternative hexagonal deployment is different from a sensor grid, or sensor perimeter. The sensor hexagons technique positions sensor nodes closer to each other, such that if the floor of a building is considered in terms of the xy axis, it can be seen that along the $y-$columns, the nodes are spaced as before, with $y-$axis node spacing, denoted $N_{\text{s-y}} = 2r$, but along the $x-$rows, the nodes are closer together. The node spacing along the $x-$axis, denoted $N_{\text{s-x}}$, is given as Eq. 4.16. Fig. 4.5 shows that the spacing between two nodes is equal to distance c and a is equal to the node coverage range r.

$$\begin{aligned} c^2 = a^2 + b^2 \ \therefore \ b = \sqrt{c^2 - a^2} \\ a = r; c = 2r; and \ b = N_{\text{s-x}} \\ \therefore N_{\text{s-x}} = \sqrt{(2r)^2 - r^2} = r\sqrt{3} \end{aligned} \tag{4.16}$$

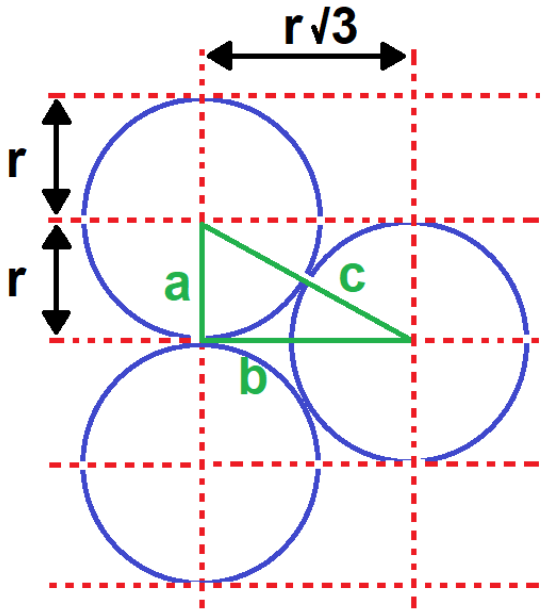


FIGURE 4.5: Spacing of nodes in sensor hexagons

Using different node spacing in the x and y dimensions, the sensor density can be calculated for the sensor hexagons deployment. This equation is derived from Eq. 4.12 and based on determining the number of nodes in each column, multiplied by the number nodes in each row, plus an additional node for every alternative row. The additional node is required because the nodes in the even rows are shifted down a distance of r, creating

a gap between the space's interior perimeter and the top node's coverage area. The sensor density of the sensor hexagon deployment in a rectangular-shaped space is given as Eq. 4.17. The constant λ_{wH} is defined in Eq. 4.18.

$$S_{\mathrm{RH}} = m\left(\left\lceil \frac{l+\lambda_{\mathrm{l}}}{2r} \right\rceil \left\lceil \frac{w+\lambda_{\mathrm{wH}}}{r\sqrt{3}} \right\rceil + \left\lceil \frac{w}{2r\sqrt{3}} \right\rceil\right) \tag{4.17}$$

$$\begin{aligned} &\textbf{if } ((2r + r\sqrt{3}(i_{\mathrm{wH}} - 1)) \geq w > (r + r\sqrt{3}(i_{\mathrm{wH}} - 1))) : \\ &\quad \lambda_{\mathrm{wH}} = 0 \\ &\textbf{else} \\ &\quad \lambda_{\mathrm{wH}} = r + r\sqrt{3}(i_{\mathrm{wH}} - 1) \end{aligned} \tag{4.18}$$

where λ_{wH} is a constant and i_{wH} is an incremental index, where $i_{wH} \in \mathbb{N}_1$.

4.5 Space Coverage

It is always useful to know what proportion of the building is being monitored. This can be used as a measure of the quality of service of a sensor network [99] and is referred to as 'Space Coverage'. Space coverage is based on the node coverage area, the number of nodes, denoted n and their position with respect to neighbouring nodes and the space's perimeter. The number of nodes and their positions are directly related to the deployment technique. As mentioned before, in this work, it has been assumed that the nodes are positioned on the ceiling and have a conical shaped coverage volume, which terminates on the floor in a circular coverage area. All of the nodes are assumed to have the same node coverage range, denoted r. Considering the non-overlapping deployments, the space coverage, denoted SC, is based on the total coverage area delivered by n divided by the total floor area, denoted A_{tot}.

$$SC = \frac{A_{\mathrm{C}} n}{A_{tot}} = \frac{n\pi r^2}{A_{tot}} \tag{4.19}$$

where A_C is the area of a circle with radius r, which is assumed to be equal to the coverage area of each node. The space coverage that can be achieved by different deployments is examined in Subsection 4.5.1 for the sensor grid and sensor hexagons deployments.

4.5.1 Sensor Grid Coverage

Based on a sensor grid deployment, the space coverage for a square space, SC_{SG}, is as Eq. 4.20 and for a rectangular space, SC_{RG}, in Eq. 4.21.

$$SC_{SG} = \frac{A_C n}{A_S} = \frac{n\pi r^2}{l^2} \tag{4.20}$$

$$SC_{RG} = \frac{A_C n}{A_R} = \frac{n\pi r^2}{lw} \tag{4.21}$$

In the following, an example deployment will be considered: If a square shaped space with a length $l = 2$ units, is monitored with a sensor grid deployment, where each node has a sensing range $r = l/4$ units, the number of nodes, S_{SG}, can be determined using Eq. 4.5 based on $m = 1$, $l = 2$, $r = \frac{l}{4}$ and $\lambda = 0$.

Eq. 4.5:

$$S_{SG} = m\left\lceil \frac{l+\lambda}{2r} \right\rceil^2 = 1\left\lceil \frac{4l+0}{2l} \right\rceil^2 = 4$$

Now, using the space coverage Eq. 4.20, variable n is replaced with sensor grid density, S_{SG}, as shown in Eq. 4.22.

$$\begin{aligned} SC_G &= \frac{n\pi r^2}{l^2} = \frac{4\pi(l/4)^2}{l^2} \\ SC_G &= \frac{4\pi l^2}{16l^2} = \frac{4\pi}{16} = \frac{\pi}{4} = 0.7854 = 78.54\% \end{aligned} \tag{4.22}$$

In this example and Eq. 4.22, space coverage is shown to be 78.54% of the space that is being monitored by the sensors, based on using the sensor grid or sensor perimeter deployment. The sensor hexagon technique should

theoretically achieve a higher rate of space coverage since it deploys more sensor nodes, positioned more closely together. This theory will be examined in Section 4.5.2.

4.5.2 Sensor Hexagons Coverage

In sensor hexagons deployment, sensors are packed more densely into the monitored space. Using the space coverage Eqs. 4.19 and 4.21, the sensor hexagons space coverage can be calculated. Here, it is assumed that the space is infinitely large, or the sensors have infinitely small node radii. Although these assumptions are not achievable, the conditions are effectively met if the hexagonal node coverage is allowed to overlap the edges of the space. These conditions mean that the monitored space is filled with hexagons, such that Eq. 4.23 is true.

$$SC_{\mathrm{H}} = \frac{A_C \; n \; Circles \; within \; Hexagon}{A_H} \tag{4.23}$$

A hexagon is comprised of 12 right angle triangles, as shown in Fig. 4.6. Using the $x-$axis node spacing, $N_{\text{s-x}}$ in Eq. 4.16, the area of a single triangle, denoted A_{T}, is given as Eq. 4.24, followed by the area of the hexagon, denoted A_{H}, in Eq. 4.25.

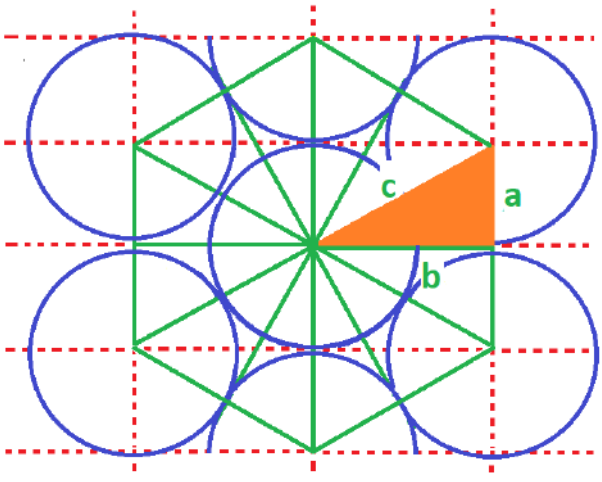


FIGURE 4.6: Spacing of nodes in sensor hexagons

$$A_{\mathrm{T}} = \frac{1}{2} B \; H = \frac{1}{2} ab = \frac{1}{2} ar\sqrt{3} = \frac{1}{2} r^2\sqrt{3} \tag{4.24}$$

$$A_{\mathrm{H}} = 12A_T = 12\frac{1}{2}r^2\sqrt{3} = 6r^2\sqrt{3} \tag{4.25}$$

Again referring to Fig. 4.6, the node coverage areas are represented by circles. It can be seen that in a hexagon, there is one full circle and one-third of six additional circles, i.e., the partial node coverage areas of the six neighbouring nodes. Eq. 4.26 determines the number of nodes, n, in a hexagon.

$$n = 1 + \frac{6}{3} = 3 \tag{4.26}$$

Eqs. 4.23 and 4.26, are applied to create Eq. 4.27, the sensor coverage, SC_{H}, that can be achieved by the sensor hexagons deployment.

$$SC_{\mathrm{H}} = \frac{3A_{\mathrm{circle}}}{6r^2\sqrt{3}} = \frac{3\pi r^2}{6r^2\sqrt{3}} = \frac{\pi}{2\sqrt{3}} = 90.69\% \tag{4.27}$$

The maximum space coverage that can be achieved with the sensor hexagons deployment is 90.69%, achieved based on using more sensors that the sensor grid deployment, which only achieves space coverage of 78.54%. The ratio of sensor density for each technique is explored in Section 4.5.3.

4.5.3 Node Density of Sensor Grid Versus Sensor Hexagons

Eqs. 4.28 to 4.31 are the ratio between the sensor density for the sensor grid deployment and the sensor hexagon deployment, based on Eq. 4.6 and 4.17.

Eq. 4.6:

$$S_{RG} = \; m\left\lceil\frac{l+\lambda_{\mathrm{l}}}{2r}\right\rceil\left\lceil\frac{w+\lambda_{\mathrm{w}}}{2r}\right\rceil$$

$$\therefore \left\lceil\frac{l+\lambda_{\mathrm{l}}}{2r}\right\rceil = \frac{S_{RG}}{m\left\lceil\frac{w+\lambda_{\mathrm{w}}}{2r}\right\rceil} \tag{4.28}$$

Eq. 4.17:

$$S_{\mathrm{RH}} = m\left(\left\lceil \frac{l+\lambda_{\mathrm{l}}}{2r} \right\rceil \left\lceil \frac{w+\lambda_{\mathrm{w}}}{r\sqrt{3}} \right\rceil + \left\lfloor \frac{w}{2.1r\sqrt{3}} \right\rfloor\right)$$

$$\therefore S_{\mathrm{RH}} = m\left(\left\lceil \frac{S_{RG}}{m\left\lceil \frac{w+\lambda_{\mathrm{w}}}{2r} \right\rceil} \right\rceil \left\lceil \frac{w+\lambda_{\mathrm{w}}}{r\sqrt{3}} \right\rceil + \left\lfloor \frac{w}{2r\sqrt{3}} \right\rfloor\right)$$

$$S_{\mathrm{RH}} = m\left(\frac{S_{RG}}{m}\left\lceil \frac{2r}{w+\lambda_{\mathrm{w}}} \right\rceil \left\lceil \frac{w+\lambda_{\mathrm{w}}}{r\sqrt{3}} \right\rceil + \left\lfloor \frac{w}{2r\sqrt{3}} \right\rfloor\right)$$

$$S_{\mathrm{RH}} = m\left(\frac{S_{\mathrm{RG}}}{m}\frac{2}{\sqrt{3}} + \left\lfloor \frac{w}{2r\sqrt{3}} \right\rfloor\right) \tag{4.29}$$

$$\therefore S_{\mathrm{RH}} = \frac{2S_{\mathrm{RG}}}{\sqrt{3}} + m\left\lfloor \frac{w}{2r\sqrt{3}} \right\rfloor \tag{4.30}$$

It should be noted, this ratio is now independent of the shape of the space, such that:

$$S_{\mathrm{H}} = \frac{2S_{\mathrm{G}}}{\sqrt{3}} + m\left\lfloor \frac{w}{2r\sqrt{3}} \right\rfloor \tag{4.31}$$

The relationship between sensor density in the sensor hexagons and sensor grid is approximately $\frac{2}{\sqrt{3}}$, or 115.47% that of the sensor grid. This means approximately 15.47% more sensors are required for the sensor hexagon deployment compared to the sensor grid or sensor perimeter deployment. In Section 4.6, these three deployment techniques will be developed further to include overlapping node coverage.

4.6 Overlapping Node Coverage Deployment Techniques

The sensor network deployment techniques investigated above did not allow any overlap of the node coverage areas between neighbouring nodes. This section will build upon the sensor grid, sensor perimeter and sensor hexagon deployment techniques to allow node coverage area overlap. The advantage of allowing overlap is that there will be less unmonitored area in the building. The disadvantage is the increase in sensor density, resulting in an increase in hardware cost and system complexity.

4.6.1 Overlapping Sensor Grid

For sensor grid and sensor perimeters deployments, the node spacing, $N_s = 2r$. This spacing will now be altered to allow node coverage overlap. The value of r, will remain the same, but the size of the grid squares will be altered, such that the grid square will sit completely inside the circular node coverage area, as illustrated in Fig. 4.7. The new node spacing, N_{so}, is given as Eq. 4.32 and the sensor density, S_{RGo}, in Eq. 4.33.

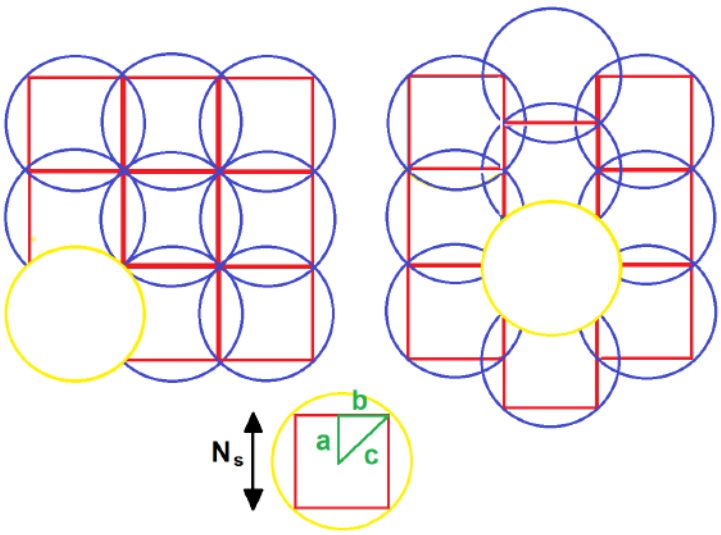


FIGURE 4.7: Spacing of nodes in overlapping sensor grid and sensor hexagons deployments

$$
\begin{aligned}
c^2 = a^2 + b^2 = 2a^2 \ \therefore \ a = \frac{c}{\sqrt{2}} \\
a = b;\ c = r;\ N_{\mathrm{so}} = 2a \\
\therefore N_{\mathrm{so}} = r\sqrt{2}
\end{aligned} \tag{4.32}
$$

$$
\begin{aligned}
S_{\mathrm{RGo}} = m\left(\left\lceil \frac{l+\lambda_{\mathrm{l}}}{N_{\mathrm{so}}} \right\rceil \left\lceil \frac{w+\lambda_{\mathrm{w}}}{N_{\mathrm{so}}} \right\rceil\right) \\
S_{\mathrm{RGo}} = m\left(\left\lceil \frac{l+\lambda_{\mathrm{l}}}{r\sqrt{2}} \right\rceil \left\lceil \frac{w+\lambda_{\mathrm{w}}}{r\sqrt{2}} \right\rceil\right)
\end{aligned} \tag{4.33}
$$

The ratio between the sensor density for a sensor grid and overlapping sensor grid is shown in Eq. 4.34. The maximum space coverage that can be achieved with all three of the overlapping deployment techniques is 100% because there are no coverage gaps between neighbouring nodes.

$$
\begin{aligned}
m\left(\left\lceil \frac{l+\lambda_{\mathrm{l}}}{r\sqrt{2}} \right\rceil \left\lceil \frac{w+\lambda_{\mathrm{w}}}{r\sqrt{2}} \right\rceil\right) \ : \ m\left(\left\lceil \frac{l+\lambda_{\mathrm{l}}}{2r} \right\rceil \left\lceil \frac{w+\lambda_{\mathrm{w}}}{2r} \right\rceil\right) \\
\therefore \ S_{\mathrm{Go}} = 2S_{\mathrm{G}}
\end{aligned} \tag{4.34}
$$

4.6.2 Overlapping Sensor Perimeters

Eq. 4.35 is the sensor density for the overlapping sensor perimeters, S_{PPo}.

$$
\begin{aligned}
S_{\mathrm{PPo}} = m\sum_{p=1}^{p=k}\left(\sum_{x=1}^{x=X_{\mathrm{total}}}\left\lceil \frac{l_{\mathrm{x}} - N_{\mathrm{so}}(2p-1)+\lambda_{\mathrm{x}}}{N_{\mathrm{so}}} \right\rceil\right) \\
\textbf{for}\ l_{\mathrm{x}} == (N_{\mathrm{so}}(2k-1)+\lambda) : S_{\mathrm{PPo}} = S_{\mathrm{PPo}} + m \\
S_{\mathrm{PPo}} = m\sum_{p=1}^{p=k}\left(\sum_{x=1}^{x=X_{\mathrm{total}}}\left\lceil \frac{l_{\mathrm{x}} - r\sqrt{2}(2p-1)+\lambda_{\mathrm{x}}}{r\sqrt{2}} \right\rceil\right) \\
\textbf{for}\ l_{\mathrm{x}} == (r\sqrt{2}(2k-1)+\lambda) : S_{\mathrm{PPo}} = S_{\mathrm{PPo}} + m
\end{aligned} \tag{4.35}
$$

The ratio between the sensor density of a sensor perimeter and an overlapping sensor perimeter is given as Eq. 4.36.

$$S_{\mathrm{Po}} \sim S_{\mathrm{P}}\sqrt{2} \tag{4.36}$$

4.6.3 Overlapping Sensor Hexagons

The relationship between the sensor density of sensor hexagons and overlapping sensor hexagons is not linear like that of the grid and perimeter techniques. This is because of the different $x-$axis and $y-$axis node spacing in the sensor hexagons deployment. For the sensor hexagons, the $y-$axis node spacing was $N_{\text{s-y}} = 2r$. The $x-$axis node spacing was $N_{\text{s-x}} = r\sqrt{3}$. For the overlapping sensor hexagons deployment, the $x-$axis and $y-$axis node spacing will be equal. To allow coverage overlap without any gaps, the node spacing should be between r and $r\sqrt{3}$ as shown in Eq. 4.37.

$$r\sqrt{3} \geq N_{\mathrm{sHo}} \geq r \tag{4.37}$$

To achieve the minimum sensor density, there should be minimal coverage overlap, so the node spacing is defined as $N_{\mathrm{sHo}} = r\sqrt{3}$. The sensor density, S_{RHo}, for the overlapping sensor hexagon deployment is given as Eq. 4.38.

$$\begin{aligned} S_{\mathrm{RHo}} &= m\left(\left\lceil\frac{l+\lambda_{\mathrm{l}}}{N_{\text{s-y}}}\right\rceil\left\lceil\frac{w+\lambda_{\mathrm{w}}}{N_{\text{s-x}}}\right\rceil + \left\lceil\frac{w}{2N_{\text{s-x}}}\right\rceil\right) \\ S_{\mathrm{RHo}} &= m\left(\left\lceil\frac{l+\lambda_{\mathrm{l}}}{r\sqrt{3}}\right\rceil\left\lceil\frac{w+\lambda_{\mathrm{w}}}{r\sqrt{3}}\right\rceil + \left\lceil\frac{w}{2r\sqrt{3}}\right\rceil\right) \end{aligned} \tag{4.38}$$

The ratio between the sensor hexagons and overlapping sensor hexagons deployments is shown in Eq. 4.39. Also, Eq. 4.40 is the ratio between the density of the overlapping sensor grid and the overlapping sensor hexagons deployment. The ratio is just over $\sqrt{\frac{2}{3}}$, or 81.65% (plus the additional nodes which are added to each even column of the overlapping sensor hexagons), meaning the overlapping sensor hexagon deployment requires approximately 81% of the sensors in the overlapping grid or overlapping

sensor perimeters deployments.

$$S_{\mathrm{RHo}} : S_{\mathrm{RH}}$$
$$m\left(\left\lceil \frac{l+\lambda_{\mathrm{l}}}{r\sqrt{3}} \right\rceil \left\lceil \frac{w+\lambda_{\mathrm{w}}}{r\sqrt{3}} \right\rceil + \left\lceil \frac{w}{2r\sqrt{3}} \right\rceil\right) \;:\; m\left(\left\lceil \frac{l+\lambda_{\mathrm{l}}}{2r} \right\rceil \left\lceil \frac{w+\lambda_{\mathrm{w}}}{r\sqrt{3}} \right\rceil + \left\lceil \frac{w}{2r\sqrt{3}} \right\rceil\right) \tag{4.39}$$

$$S_{\mathrm{RHo}} : S_{\mathrm{Go}}$$
$$m\left(\left\lceil \frac{l+\lambda_{\mathrm{l}}}{r\sqrt{3}} \right\rceil \left\lceil \frac{w+\lambda_{\mathrm{w}}}{r\sqrt{3}} \right\rceil\right) + m\left\lceil \frac{w}{2r\sqrt{3}} \right\rceil \;:\; m\left(\left\lceil \frac{l+\lambda_{\mathrm{l}}}{r\sqrt{2}} \right\rceil \left\lceil \frac{w+\lambda_{\mathrm{w}}}{r\sqrt{2}} \right\rceil\right)$$
$$\therefore\; S_{\mathrm{Ho}} = S_{\mathrm{Go}}\sqrt{\frac{2}{3}} + m\left\lceil \frac{w}{3r} \right\rceil \tag{4.40}$$

4.7 Evaluation of WSN Deployment Techniques

To evaluate the WSN deployment techniques, each of the networks are simulated using the MIoTs building simulation tool. MIoTs was developed for this work and is described in Chapter 3. MIoTs determines the position of nodes, generates building data and captures building data using for a number of sensors. Firstly, the simulated deployments are evaluated based on the sensor density, space coverage and sensor coverage overlap. Secondly, the deployments are evaluated regarding the ability to accurately capture multimodal building data and detect occupants, both compared to the simulated ground truth building data.

For this evaluation, two MIoTs variables are changed; *pack_scheme* and *overlapped*. All of the other variables are kept constant. The MIoTs variables

are detailed in Chapter 3, but for this evaluation they are defined as:

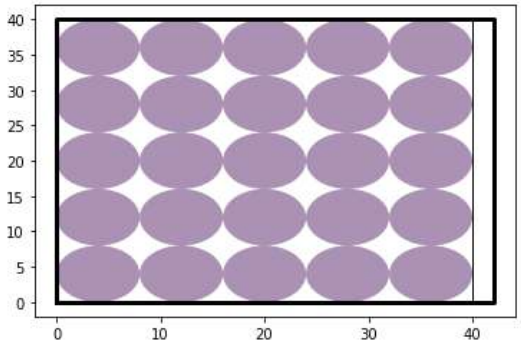
$$
\begin{aligned}
&initial\_n\_people = 0 \\
&n\_people\_min = 0 \\
&n\_people\_max = 200 \\
&moving\_in\_room\_prob = 0.1 \\
&change\_room\_prob = 0.08 \\
&in\_prob = 0.8 \\
&out\_prob = 0.8 \\
&in\_time = 8 \\
&out\_time = 17 \\
&n\_in\_at\_time = 20 \\
&n\_out\_at\_time = 20 \\
&random\_in\_out\_prob = 0.05 \\
&n\_measure\_per\_epoch = 30 \\
&n\_epoch = 24 \\
&sensor\_range = 4 \\
&r = Room([0,0,40,40],[0,40,40,0])
\end{aligned}
\tag{4.41}
$$

The different sensor networks are deployed in a large single room measuring $40m \times 40m$. A room of this size is selected. Each sensor node contains a single IRS, PIRS and CO_2S. The simulated sensors are all modelled with a conical sensor range $r = 4m$ which is based on a sensor with a FOV of 110° deployed at a height of $3m$, like PIRS model HC-SR501 [144]. Four different deployment techniques are compared including the sensor grid and overlapping sensor grid using variables $packing\_scheme = square$, $overlapped = False$ or $True$ respectively; or sensor hexagons and overlapping sensor hexagons using variable $packing\_scheme = hexagon$, $overlapped = False$ or $True$ respectively. The sensor perimeters and overlapping sensor perimeters are not evaluated since the deployment in a square-shaped space is identical to the sensor grid and overlapping sensor grid, respectively. For all MIoTs evaluations, the simulation is repeated ten times and average data are recorded.

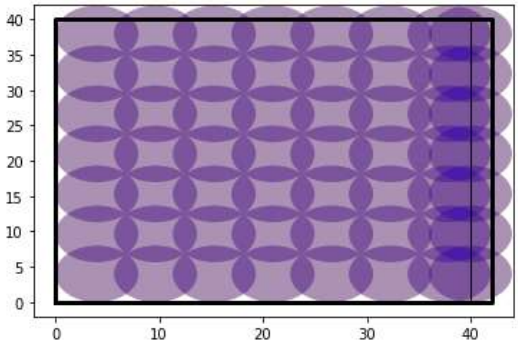
4.7.1 Evaluation 1: Node density, coverage and overlap

The aim of this first evaluation is to use MIoTs to compare the physical differences of the network based on deployment techniques. The characteristics that are compared include the simulated node density, node position, total sensor coverage and total amount of sensor coverage overlap.

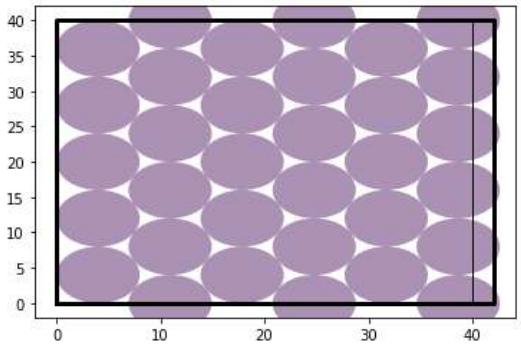
The four different network deployments are shown in Figs. 4.8a to 4.8d. The shaded areas show the combined coverage areas of each node. The areas of overlap are visible as slightly darker shaded regions. The sensor nodes are only effective within the room that they are deployed, hence the areas of node coverage that overlap the perimeter of the room are shown for illustration purposes only.



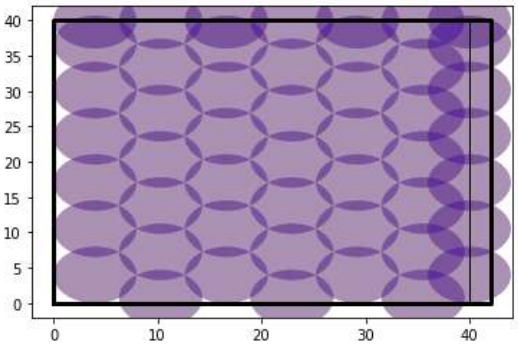
(A) Sensor grid deployment



(B) Overlapping sensor grid deployment



(C) Sensor hexagons deployment



(D) Overlapping sensor hexagons deployment

FIGURE 4.8: Horizontal view of 4 WSN deployments illustrating regions of node coverage

Regarding the cost of the deployment for this $1600m^2$ room; the sensor grid requires the lowest number of sensor nodes, 25. The sensor hexagons requires 33 nodes, that is 32% more than the sensor grid deployment. The sensor hexagons deployment requires 49 nodes, an increase of 96% and the overlapping sensor hexagons deployment requires 56 nodes, an increase of 124% on the sensor grid deployment. So, considering just the hardware

costs, the overlapping sensor grid costs more than double that of the sensor grid. The sensor grid achieves the lowest level of space coverage, leaving 21% of the room unmonitored, but it also has no sensor coverage overlap. This is followed by the sensor hexagon which achieves a higher level of coverage, monitoring 88% of the room, also with no sensor coverage overlap.

In comparison, the overlapping sensor grid and overlapping sensor hexagons deployments both achieve very high space coverage, leaving just 3.2% and 2.3% of the room uncovered respectively. To achieve this high coverage, both techniques also have a significant amount of sensor coverage overlap, resulting in some coverage redundancy. Redundancy may be necessary for some critical WSN deployments to ensure that no data are missed. But for the purpose and nature of building monitoring, coverage redundancy is not necessary.

Tables 4.1 summarises the physical features of the different network deployments including the node density, sensor density, space coverage and sensor overlap. This evaluation supports the findings of Section 4.4 to 4.6 and demonstrates that the sensor grid deployment requires the lowest number of nodes, followed by and the sensor hexagons deployment which required 32% more nodes. In comparison, the overlapping sensor hexagons deployment required 96% more nodes and the overlapping sensor grid 124% more. Based solely on hardware costs, the sensor grid or sensor hexagon deployment create the optimal network. The next evaluation, will study the impact of the network deployment technique on data collection.

TABLE 4.1: Comparison of sensor network features based on deployment techniques

	Sensor Grid	Overlapping Sensor Grid	Sensor Hexagons	Overlapping Sensor Hexagons
Node density	25	56	33	49
Sensor density	75	168	99	147
Space coverage (m^2)	1256.64 (78.54%)	1548.50 (96.78%)	1413.72 (88.36%)	1562.24 (97.64%)
Sensor overlap (m^2)	0	1139.52 (71.22%)	314.16 (19.64%)	900.77 (56.30%)

4.7.2 Evaluation 2: Data collection

The aim of this second evaluation is to determine if different network deployment techniques affect the accuracy of capturing building and occupancy data. This is determined by comparing

1. the simulated ground truth building against the simulated captured building data (PIRS and CO_2S data) and
2. the simulated ground truth occupant data against the simulated captured IRS data which is processed to determine occupancy.

4.7.2.1 Evaluation of PIRS data

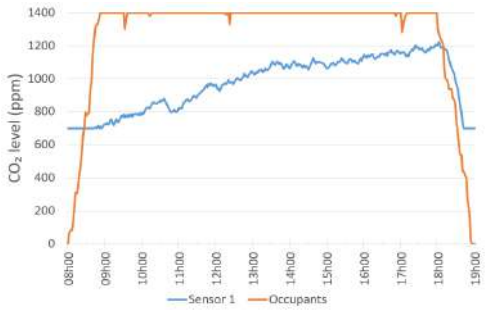
PIRS detect the movement of heat particles, which enables them to be used to detect occupants. As mentioned in Section 2.2, these sensors are not able to detect stationary occupants. PIRS generally demonstrate an accuracy around $60 - 70\%$ to indicate the presence of an occupant. In this evaluation, during working hours, simulated occupant levels increase from 0 to 200 occupants very quickly. The level of occupants remains quite constant until it drops back to 0 at the end of the day. This means that during working hours, within the coverage area of each node there are between $4 - 16$ occupants[1]. As such, based on this high level of occupancy there is a high level of occupant movement, resulting in the the PIRS constantly detecting this movement causing the PIRS to achieve 100% accuracy. This PIRS performance is demonstrated by all of the deployments. This high level of performance is more a reflection of the high density of occupants, rather than the performance of PIRS or the deployment techniques. It is expected that for a lower density of occupants, the performance of the PIR would be more closely aligned with the expected level of performance.

4.7.2.2 Evaluation of CO_2S data

For a large room with high maximum occupancy, a ventilation system is required to maintain occupant comfort levels. This is particularly true for CO_2, which would quickly rise to dangerous levels if an unventilated room was filled with 200 occupants. The behaviour of a mechanical ventilation system is modelled by the simulation tool. As such, the simulated CO_2 level is dependant upon the level of ventilation, as well as the number of

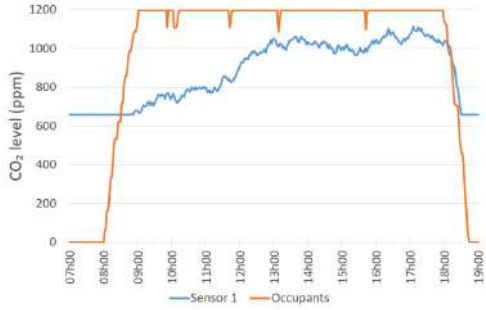
[1] Where 4 is calculated based on 200 occupants being equally spaced between 49 nodes. 16 is based on each occupant requiring a workspace of $4m^2$.

occupants in the room. In this evaluation, the level of ventilation and the number of occupants are identical for each simulation. The detected CO_2 data are shown in Figs. 4.9a to 4.9d. These figures illustrate that across all four deployments the CO_2 levels rise throughout the day until a maximum level is reached. This demonstrates that the level of $CO_2$is measured is not affected by the type of WSN deployment.

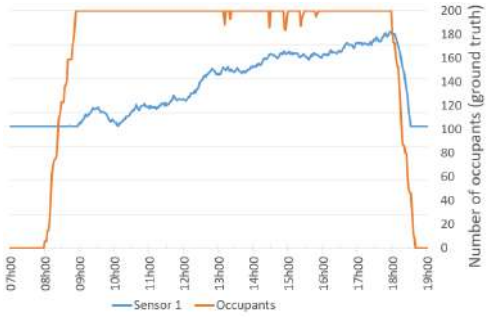


(A) Detected CO_2 level in sensor grid

(B) Detected CO_2 level in overlapping sensor grid



(C) Detected CO_2 level in sensor hexagons



(D) Detected CO_2 level in overlapping sensor hexagons

FIGURE 4.9: Graphs showing the detected CO_2 level and the number of occupants for each WSN deployment

4.7.2.3 Evaluation of IRS data

To determine the level of occupancy, the simulated thermal IRS data that is output by MIoTs is separately processed using the blob detection function which is part of the Python OpenCV image processing library. The blob detection process is utilised to count regions of heat within each IR image to determine the detected occupant number within the IRS's coverage area [64]. The accuracy of the detected occupancy number, α, is calculated based on x_{Ai}, which is the actual occupant number at each instance i, x_{Di}, the detected occupant number at each instance i and n the number of

instances:

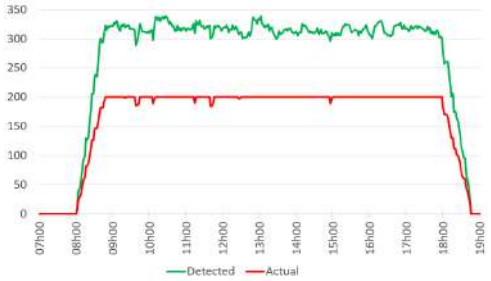
$$\alpha = 1 - \frac{1}{n} \sum_{i=1}^{i=n} \frac{|x_{\mathrm{Ai}} - x_{\mathrm{Di}}|}{x_{\mathrm{Ai}}} \tag{4.42}$$

The actual number of occupants and the detected number are shown in Figs. 4.10a to 4.10d. The sensor grid deployment achieves an accuracy of 81.2% when comparing the number of detected occupants against actual. This deployment offers the lowest level of space coverage at 78% which means that just over 21% of the room is unmonitored. As a result of these unmonitored areas, occupants in these areas are not being detected, causing under-counting. The hexagonal deployment achieves a higher accuracy of 91.7%, but it is still slightly under-counting the occupants. The improvement in occupant counting accuracy between the two deployments is due to this deployment achieving higher space coverage, over 88%.

The overlapping hexagon deployment detects approximately $227 - 275$ occupants, counting up to 37.5% more occupants than are present and achieving an accuracy of 70.3%. The overlapping grid deployment detects approximately $289 - 339$ occupants, over-counting by as much as 69.5%, achieving an accuracy of 40.9%. Generally, network redundancy improves the accuracy of data capture, but in this scenario the drop in the accuracy is a direct result of the sensor coverage overlap combined with the blob detection process. For the network deployments with large areas of sensor coverage overlap, the coverage overlap, combined with the blob detection process has led to heat regions being counted multiple times, which in turn has led to errors in determining the number of occupants. The sensor grid and sensor hexagons deployments do not have any areas of coverage overlap, but in comparison, the two overlapping deployments have a significant amount of coverage overlap. To enable accurate occupant numbers to be detected, for the sensor grid and sensor hexagons deployments, the sensor coverage gaps need to be reduced. For the overlapping deployment techniques, an alternative process to count occupants would need to be developed which prevents the double-counting of detected occupants.

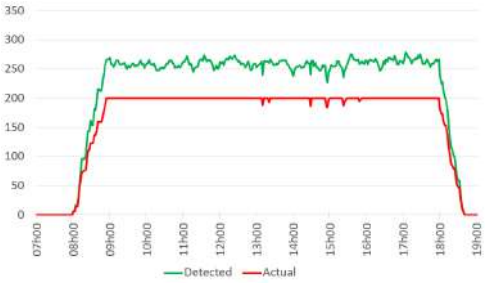
Table 4.2 summarises the performance data collection for the different sensor networks, demonstrating that the sensor grid and sensor hexagon deployment achieve the best overall performance.

(A) Detected occupants in sensor grid



(B) Detected occupants in overlapping sensor grid

(C) Detected occupants in sensor hexagons



(D) Detected occupants overlapping sensor hexagons

FIGURE 4.10: Graphs showing the actual number of occupants compared with the detected number for each WSN deployment

TABLE 4.2: Comparison of the performance of sensor data based on deployment techniques

	Sensor Grid (%)	Overlapping Sensor Grid (%)	Sensor Hexagons (%)	Overlapping Sensor Hexagons (%)
IRS (blob detection)	81.2	40.9	91.7	70.4
PIRS	100	100	100	100
CO_2	100	100	100	100

4.8 Chapter Summary

In this chapter, a number of sensor network deployment techniques were presented. Algorithms were proposed to determine the attributes of the different deployments including the sensor node density and space coverage. In addition, each WSN was simulated using the MIoTs simulation tool to allow further analysis of the different deployment techniques. The network performance was assessed by studying the simulated sensor data from PIRS, IRS and CO_2S and using this data to detect and count occupants. The deployments were evaluated based on the hardware cost and network

performance.

Based on the evaluation of the different WSN deployment techniques, the sensor grid and sensor hexagons deployments were found to be optimal both in terms of having the lowest hardware cost and for accurately monitoring occupants and building comfort levels. The hardware cost was similar for both deployments; for a room $40m \times 40m$ the sensor grid deployment required 25 nodes and the sensor hexagons deployment required 32% more nodes. These deployments achieved more than 78% and 88% space coverage respectively, with no coverage overlap between neighbouring nodes. Both deployments did create a small amount of redundant coverage overlap around the perimeter of the room. Utilising the blob detection image processing function, both deployments achieved over 81% and 91% occupancy accuracy respectively. In comparison, the two overlapping sensor deployments required significantly more hardware and when the blob detection function was applied to the sensor data simulated for these deployments, the processed data delivered a significant reduction in occupancy accuracy. This reduction in performance is counter intuitive due to increased in the amount of data that was captured, but was a result of the blob detection function double counting occupants that were detected in regions of coverage overlap between neighbouring nodes. To improve the accuracy of detected occupancy levels a different method of processing the IRS data would be required. The performance of the PIRS and CO_2S were not affected by the different deployment techniques.

Chapter 5

A Deep Neural Network for Edge-based Occupancy Detection

5.1 Introduction

In the previous chapter, a number of WSN deployment techniques were evaluated. They demonstrated some errors when monitoring occupancy levels. This chapter will investigate if ML techniques can be used to improve the accuracy of occupancy detection. The system that has been developed is the **De**ep **N**eural **N**etwork **O**ccupancy detection **T**hermal **E**dge-based system, referred to as DeNNOTE. It has been developed with SMEs operating in small commercial buildings in mind, to enable a reduction in energy consumption within their buildings. The DeNNOTE system will enable an increased temporal match between occupant presence and energy-consuming systems, including heating and lighting [64]. To make the system accessible to SMEs, the system costs will be kept low. Also, SMEs may not have access to a powerful workstation or server such as the type of machine which are typically used to train and test a ML model. To mirror these limitations the complete DeNNOTE life cycle will be carried out on an edge device. This life cycle includes train and test data collection, data preprocessing, ML model training, model testing, model optimisation, DeNNOTE system testing and finally DeNNOTE system deployment. The DeNNOTE system will determine whether occupants are present or not, rather than counting occupants which complements the most common type of heating systems found in SME buildings; a boiler and radiators. This determination of occupants being present or absent will keep system complexity to a minimum, further reducing system costs.

The DeNNOTE system is an IoT sensor-based system which utilises ML to

determine occupancy using IR thermal data. The thermal data will be captured using an IRS. IRS offer a number of significant advantages compared to other sensors. Firstly, IRS do not require user agreement or participation, unlike communication or phone based systems. Unlike other image-based systems, IRS create IR thermal images not photographic images and so do not introduce privacy issues. IRS can detect occupants in real-time, in comparison to gas sensors which create a time lag. Finally, IRS can detect occupants irrespective of whether they are stationary or moving, in comparison to commonly used PIRS, which can only detect moving occupants. As part of this research, a number of different ML techniques were trialled on the edge device. It was determined that the most suitable was a feed-forward DNN due to the vast amount of development support available through edge-based Open Source software, the flexibility of DNNs such that a model can be developed which performs well with thermal data that has received minimal preprocessing and the edge device being able to meet the processing requirements to define, compile, train, validate, test and deploy a DNN. The software includes the TensorFlow Open Source AI tool installed on the Raspberry Pi edge device. Python scripts executed on the Raspberry Pi are used for data capture and once the system is deployed, to enable control of building heating and lighting.

The DNN model is tested in multiple internal building environments, with varying room layouts and occupancy patterns. Across these environments the model demonstrates a high generalisation ability[1], creating confidence that it will also function optimally in new building environments.

This chapter makes a number of contributions which includes the DeNNOTE occupancy detection system and the DNN model architecture to solve the problem of occupancy detection within a building. The full data set comprising of building monitoring and occupancy data from three environments which will be published. The methodology to determine the most suitable data set for DNN model training, validation and testing, as well as the optimal DNN model hyperparameters to solve the given problem.

The layout of the remainder of this chapter includes the proposed DNN architecture, which is defined in Section 5.2. The process of occupancy

[1]Generalisation ability refers to a model's ability to accurately interpret new and unseen data. From this point onward, it will refer specifically to this model's ability to interpret test data that was captured in a new building environment relative to the training environment.

classification is explained in Section 5.3 including the hardware, software, data capture, data preprocessing and optimisation of the DNN. The evaluation of the system is carried out in Section 5.4. In Section 5.5, the DeNNOTE system is deployed using the WSN positioning techniques introduced in Chapter 4 and the performance of the deployments are evaluated. Section 5.6 is the Chapter Summary.

5.2 DeNNOTE Architecture

The concept of occupancy detection is introduced in Section 2.2. Here, it is formulated as a binary classification problem and a DNN architecture is used to learn the discriminating features of IR image data, $\mathbf{X} \in \mathbb{R}^{a \times c \times d \times e}$, in a supervised manner using the corresponding ground-truth labels, $\mathbf{y} \in \mathbb{R}^{a}$. Throughout this work, boldface uppercase alphabets and boldface lower case alphabets represent matrices and column vectors respectively; a batch size of IR image data; c image width; d image height; and e image channels.

The proposed DNN architecture presented in Algorithm 4 is comprised of a flatten layer, two dense hidden layers and a dense output layer. The flatten layer transforms the dimension of a sample of $\mathbf{X}$ from $1 \times c \times d \times e$ into $1 \times g$ as stated in Eq. 5.1. Different numbers of hidden layers were explored, but it was determined that two hidden layers was optimal in terms of performance and processing requirements.

$$\mathbf{f} = \beta\left(\mathbf{X}\right), \tag{5.1}$$

where $g = c \times d \times e$; and $\beta(\cdot)$ is the flatten function. The first dense layer transforms the flattened batch of IR image data using a weight matrix, $\mathbf{W}_1 \in \mathbb{R}^{g \times g}$, a bias vector, $\mathbf{b}_1 \in \mathbb{R}^{1 \times g}$ and an activation function, $\sigma_1\left(\cdot\right)$, as stated in Eq. 5.2.

$$\mathbf{h}_1 = \sigma_1\left(\mathbf{f} \cdot \mathbf{W}_1 + \mathbf{b}_1\right), \tag{5.2}$$

The second dense layer transforms the output of the first hidden layer using a weight matrix, $\mathbf{W}_2 \in \mathbb{R}^{g \times g}$, a bias vector, $\mathbf{b}_2 \in \mathbb{R}^{1 \times g}$ and an activation function, $\sigma_2(\cdot)$, as stated in Eq. 5.3.

$$\mathbf{h}_2 = \sigma_2\left(\mathbf{h}_1 \cdot \mathbf{W}_2 + \mathbf{b}_2\right), \tag{5.3}$$

The output layer transforms the output of the second hidden layer using an activation function, $\phi\ (\cdot)$, as stated in Eq. 5.4.

$$\tilde{\mathbf{y}} = \phi\ (\mathbf{h}_2), \tag{5.4}$$

where $\tilde{\mathbf{y}}$ is the labels predicted by the DNN model. The difference between $\tilde{\mathbf{y}}$ and $\mathbf{y}$ is given by Eq. 5.5.

$$L = \sum_{k=1}^{n} \theta(\mathbf{y}_k, \tilde{\mathbf{y}}_k), \tag{5.5}$$

where θ is a sparse categorical cross-entropy loss function; and n is the number of samples in a batch of IR image data. In order to minimise loss, the trainable parameters of the DNN model are adjusted over u epochs based on the current loss score, L, using an efficient first-order stochastic gradient descent function named Adam, ψ, [145] as stated in Eq. 5.6.

$$\mathbf{W}^{'}_{(\cdot)}, \mathbf{b}^{'}_{(\cdot)} = \psi\left(L, \mathbf{W}_{(\cdot)}, \mathbf{b}_{(\cdot)}\right), \tag{5.6}$$

where $\mathbf{W}_{(\cdot)}$ are the previous weight matrices; $\mathbf{W}^{'}_{(\cdot)}$ are the new weight matrices; $\mathbf{b}_{(\cdot)}$ are the previous bias vectors; and $\mathbf{b}^{'}_{(\cdot)}$ are the new bias vectors.

Algorithm 4: DNN Algorithm

$\mathbf{f} = \beta(\mathbf{X})$
for $i = 1$ **to** u **do**
  for $k = 1$ **to** n **do**
    $\mathbf{h}_{1i,k} = \sigma_1\ (\mathbf{f}_k \cdot \mathbf{W}_{1i} + \mathbf{b}_{1i})$
    $\mathbf{h}_{2i,k} = \sigma_2\ (\mathbf{h}_{1i,k} \cdot \mathbf{W}_{2i} + \mathbf{b}_{2i})$
    $\tilde{\mathbf{y}}_{i,k} = \phi\ (\mathbf{h}_{2i,k})$
    $L_{i,k} = \theta(\mathbf{y}_{i,k}, \tilde{\mathbf{y}}_{i,k})$
  end
  $L_i = \sum_{k=1}^{n} L_{i,k}\ \mathbf{W}^{'}_{i(\cdot)}, \mathbf{b}^{'}_{i(\cdot)} = \psi\left(L_i, \mathbf{W}_{i(\cdot)}, \mathbf{b}_{(i\cdot)}\right)$
end
$L = \sum_{i=1}^{u} L_i$

5.3 Process of Occupancy Classification

In this Section, we develop the process of occupancy classification which includes IR image capturing, data preprocessing and optimisation of the

DNN model. The system is implemented on a Raspberry Pi 3B+ and its output signal is sent to the heating and lighting within the building. The flowchart of the DeNNOTE system process is shown in Fig. 5.2.

5.3.1 System Hardware and Software

The DeNNOTE system includes four types of sensors and a processor. The thermal data collected by the IRS will be used to train the DNN, the other multimode data is not used by the initial DeNNOTE system, but in the later versions which are detailed in Chapter 6. The sensors include the MLX90640 24×32 pixel IR thermal sensor [146], Raspberry Pi 5 MP $1080p$ camera module [147], K30 1% $10,000$ ppm CO_2 sensor (CO_2S) [148] and HC-SR501 PIR motion detection sensor [144]. The processor is a Raspberry Pi 3B+ [149] with $128GB$ SD memory card and $1GB$ RAM. The system hardware is shown in Fig. 5.1 and the architecture is shown within the **IR Image Capturing Module** in Fig. 5.2.

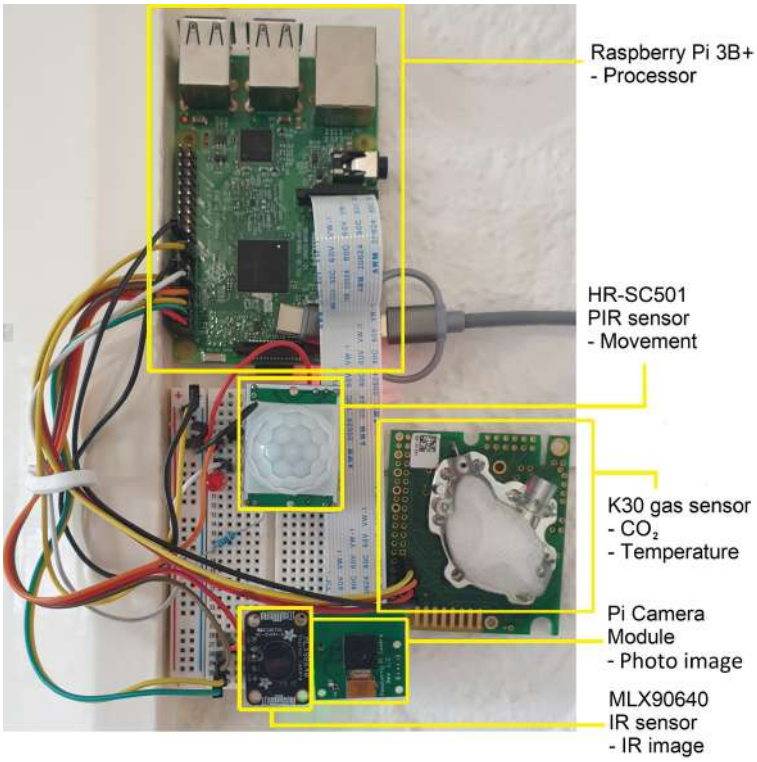


FIGURE 5.1: System hardware

The MLX90640 has been selected due to its low cost and specifications which include a range of $11m$ and a wide field of view of $110 \times 70°$. Based on being positioned at a height of $3m$ it creates a conical coverage area of $\sim 36m^2$. Within the DeNNOTE system, a single MLX90640 sensor can be

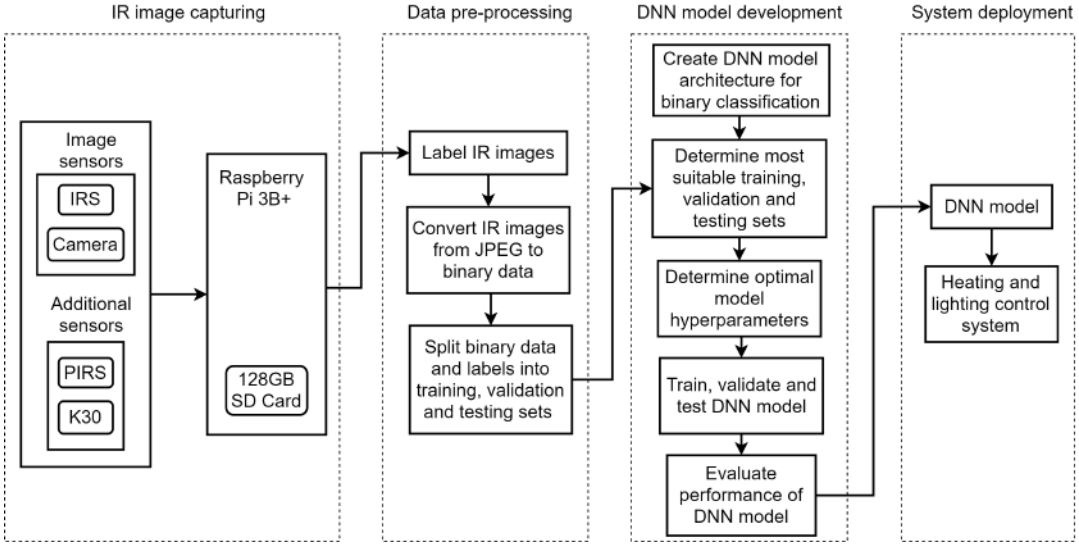


FIGURE 5.2: Overview of the DeNNOTE occupancy detection process

used or multiple MLX90640 sensors can be combined to create a larger coverage area. Throughout this chapter, a single sensor is used. During the data capture phase, simultaneously with the IRS, the Raspberry Pi camera module captures a photographic image which is used to create the ground truth label for the corresponding IR image. The photographic images are only used for data labelling and will not be part of the published data set.

5.3.2 Data Capture

5.3.2.1 Monitored environment

This system was developed to be deployed within an SME in Greater Manchester, capturing data in three similar environments; the office, meeting room and kitchen. The environments are real building environments, not highly regulated like that in [49]. As such, environment variables including room temperature, light level, ventilation and occupant levels vary. Indoor temperature is increased using a single radiator in each environment or decreased by opening windows. The occupant levels in each of the environments vary up to a maximum of six occupants.

5.3.2.2 Data capture method

A Python script run on the Raspberry Pi captures thermal IR and photographic data at 60 second intervals. The IR image and the photographic image are each saved as a timestamped JPEG file. When

studying the state-of-the-art [28], [40], [49], [50], [65], it was noted that there is a significant amount of building monitoring data which includes monitored attributes such as motion, temperature, humidity, light and CO_2, but there is a shortage of published building data sets which include thermal data. As such, the data that is collected for this research will be published.

5.3.2.3 Data set creation

Using data captured within the office, five train/test data sets are created and labelled `Data.i1` to `Data.i5`. The data sets are differentiated based on when the data was captured and data set size. Two additional test data sets are created from data captured in two additional environments, the kitchen and meeting room. Validation data, $\mathbf{X}_{\text{val}}$, is created by extracting 20% of data from the train data sets. The data sets are evaluated by comparing the performance of the DNN, trained and tested with them. For the evaluation of the data sets the DNN will be defined with activation functions σ_1 and σ_2 set to Relu in both hidden dense layers and the model will be trained with 5 epochs.

`Data.i1` to `Data.i3` all comprise 1440 train samples and 717 test samples. This data are captured over a 36 hour period. `Data.i1` is captured $12:00\ 23/04 - 23:59\ 24/04$ and separated into train and test data based on the image capture time:

- test data: $12:00 - 23:59$ on $23/04$. The outdoor temperature varied between a high of 18 °C to a low of 6 °C.
- train data: $00:00 - 23:59$ on $24/04$. The outdoor temperature varied from a high of 21 °C to a low of 9 °C.

`Data.i2` is captured between $24/04 - 29/04$. This data set differs from `Data.i1` because the train and test data are captured a few days apart when the outdoor temperature varied significantly:

- test data: $12:00 - 23:59$ on $29/04$. The outdoor temperature ranged from a high of 11 °C to a low of 4 °C
- train data: $00:00 - 23:59$ on $24/04$. The outdoor temperature varied from a high of 21 °C to a low of 9 °C.

`Data.i3` is comprised of the data previously captured for `Data.i1`, randomly separated into a train and test data set. `Data.i4` and `Data.i5` are large data sets and comprise 9300 train samples and 2675 test samples.

`Data.i4` is captured between 23/4-14/05 and separated into train and test sets based on capture time:

- test data: $12:00\ 11/05 - 23:59$ on $14/05$. The outdoor temperature ranged between highs of $13 - 10$ °C and lows of $5 - 3$ °C.
- train data: $12:00\ 23/04 - 23:59$ on $03/05$. The outdoor temperature ranged between highs of $21 - 10$ °C and lows of $9 - 4$ °C.

`Data.i5` is captured $12:00\ 23/04 - 23:59$ on $22/05$ and separated into a randomly mixed train data set, $\mathbf{X}_{\text{train.i5}}$ and test data set, $\mathbf{X}_{\text{otest}}$. The model trained with this data set is referred to as `Model.i5`. Additional test data sets, $\mathbf{X}_{\text{mtest}}$ and $\mathbf{X}_{\text{ktest}}$, are captured between $03/06 - 05/06$ in the meeting room and $26/05 - 30/05$ in the kitchen.

5.3.3 Data Preprocessing

The DNN is developed so that it can classify IR data which has received minimal preprocessing. There are two preprocessing steps applied to each IR image as shown within the **Data Preprocessing Module** in Fig. 5.2. The first step utilises the corresponding photographic image to determine its ground truth label. Next, the labelled IR image is converted from a JPEG image to a binary data array. The sensor deployments are shown in Chapter 4 in Figs. 4.8a to 4.8d. Based on a room $40m \times 40m$ and a sensor coverage range $r = 4$, the number of sensor nodes were calculated as 25, 33, 49 and 56 for the sensor grid, sensor hexagons, overlapping sensor hexagons and overlapping sensor grid deployments respectively. Additional hardware is

5.3.4 Optimisation of DNN Model

Research studying the performance of DNNs across a range of fields shows that altering the model's hyperparameters can have a significant effect on the model's performance [150]–[152], though the existing systems [28], [49], [50], [65] do not demonstrate optimising their AI classifier using this process. Specific hyperparameters are varied, including the number of dense hidden layers, denote h, number of neurons per dense hidden layer, denoted η, loss function, denoted θ, dense hidden layer activation functions, denoted σ and number of training epochs, denoted u. The activation functions within each of the dense hidden layers can differ from each other, they are selected from the following list: Elu, Relu, Selu, Sigmoid, Softplus, Softsign and Tanh. The

number of epochs used within model training is varied from $2-10$. For all model evaluation within this work, the train and test steps are repeated 10 times and the mean results are reported.

Based on the evaluation of data sets `Data.i1` to `Data.i5`; `Data.i5` is used to determine the generalisation ability of the proposed DNN. The model is first trained and tested with data set `Data.i5` which is comprised of train data denoted $\mathbf{X}_{\text{train.i5}}$ and test data denoted $\mathbf{X}_{\text{otest}}$. Next, the model's test performance is compared against its performance with two new test sets, $\mathbf{X}_{\text{ktest}}$ and $\mathbf{X}_{\text{mtest}}$, collected in the kitchen and meeting room respectively.

5.4 Performance Evaluation of DNN Model

We evaluate the performance of DNN model based on Accuracy, Balanced Accuracy (BAcc), Sensitivity, Precision and F1, which are defined in Eq. 5.7 to 5.12. Accuracy is a measure of the proportion of true results, but it can be misleading if there is an imbalance in the size of each data class [153]. BAcc is a normalised measure of the proportion of the true results; less prone to issues caused by class imbalance. Precision is a measure of how many true results are correctly reported. Sensitivity is a measure of the proportion of true results that are reported. F1 is a combination of sensitivity and precision. Specificity is a measure of the proportion of negative results correctly reported.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.7}$$

$$Precision = \frac{TP}{TP + FP} \tag{5.8}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{5.9}$$

$$F1 = \frac{2TP}{2TP + TP + FP} \tag{5.10}$$

$$Specificity = \frac{TN}{TN + FP} \tag{5.11}$$

$$BAcc = \frac{TP}{2(TP+FN)} + \frac{TN}{2(TN+FP)} \tag{5.12}$$

$$BAcc = \frac{Sensitivity + Specificity}{2}$$

where for this binary classification task, ***True*** indicates ***occupant(s) present*** and ***False*** indicates ***occupant(s) absent***. True Positive (TP) indicates the frequency with which the model correctly classifies occupants as present, False Positive (FP) indicates the frequency that the model wrongly classifies occupants as present. Similarly, True Negative (TN) and False Negative (FN) indicate the frequency that the model correctly and incorrectly classifies occupants as absent. A FN is a more significant issue than a FP because it will result in the DeNNOTE system switching off heating and lighting when they are still required since there are occupants present. This incorrect response could make the building environment uncomfortable, or at worst, unusable for occupants. In comparison, a FP result is a less significant issue because it will not create an unusable environment, though it will cause the system to keep the heating and lighting on when there are no occupants which is counter to the aim of the system. As such, a FP result and will impact on the system's ability to meet its aim to reduce energy consumption. Improvements in sensitivity, precision and F1 demonstrate an improvement in the correct detection of occupants. In comparison improvements in specificity demonstrate an improvement in the correct detection of the absence of occupants.

5.4.1 Evaluation of Data Sets

The aim of this evaluation is to determine which train and test data set, `Data.i1` to `Data.i5`, enable the DNN to achieve the best performance. For this evaluation the DNN will be defined with activation functions σ_1 and σ_2 set to Relu in both hidden dense layers and the model will be trained with 5 epochs.

Fig. 5.3 illustrates the test performance of the DNN trained and tested with `Data.i1` to `Data.i3` and `Data.i5`. It shows that all of these data sets achieve BAcc above 96%. The model trained and tested with `Data.i5` closely followed by `Data.i3` achieved the highest test performance, showing that

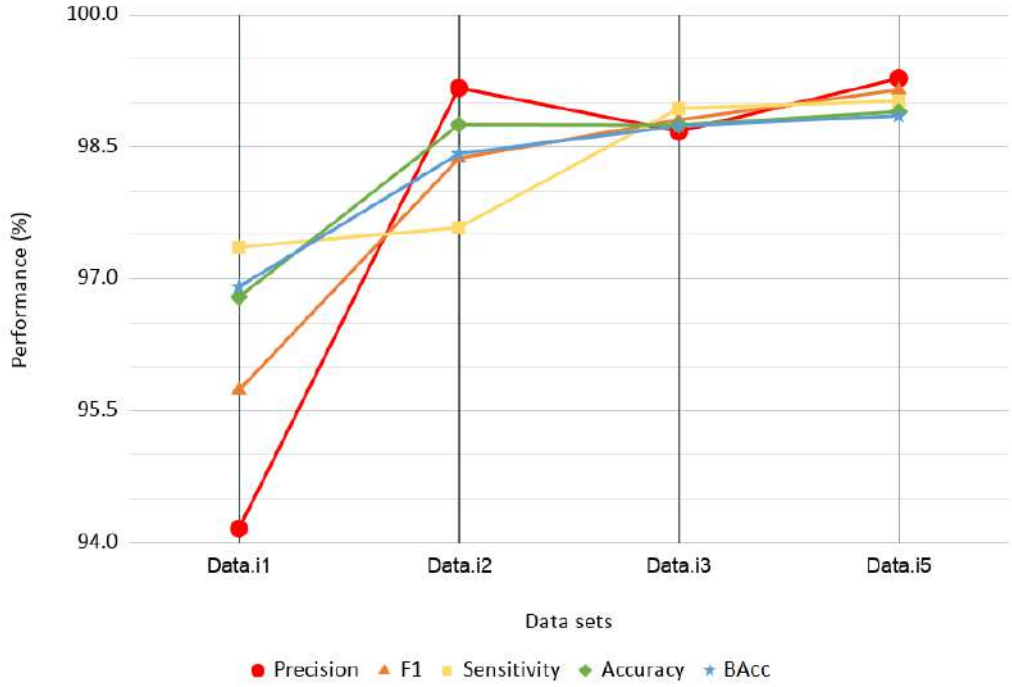


FIGURE 5.3: Comparison of model performance achieved with Data.i1, i2, i3 and Data.i5

the data sets which enable the best performance are the most diverse data sets, created by mixing the dates of the train and test data. The model trained with data set `Data.i5` achieves better performance than `Data.i3` since it is a larger data set, comprising more samples, providing the DNN with the most data for it to learn from, resulting in a model with the highest performance. The model trained and tested with `Data.i4` achieved the lowest BAcc of 93%. This lower performance occurred because the train and test data were collected during periods with different external temperature ranges, this temperature variation caused the occupants to change their use of the heating system, which in turn affected the discriminating features in the train data in comparison to the features in the test data, which detrimentally affected the model's performance.

5.4.2 Evaluation of DNN's Generalisation Ability

The aim of this evaluation is to determine the generalisation ability[2] of the DNN model when it is tested with new and unseen data collected in a

[2]Generalisation ability refers to a model's ability to accurately interpret new and unseen data. From this point onward, it will refer specifically to this model's ability to interpret test data that was captured in a new building environment relative to the training environment.

number of different environments, the office, meeting room and kitchen. Unseen data refers to a data set which was not used during the training of the DNN model and is therefore unseen by the DNN. For this evaluation the DNN model `Model.i5` will be defined with activation functions σ_1 and σ_2 set to Relu in both hidden dense layers, trained over 5 epochs and with data set `Data.i5`.

Table 5.1 shows the performance of `Model.i5` when comparing the test data, $\mathbf{X}_{\text{otest}}$, $\mathbf{X}_{\text{mtest}}$ and $\mathbf{X}_{\text{ktest}}$ captured in the office, meeting room and kitchen respectively. It can be seen that the test accuracy is high for all three environments (99.63% to 81.57%). Since there is data classification imbalance, BAcc is a better metric than accuracy to give a more complete picture of the model's performance [153]. For the office, the BAcc performance is above 99%, it is also high in the meeting room, above 87%, but a little lower in the kitchen at 74%. So, the DNN model demonstrates a relatively high generalisation ability across these different environments. The reduction in performance for both the meeting room and kitchen data is because the model returned a higher rate of FN in comparison to the office, demonstrating a lower sensitivity.

TABLE 5.1: Generalisation ability of Model.i5

Environ	Precision (%)	F1 (%)	Sensitivity (%)	Accuracy (%)	BAcc (%)
office	99.19	99.95	99.95	99.63	99.70
Meeting	98.33	90.21	90.21	89.72	87.68
kitchen	95.99	83.22	83.22	81.57	74.09

The slightly lower performance in the kitchen can be explained as a result of this environment being quite different to the office environment where the model was trained. The difference is particularly with respect to the background temperature which is higher due to the regular use of kitchen equipment, meaning there is a smaller difference between background and body temperature, which makes it more difficult for the DNN to detect occupants. Such results suggest that determining occupancy using a DNN trained with thermal data works well in environments where the background temperature is significantly lower than body temperature, for example 18°C to 23°C, but it may not be well suited for warmer internal environments such as server rooms or commercial kitchens.

5.4.3 Evaluation of DNN's Hyperparameters

The aim of this evaluation is to determine the hyperparameters which enable the DNN to achieve the optimal performance in terms of high generalisation ability demonstrated across different environments. For this evaluation the model is trained with data set `Data.i5`. The initial hyperparameters will be activation functions σ_1 and σ_2 set to Relu and $u = 5$ training epochs. The results of varying the activation functions and number of training epochs are shown below.

When the model is tested with office data, varying the hyperparameters does not have a significant effect on its performance. The model achieves a maximum BAcc of over 99% when both of the activation functions σ_1 and σ_2 are set to Sigmoid and the model has been trained over 10 epochs. The maximum variation in the performance of the model with this data is less than 1%, which is achieved with the same activation functions and the model is trained over 5 epochs. In comparison, a large variation of 17% is seen when the model is tested with data captured in the meeting room and the model is defined with different hyperparameters. The maximum performance is achieved when the model has been trained over 2 epochs and the activation functions σ_1 is set to Sigmoid, σ_2 is set to Relu. In these tests, the model achieved BAcc above 93%. The lowest performance, when tested with the same meeting room data, is achieved when both activation functions σ_1 and σ_2 are changed to Sigmoid and the model has been trained over 10 epochs, achieving 76%. The variation in performance achieved by the model when tested with data captured in the kitchen is similar, at nearly 16%. The maximum performance, BAcc slightly above 80%, is achieved when the model is defined with both activation functions σ_1 and σ_2 are set to Sigmoid and trained with 2 epochs. The lowest performance is seen when both activation functions σ_1 and σ_2 are set to Tanh and the model has been trained with 10 epochs.

Table 5.2 to 5.4 compare the model's performance based on varying the number of training epochs and testing the model with data captured in the office, meeting room and kitchen respectively. Throughout these tests, the Sigmoid activation function is used in both dense hidden layers. Varying the number of training epochs has minimal effect on the model's performance when it is tested with office data. In comparison, when the model is tested with meeting room data, it performs best when it is trained with 2 epochs, closely followed by 5 epochs. When the model is tested with

kitchen data, it performs best when trained with 2 epochs. These results clearly demonstrate that a higher number of epochs causes this model to overfit to the training data which was captured in the office and reduces the model's performance when it is tested with data captured in the other similar environments. So, the optimal number of training epochs to enable this model to achieve the highest generalisation ability is two epochs.

TABLE 5.2: Test performance of DNN based on varied epochs, Sigmoid and Sigmoid activation functions and office test data

Epochs	Precision (%)	F1 (%)	Sensitivity (%)	Accuracy (%)	BAcc (%)
2	99.32	98.92	98.51	98.60	98.63
5	98.16	98.93	99.71	98.63	98.24
8	99.37	99.06	98.75	98.79	98.80
10	99.32	99.32	99.32	99.13	99.05

TABLE 5.3: Test performance of DNN based on varied epochs, Sigmoid and Sigmoid activation functions and meeting room test data

Epochs	Precision (%)	F1 (%)	Sensitivity (%)	Accuracy (%)	BAcc (%)
2	99.36	94.85	90.74	91.04	92.42
5	99.09	94.67	90.63	90.72	91.19
8	98.66	94.77	91.18	90.95	90.05
10	98.61	94.46	90.64	90.39	89.32

TABLE 5.4: Test performance of DNN based on varied epochs, Sigmoid and Sigmoid activation functions and kitchen test data

Epochs	Precision (%)	F1 (%)	Sensitivity (%)	Accuracy (%)	BAcc (%)
2	99.32	88.97	80.57	80.57	80.56
5	95.36	89.37	84.09	82.11	74.71
8	94.23	90.35	86.77	84.12	77.52
10	96.29	88.76	82.32	80.76	72.26

Figs. 5.4 to 5.6 compare the model's performance based on varying the activation functions and testing it with data captured in each of the environments. Throughout these tests, two training epochs are used. When the model is tested with data captured in the office, varying the activation

functions in the hidden layers has a minimal effect on the model's performance, resulting in a maximum variation of less than 0.5%. The optimal performance was achieved with the Elu activation function in both hidden layers. When the model was tested with data captured in both the meeting room and the kitchen, the model achieved optimal performance with the Sigmoid activation function in both hidden layers.

To summarise, the DNN model demonstrates the best overall performance, also achieving highest generalisation ability when it is defined with hyperparameters Sigmoid activation function in both dense hidden layers and trained with 2 epochs. This configuration achieves BAcc of 98.63% with office data, 92.42% with meeting room data and 80.56% with kitchen data. This DNN model will be referred to as DNN-S2e based on these hyperparameters. These results are summarised in Table 5.5.

TABLE 5.5: Test performance of DNN demonstrating highest generalisation ability (Sigmoid activation function and 2 training epoch)

Environ	Precision (%)	F1 (%)	Sensitivity (%)	Accuracy (%)	BAcc (%)
office	99.32	98.92	98.51	98.6	98.63
meeting room	99.36	94.85	90.74	91.04	92.42
kitchen	99.32	88.97	80.57	80.57	80.56

5.5 Deployment of DNN using WSN Techniques

It has been established that the DeNNOTE system can achieve a high performance when determining if occupants are present. But, the DeNNOTE system is significantly limited since it is comprised of a single IoT node which means it can only monitor a small area. Using the WSN deployments that are defined in Chapter 4, this section will evaluate whether multiple nodes can be combined to monitor a larger area of a building. The cost and complexity of designing and deploying four WSNs are significant as explained in Section 2.5. To reduce these overheads and enable more resources to be dedicated to evaluating the deployments, a smart building mapping and simulation tool was developed. The tool is referred to as MIoTs and is described and evaluated in Chapter 3. The tool is evaluated in terms of its ability to simulate sensor data which is similar to

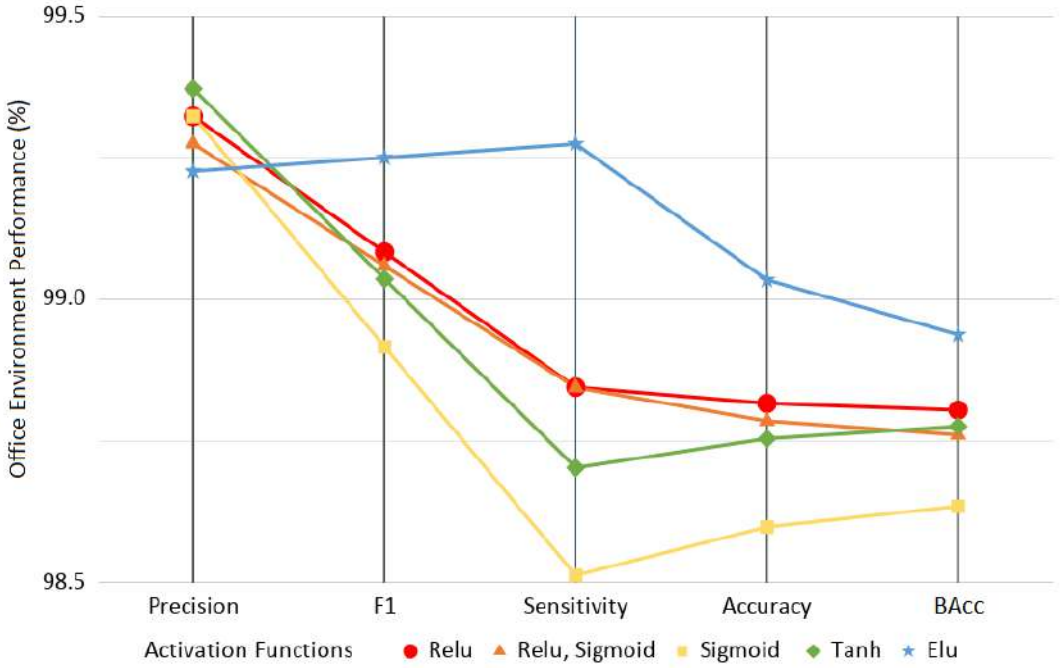


FIGURE 5.4: Test performance of DNN based on 2 training epochs and varied activation functions using office test data

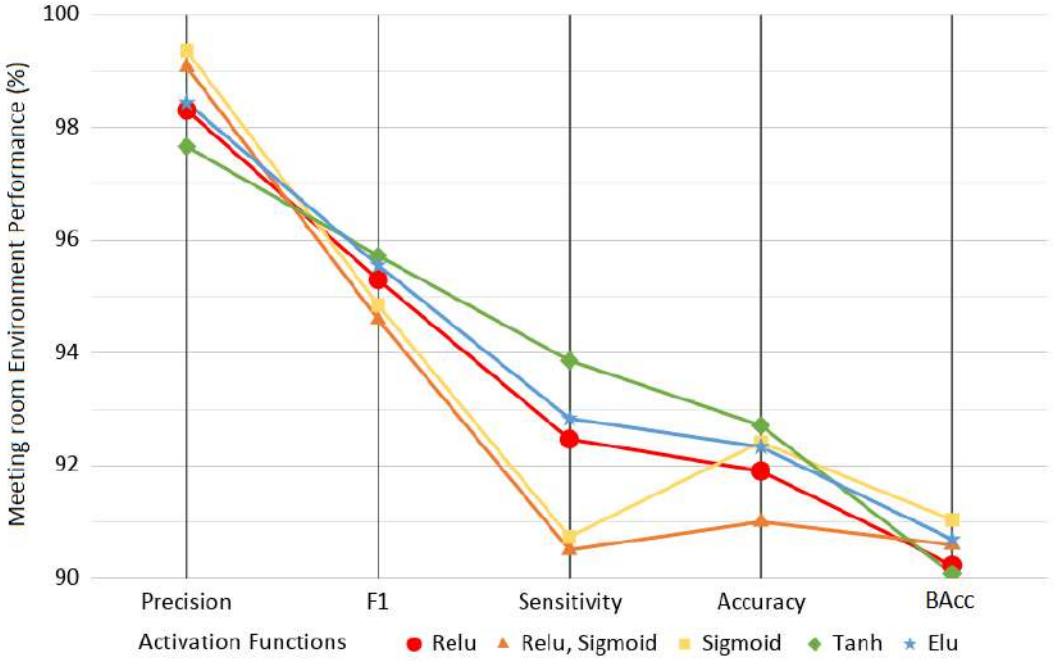


FIGURE 5.5: Test performance of DNN based on 2 training epochs and varied activation functions and meeting room test data

real-life data and the results demonstrate that the MIoTs simulator achieves this. The different WSNs are simulated using MIoTs simulation tool.

This combination of multiple DeNNOTE nodes will be evaluated using two

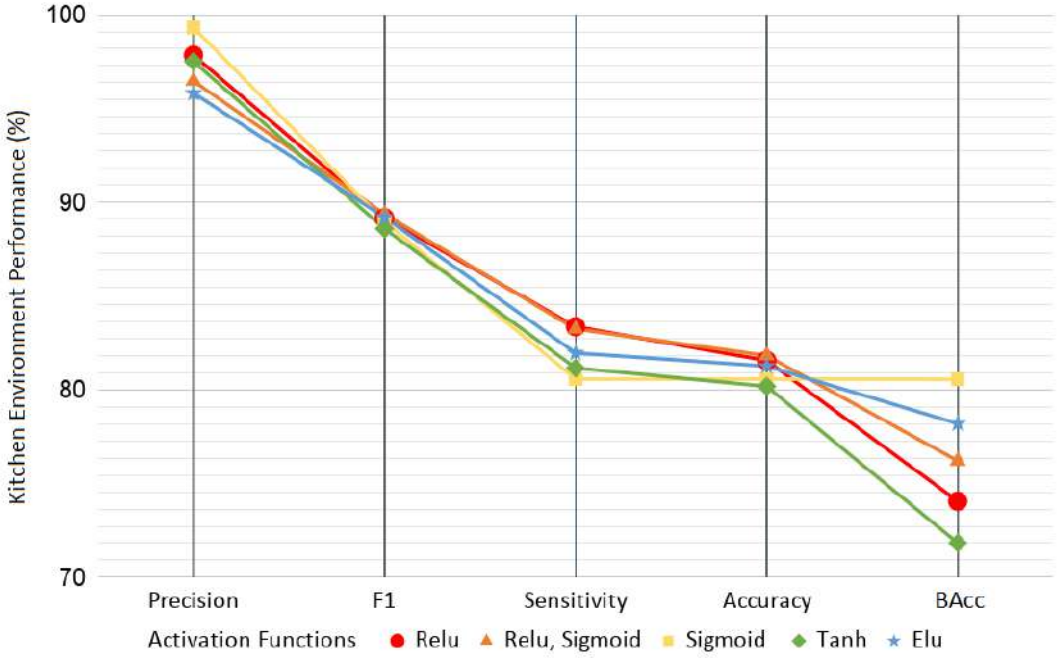


FIGURE 5.6: Test performance of DNN based on 2 training epochs and varied activation functions and kitchen test data

different network architectures. The first architecture will comprise nodes with different functions; sensor nodes which will capture both train and test data and processing nodes which will house the DNN to process the captured sensor data. The sensor data captured at multiple nodes will be transmitted to a single processing node for processing. In contrast, the second architecture will comprise multiple identical nodes; sensor nodes which also include the DNN. In this second architecture, the train and test data captured at each node will be processed locally on the same node by the integrated DNN. From both architectures, the processed data output from the DNN will be sent to a sink device to control the heating and lighting in the building. This evaluation will compare the cost of each architecture and the accuracy of occupancy and building monitoring data.

5.5.1 Evaluation of Deployment of Nodes with Separate DNN

The aim of this evaluation is to determine if multiple DeNNOTE systems can be combined to monitor a larger area of a building. The unprocessed sensor data of 4 nodes will be transmitted to a single processing node where the data is fused before being processed by the DNN. The processing node uses

the DNN to carry out binary classification and determine if occupants are present. The WSN configuration is illustrated in Fig 5.7.

MIoTs will be used to simulate each of the four WSN deployments defined in Chapter 4. The setup of MIoTs will be the same as that defined in Section 4.7. For each simulation two MIoTs variables are changed to define the node deployment technique; *pack_scheme* and *overlapped*. All of the other variables are kept constant. Each sensor node will include a PIRS, IRS and CO_2S. The simulation configuration is illustrated in Fig. 5.8. Each simulation will be compared based on the cost and performance to determine whether the network architecture affects the system's performance.

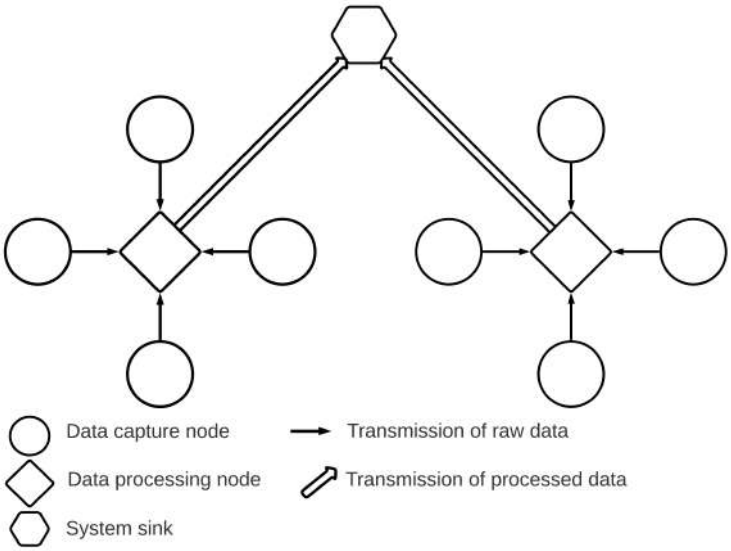


FIGURE 5.7: WSN architecture showing separate data capture and data processing

The sensor deployments are shown in Chapter 4 in Figs. 4.8a to 4.8d, where it was determined that the number of sensor nodes were 25, 33, 49 and 56 for the sensor grid, sensor hexagons, overlapping sensor hexagons and overlapping sensor grid deployments respectively, based on a room $40m \times 40m$ and a sensor coverage range $r = 4$. Additional hardware is required for the sink node which comprises a Raspberry Pi 3B+ with WiFi communication capabilities and sufficient memory to store the received sensor data and for the DNN to process the data. The sensor grid deployment requires 7 sink nodes, 9 for the sensor hexagon deployment, 13 for the overlapping sensor hexagons deployment and 14 for the overlapping sensor grid deployment.

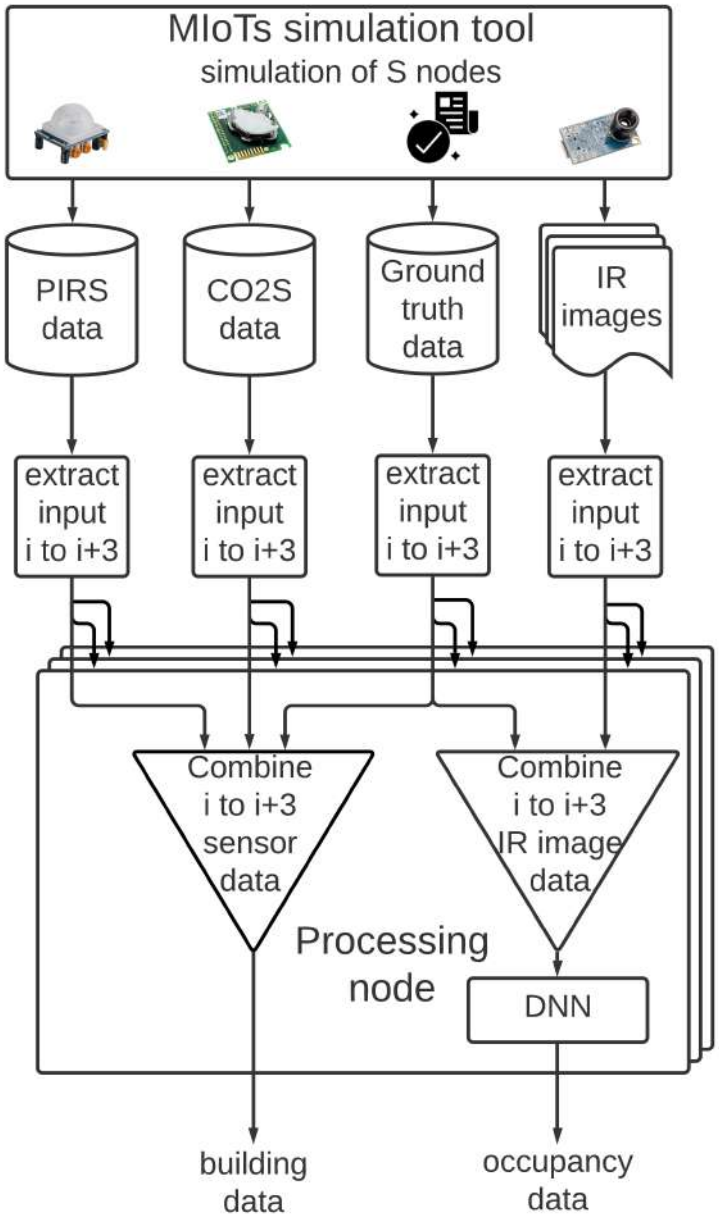


FIGURE 5.8: Simulation process illustrating creation, capture, fusion and processing of sensor data based on separate data capture and processing

Once per minute every sensor node will transmit sensor data to its corresponding sink node. Considering just the payload data and excluding control data; payload data includes a 32×24 pixel image and two pieces of numeric data. This is approximately 100 bytes (B) for the image, 2B for the CO_2 data and 1B for the PIR data, which is equal to $\sim$ 148KB per node per day. For the sensor grid deployment, this is $\sim$ 3.7MB data transmitted around the network each day, or $\sim$ 8.8MB for the overlapping sensor grid deployment. Based on receiving data from 4 sensor nodes, the memory card

on the sink node will require the sensor data to be wiped every few weeks. If data are not removed regularly, the reduced memory will inhibit the operation of the DNN and eventually disable the node entirely.

Similar to the single node PIRS data and the CO_2 data that was evaluated in Section 4.7.2, the fused PIRS data achieves 100% accuracy indicating the presence of occupants and the CO_2 data are only effected by the ventilation rate and the number of occupants. In comparison, fused IR images differ based on the different WSN deployment. Figs. 5.9a to 5.9h illustrate the fused sensor data and the corresponding room maps that illustrate the ground truth position of the occupants. The ground truth room maps were generated from the occupant position data generated by MIoTs. For the sensor grid and the sensor hexagon deployment, it can be seen that the number of occupants in the room map is identical to the number of hot zones in the corresponding IR images, but this is not true for the other deployments, as shown in Table 5.6. The overlapping deployments create errors within the fused IR images, which has the effect of increasing the number of heat regions. If blob detection was being used to count occupants, this would create errors. In comparison, the DNN does not count the number of occupants, instead, it determines whether occupants present or absent. This combined erroneous image data may change the heat patterns within the image, ie the nature of the features that the DNN is trained to find. This can result the DNN incorrectly detecting occupants, particularly if the heat regions were not generated by occupants but by inanimate objects such as radiator or computers. These combined erroneous images can reduce the accuracy of the DNN to determine whether occupants were present. The simulated data does include some regions of heat that are generated by a radiator, but not heat regions generated by a computer at each work station. It is difficult to test if duplication of non-occupant based heat regions affect the accuracy of the DNN.

TABLE 5.6: Number of actual occupants compared to detected occupants

	Sensor grid	Overlapping sensor grid	Sensor hexagons	Overlapping sensor hexagons
Actual	8	10	8	12
Detected	8	13	8	14

The average DNN performance for the different WSN deployments is

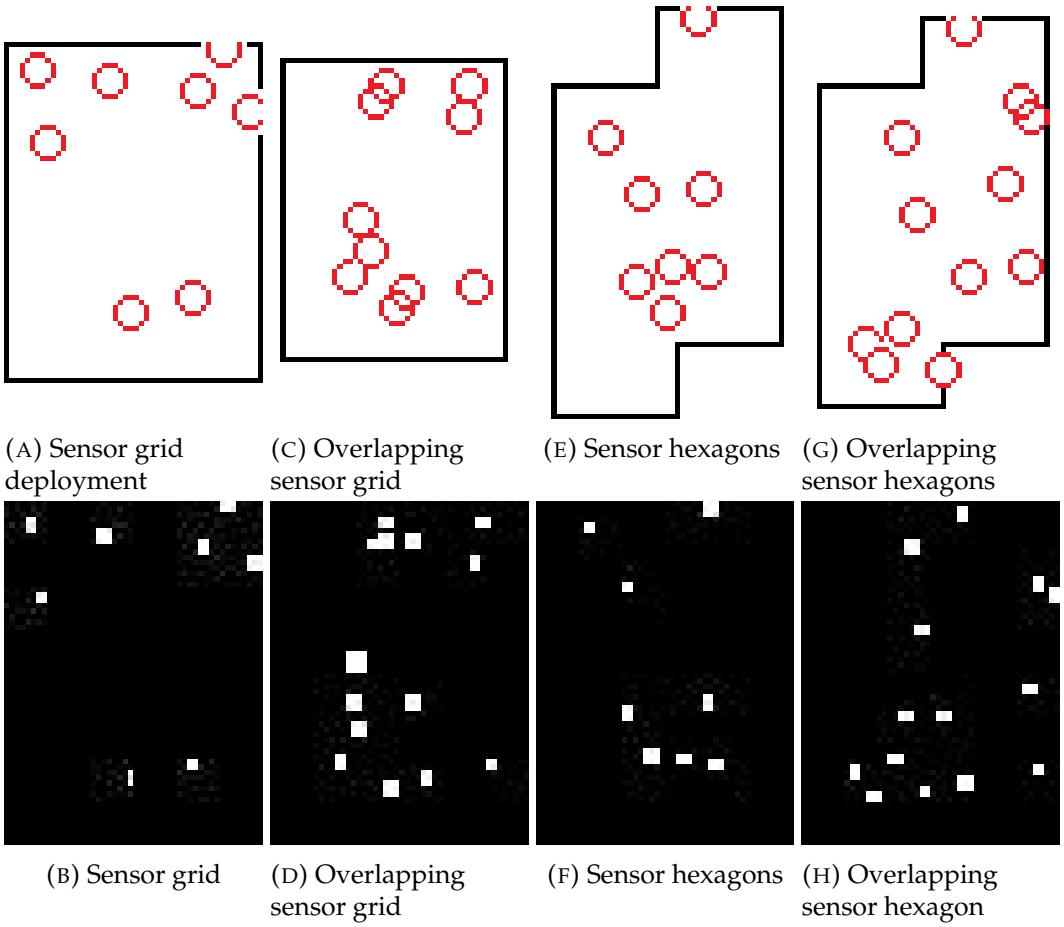


FIGURE 5.9: Room maps (top) and corresponding 4 fused IR images (bottom) for each WSN deployment

shown in Fig. 5.10. It can be seen that all four deployments achieve a very similar level of accuracy (93% to 96%) and BAcc (94% and 97%). The hexagonal deployment achieves the highest performance and the overlapping hexagonal deployment achieves the lowest performance. It is noted that the overlapping sensor hexagon deployment generates almost double the data of the sensor grid deployment and the overlapping sensor grid deployment generates more than double the amount of data compared to the sensor grid deployment. This significant increase in data does not produce an improvement in the DNN's performance compared to the sensor grid deployment. But the additional data transmissions and data processing may cause issues including increased power, processing time and data storage. Since the monitoring and processing are carried out on edge devices, the whole system is resource-constrained and therefore consideration of how these resources are used is important.

All four deployment techniques achieve a similar level of accuracy when

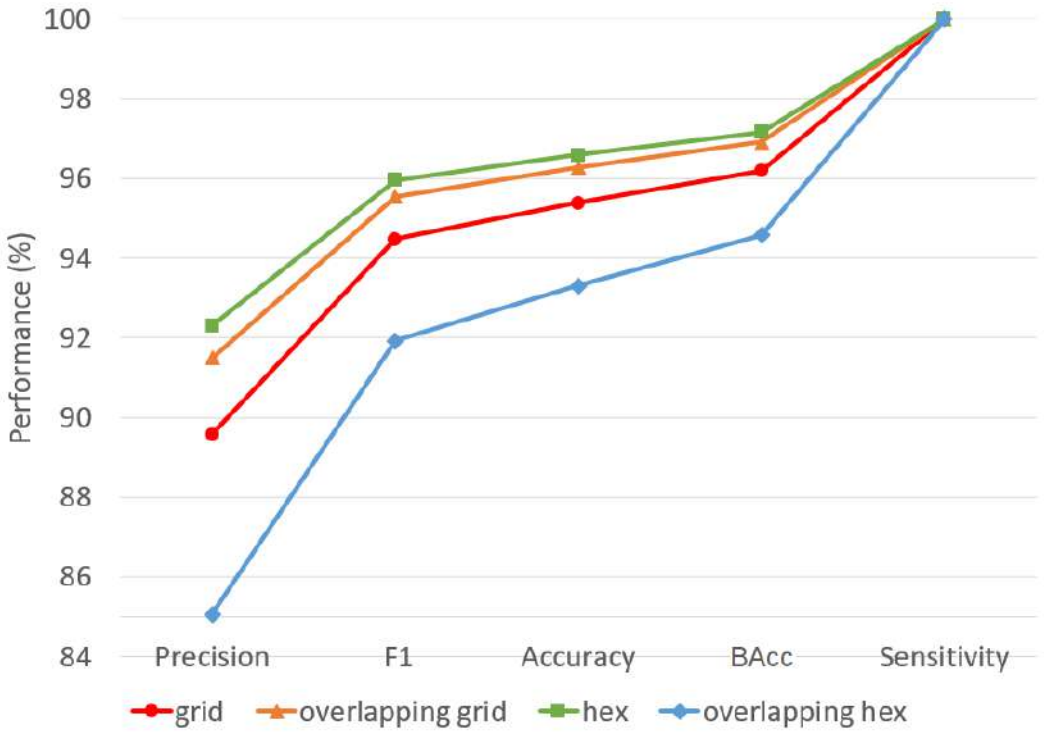


FIGURE 5.10: Graph showing the performance of the DNN based on processing data fused from 4 sensor nodes, comparing 4 WSN deployments

their simulated data are processed by the DNN to determine whether occupants were present or absent.

5.5.2 Evaluation of Deployment of Nodes with Integrated DNN

The cost of deploying a DNN within each sensor node significantly reduces the hardware cost in comparison to architecture with the DNN in a separate sink node. This is because the DNN does not require any hardware to be added to the existing sensor node since it already includes a suitable processor and sufficient memory. For this evaluation, the trained DNN model, DNN-S2e, is integrated in the sensor node. The WSN configuration is illustrated in Fig 5.11.

Again, MIoTs will be used to simulate each of the four WSN deployments and the setup of MIoTs will be the same. The simulation configuration is illustrated in Fig. 5.12. Each simulation will be compared based on the cost and performance to determine whether the network architecture affects the system's performance.

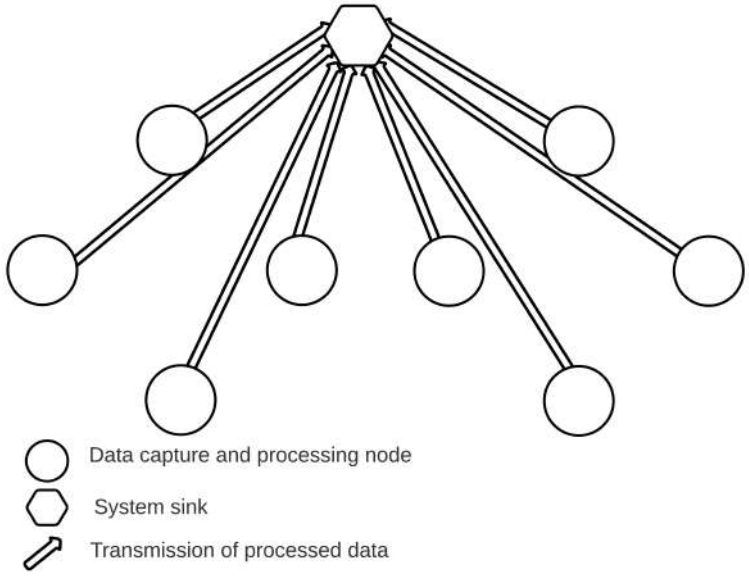


FIGURE 5.11: WSN architecture showing integrated data capture and data processing

The sensor data are processed on the same node where it is captured. The result of this integration is that unprocessed sensor data are not transmitted across the network, instead only the processed data will be transmitted. The processed data will include the output of the DNN indicating the presence of occupants and whether the building monitoring data meets the occupant comfort levels. The building data including CO_2 levels and temperature are evaluated against the occupant comfort levels by a program installed on the sensor node processor. The processed data will include a single byte of payload data. The first bit will indicate occupant / no occupant. The next four bits will indicate CO_2 levels and temperature levels, where 00, 01 or 10 will indicate levels are correct / too high / too low respectively. Each sensor node will transmit 1B of payload data (excluding control data) every minute, totalling $\sim$ 1.44KB per node per day. This 1.44KB is in comparison to $\sim$ 148KB transmitted per sensor node per day in the alternate architecture with separate sensor and sink nodes. For both architectures, to prevent the node's memory becoming full, sensor data can be cleared as soon as it has been processed by the DNN.

The average performance of the DNN for the different WSN deployments based on capturing and processing data on the same node is shown in Fig. 5.13. The deployments all achieve high performances, with accuracy (95% to 98%) and BAcc (96% to 98%). This performance is slightly better than what was achieved when the data was processed on a separate

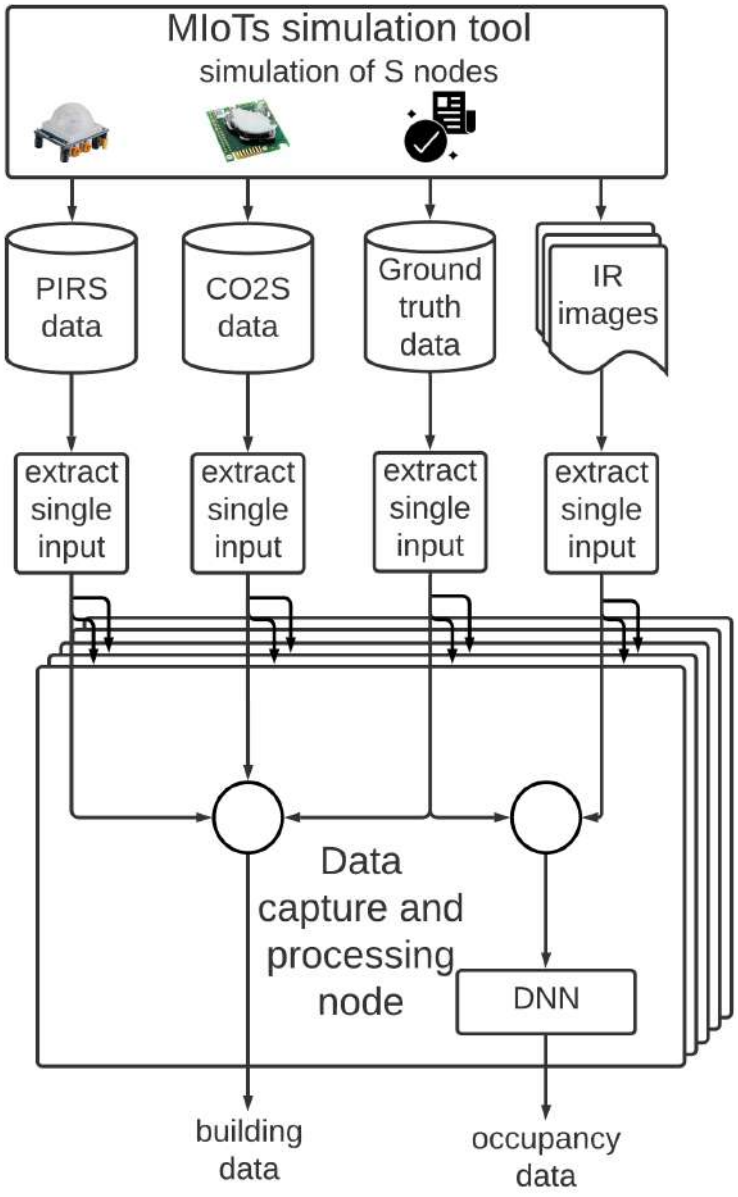


FIGURE 5.12: Simulation process illustrating creation, capture and processing of sensor data based on integrated data capture and processing

sink node. The overlapping sensor hexagon achieved the highest performance with an accuracy of 98% and BAcc of 98%.

Tables 5.7 and 5.8 summarise the performance of the two different DeNNOTE architectures, simulated in four different WSN deployments. It can be seen that the architecture with the DNN integrated into each sensor node requires significantly less hardware. This integrated architecture creates less duplicated data and less data traffic. It also achieves the highest level of performance compared to the alternate architecture. The alternate

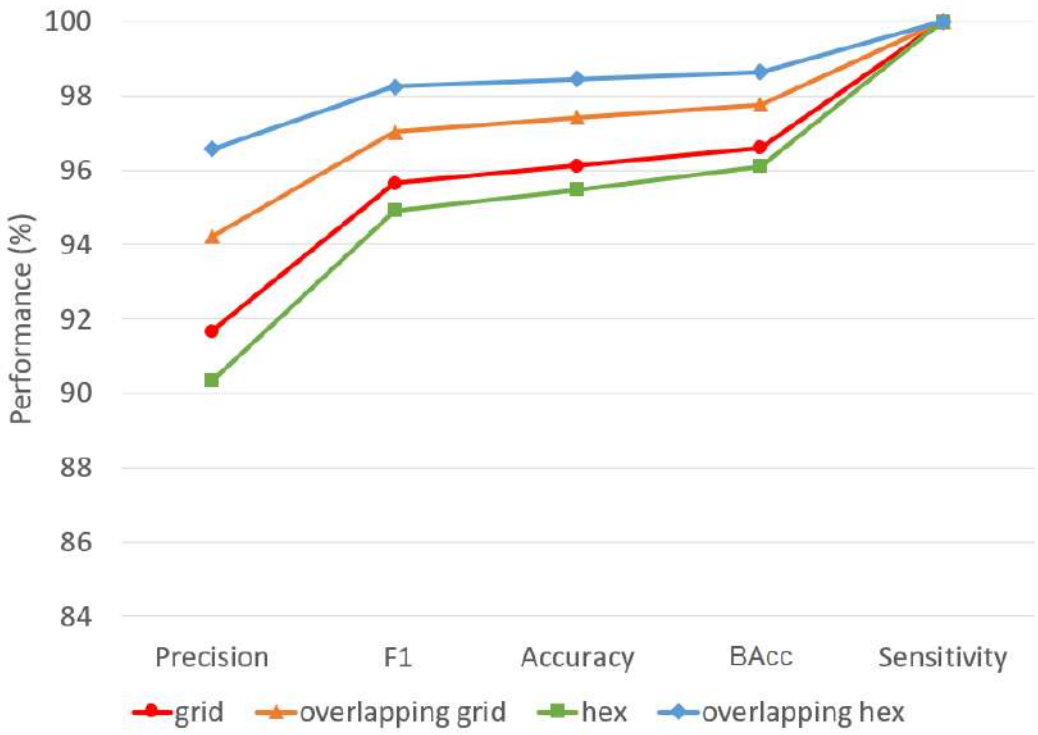


FIGURE 5.13: Graph showing the performance of the DNN based on processing data locally on sensor node, comparing 4 WSN deployments

architecture, with a sensor data transmitted from 4 sensor nodes to a separate sink node, requires more hardware and generates significantly more data. It seems that the duplication errors created when four IR images are fused has also slightly reduced the performance of the DNN. Comparing the networks based on the WSN deployment techniques; for both the separate DNN and integrated DNN architectures, the overlapping sensor grid and sensor grid consistently achieve the highest levels of performance. The sensor grid requires less than half the hardware of the overlapping sensor grid.

TABLE 5.7: Comparing 4 WSN deployments; each comprising groups of 4 sensor nodes transmitting data to separate sink node with DNN

	Sensor Grid	Overlapping Sensor Grid	Sensor Hexagons	Overlapping Sensor Hexagons
Node density	32	70	42	62
Daily data transmissions (payload data)	3.7MB	8.2MB	4.8MB	7.2MB
DNN performance (BAcc)	96.19%	96.91%	97.16%	94.59%

TABLE 5.8: Comparing 4 WSN deployments; each comprising DeNNOTE nodes with both sensors and integrated DNN

	Sensor Grid	Overlapping Sensor Grid	Sensor Hexagons	Overlapping Sensor Hexagons
Node density	25	56	33	49
Daily data transmissions (payload data)	36KB	80.1KB	47.5KB	70.6KB
DNN performance (BAcc)	96.61%	97.77%	96.10%	98.63%

5.6 Chapter Summary

Based on the financial barriers and IT limitations that SMEs face, a low cost and high-performance occupancy detection system comprising of a single input DNN has been developed and deployed on an edge-based system. The DNN is trained and tested with thermal IR data that was collected in a number of real building environments. The evaluations in Section 5.4.2 and 5.4.3 demonstrate that this system can be deployed across many similar building environments to control the heating and lighting, enabling the automation of reducing energy consumption, whilst maintaining occupant comfort levels. The DNN carries out binary classification to determine whether occupants are present. It has been developed to function well when trained with a large data set that does not require high computation image

processing, feature reduction or additional multimode monitoring data. By determining the optimal hyperparameters, the DNN (model DNN-S2e) has been optimised to achieve a high generalisation ability, demonstrating BAcc between $81\% - 99\%$ in different environments. These hyperparameters include two hidden dense layers, the Sigmoid activation function in the dense hidden layers and the number of training epochs is kept low during DNN training to reduce overfitting to the training data, which was observed while evaluating performance between $2 - 10$ epochs.

The DNN model, DNN-S2e, performed well in two similar environments, though its performance dropped slightly in a third kitchen environment. This was due to the rise in ambient temperature causing an increased rate of false negatives. As such, this demonstrates that an occupancy detection system wholly dependant on thermal data is not the most suitable system for warm environments such as kitchens or server rooms. The limitation of this work is that the DNN is only tested within three environments, with a maximum of six occupants. Due to cost and logistical limitations, it was not possible to deploy multiple DNN node simultaneously. Instead, eight different multi-node deployments were simulated, where four deployments were based on all of the sensor nodes transmitting their data to a separate sink node housing the DNN. The other four deployments were based on each sensor node having the DNN integrated for local data processing. The WSN deployment techniques that were introduced in Chapter 4 were used to define the positions of the simulated sensor nodes. This evaluation demonstrated that the sensor grid deployment with the DNN integrated with the sensor node is the optimal network architecture. This is because the architecture requires the minimum hardware, 25 nodes for a $40m \times 40m$ room, additionally, the DNN demonstrates a high rate of accuracy indicating whether occupants were present.

Chapter 6

IoT Node Optimisation

6.1 Introduction

In Chapter 4, a number of WSNs were evaluated, and the results showed that the sensor grid and the sensor hexagons deployments were optimal for occupant and building monitoring. In Chapter 5, the DeNNOTE system was developed for edge-based occupancy detection, and it achieved a high level of performance. In this chapter, these contributions will be further advanced to improve the performance of the system at a lower cost, by the addition of more sensors in each node and extending the DNN. In Chapter 7, these components will be combined to create an EMS to monitor a small commercial building and automate its energy reduction.

The remainder of this chapter includes Section 6.2, which details the reconfiguration of the node sensors. Section 6.3 describes the architecture of the two multimode (MM) DNNs In Section 6.4, four sensor configurations are assessed. In Section 6.5, the deployment of the sensor configurations are evaluated and in Section 6.6, the two MM DNNs are evaluated.

6.2 Reconfiguration of Node Sensors

The WSN deployments in Chapter 4 incorporated nodes with one PIRS, IRS, CO_2S and temperature sensor. It was assumed the combination of sensors enabled the node to form a circular coverage area. This section proposes the reconfiguration of the sensors in the node and presents how the combined sensor coverage area can be determined. The aim of this reconfiguration is to achieve a larger node coverage area and reduce coverage gaps that occur when the nodes are deployed in a WSN.

The CO_2S, PIRS and IRS all have defined coverage areas. In an enclosed environment, the K30 CO_2S has an approximate coverage area of $115m^2$ [154][1]. The HC-SR501 PIRS has a maximum coverage range of 7m and a FOV of 110° [144]. The MLX90640 IRS has a maximum coverage range of 11m and a FOV of $110 \times 70°$ [146]. Multiple PIRS or multiple IRS can be combined together in an *OR* configuration to create a larger single-mode coverage area. These single-mode sensors can then be combined together in an *AND* configuration to create a larger combined coverage area. The combined coverage area of the node is given in Eq. 6.1.

$$A_c = A_{\mathrm{CO}_2} \bigcap (A_{Pi} \bigcup_{i=1}^{m_P-1} A_{Pi+1}) \bigcap (A_{Ik} \bigcup_{k=1}^{m_I-1} A_{Ik+1}) \tag{6.1}$$

where A_c is the combined coverage area of the sensors in the node; A_{CO_2}[1], A_P and A_I are the coverage areas of a single CO_2S, PIRS and IRS respectively; m_P and m_I are the number of PIRS and IRS in a single node.

The combined multimodal data that are captured by the node, $\mathbf{x_{Cc}}$[1], $\mathbf{x_{Pc}}$ and $\mathbf{x_{Ic}}$ respectively, are given in Eqs. 6.2 to 6.4.

$$\mathbf{x_{Cc}} = \mathbf{x_C} \tag{6.2}$$

$$\mathbf{x_{Pc}} = \mathbf{x_{Pi}} \bigoplus_{i=1}^{m_P-1} \mathbf{x_{Pi+1}} \tag{6.3}$$

$$\mathbf{x_{Ic}} = \mathbf{x_{Ik}} \bigoplus_{k=1}^{m_I-1} \mathbf{x_{Ik+1}} \tag{6.4}$$

where $\mathbf{x_C}$, $\mathbf{x_P}$ and $\mathbf{x_I}$ are the individual data samples captured by a single CO_2S, PIRS and IRS respectively. Symbol $\oplus$ indicates that the individual samples are concatenated together.

A shape that can be repeated without the creation of any gaps is a tessellating shape [155]. A square is one of the simplest tessellating shapes. Most rooms inside buildings are rectangular [156], therefore combining the sensors in a node, such that they form a rectangular coverage area will

[1]Due to the large coverage area of the CO_2S, a single sensor will be included in each node.

enable a WSN deployment to have no coverage gaps[2]. The combined coverage area will be redefined in Eq. 6.5 to determine the size of the rectangular region encapsulating the combined coverage area. This region will be referred to as the tessellating coverage area.

$$A_t = \{A_c\} \tag{6.5}$$

where A_t is the tessellating node coverage area and Symbol { } indicates the sample is transposed with its tessellating rectangle.

In Section 6.4, a number of sensor configurations are assessed to find the maximum tessellating node coverage area. In the next section, the DNN that was developed in Chapter 5 will be expanded.

6.3 Multimode DNN

The DNN that has been proposed and evaluated thus far is a single-mode (SM) model, i.e., it analyses a single type of input data. The input is thermal data, which is analysed to determine whether occupants are present. The DNN achieved a high level of performance across a number of environments, though the performance was lowest in the environment with a higher ambient temperature. In this Section, the DNN architecture is further developed to create two MM DNN architectures which will process both IR data and additional data. This will improve the performance of the DNN model when the discriminating features in the IR data are distorted by the ambient room temperature.

The state of occupancy will still be defined as a binary classification problem. The two MM DNN architectures will be developed and assessed, with respect to their ability to learn the discriminating features of multimodal sensor data. The DNN models will learn the discriminating features in a supervised manner using the corresponding ground-truth labels, $\mathbf{y} \in \mathbb{R}^a$. The multimodal data inputs include $\mathbf{X_N} \in \mathbb{R}^{a \times j}$ and $\mathbf{X_I} \in \mathbb{R}^{a \times c \times d \times e}$. The data input $\mathbf{X_N}$ is comprised of j single dimension, standardised numeric inputs $\mathbf{x_N}$, captured by a CO_2S, PIRS and temperature sensor. Before being processed and analysed by the DNN, each dimension of numeric data is standardised as stated in Eq. 6.6.

[2]A rectangle is not a tessellating shape, but it can be used to create repeatable coverage regions that do not create coverage gaps based on keeping all of the rectangles identically orientated.

$$\mathbf{x_N} = \frac{x_{Ni} - \overline{x_N}}{s} \tag{6.6}$$

where x_{Ni} is a single numeric data point from $\mathbf{x_N}$, $\overline{x_N}$ is the mean value of $\mathbf{x_N}$ and s is the standard deviation of $\mathbf{x_N}$.

As it was earlier stated in Section 5.2, $\mathbf{X_I}$ is the multi-dimension thermal image data that is captured by a single IRS; a is the batch size of input data; c the thermal image width; d the thermal image height; and e the thermal image channels.

Based on the high performance of the DNN which was developed in Chapter 5, this DNN architecture is adapted to enable it to simultaneously process multimode building data. This purpose of adapting the existing DNN is to determine if additional building data can further improve the performance of the DeNNOTE system which has been developed and deployed as a low cost edge-based system, without significantly increasing the system's computational requirements or hardware requirements. DNN-2 and DNN-3 differ from each other based on when the two modes of data are combined; i.e. in DNN-2, image and numeric data are combined before they are processed, in comparison, in DNN-3, both the image data and numeric data are processed separately before being combined and processed further. The DNN-2 architecture is presented in Algorithm 5 and shown in Fig. 6.1. DNN-2 has a single analysis branch and is comprised of a flatten layer, a concatenation layer, a number of NN layers and a dense output layer. The DNN-3 architecture is presented in Algorithm 6 and shown in Fig. 6.2. This DNN has two NN analysis branches, a concatenation layer and a number of neural network (NN) layers.

6.3.1 DNN-2: Single-branch Multimode DNN Architecture

The proposed single-branch MM DNN architecture presented in Algorithm 5 is comprised of two input branches and a DNN which is similar to the SM DNN proposed in Section 5.2. The branch shown on the left of Fig. 6.1 is comprised of a flatten layer. The flatten layer transforms the dimension of a sample of $\mathbf{X}_{\mathrm{I}}$ from $1 \times c \times d \times e$ into $1 \times g$ as stated in Eq. 6.7.

$$\mathbf{f} = \beta\left(\mathbf{X}_{\mathrm{I}}\right), \tag{6.7}$$

where $g = c \times d \times e$; and $\beta(\cdot)$ is the flatten function.

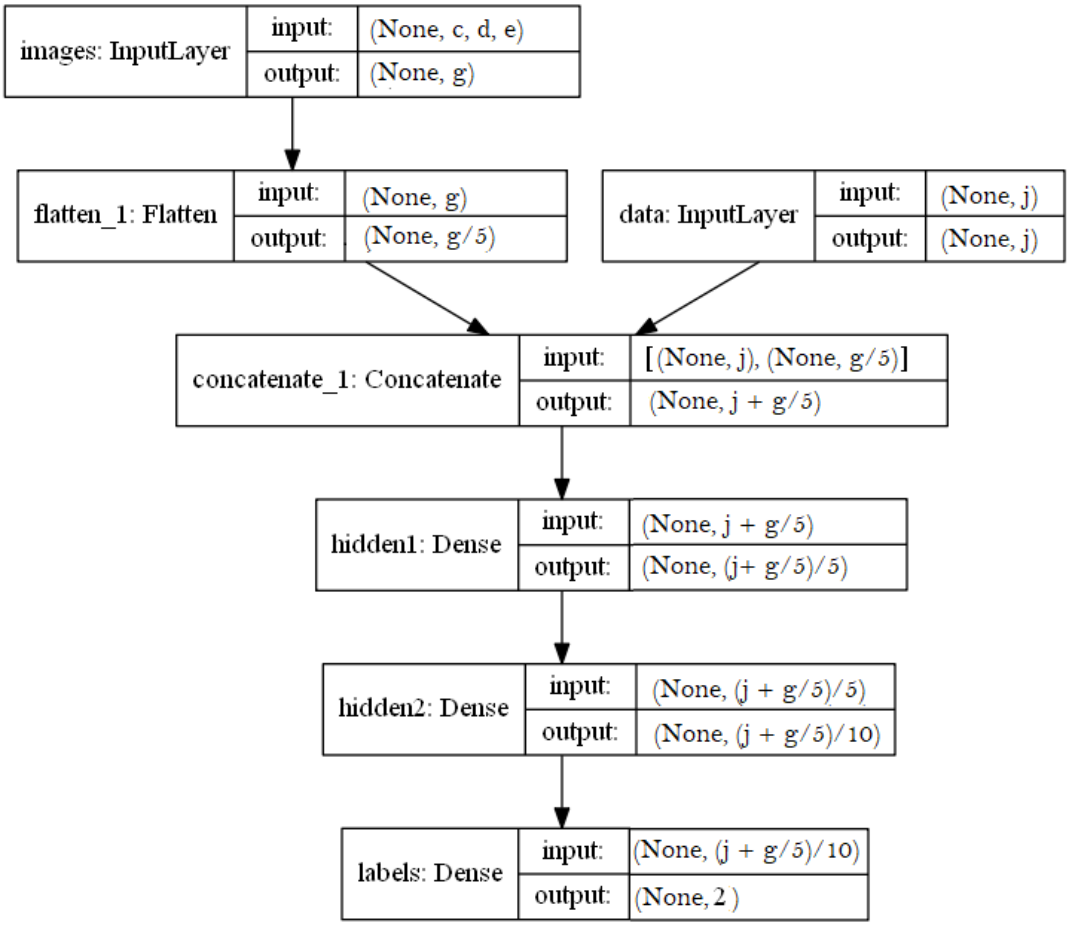


FIGURE 6.1: DNN-2: Single-branch Multimode DNN Block Diagram

A sample of numeric data, $\mathbf{X}_{\mathrm{N}}$, has dimensions $1 \times j$. Both the numeric data and the flattened IR image data are single dimension vectors. The concatenation layer combines the vectors as stated in Eq. 6.8.

$$\mathbf{c}_c = \mathbf{f} + \mathbf{x}_{\mathrm{N}} \tag{6.8}$$

The first hidden dense layer transforms the concatenated batch data using a weight matrix, $\mathbf{W}_1 \in \mathbb{R}^{q \times q}$, a bias vector, $\mathbf{b}_1 \in \mathbb{R}^{1 \times q}$ and an activation function, $\sigma_1(\cdot)$, as stated in Eq. 6.9.

$$\mathbf{h}_1 = \sigma_1 \left(\mathbf{c}_c \cdot \mathbf{W}_1 + \mathbf{b}_1\right), \tag{6.9}$$

where $q = g + j$. The next hidden dense layer transforms the output of the previous hidden layer using a weight matrix, $\mathbf{W}_2 \in \mathbb{R}^{q \times q}$, a bias vector, $\mathbf{b}_2 \in$

$\mathbb{R}^{1\times q}$ and an activation function, $\sigma_2(\cdot)$, as stated in Eq. 6.10.

$$\mathbf{h}_2 = \sigma_2\left(\mathbf{h}_1 \cdot \mathbf{W}_2 + \mathbf{b}_2\right), \tag{6.10}$$

The dense output layer transforms the output of the second hidden layer using an activation function $\phi(\cdot)$ as stated in Eq. 6.11.

$$\tilde{\mathbf{y}} = \phi\left(\mathbf{h}\right) \tag{6.11}$$

The difference between $\tilde{\mathbf{y}}$ and $\mathbf{y}$ is given by Eq. 6.12.

$$L = \sum_{k=1}^{n} \theta(\mathbf{y}_k, \tilde{\mathbf{y}}_k), \tag{6.12}$$

where θ is the sparse categorical cross-entropy loss function; and n is the number of samples in a batch of data.

Similar to the SM DNN model, to minimise the loss, the trainable parameters of the MM DNN model are adjusted over u epochs, based on the current loss score L and the optimiser named Adam, denoted ψ, [145] as stated in Eq. 6.13.

$$\mathbf{W}^{'}_{(\cdot)}, \mathbf{b}^{'}_{(\cdot)} = \psi\left(L, \mathbf{W}_{(\cdot)}, \mathbf{b}_{(\cdot)}\right), \tag{6.13}$$

where $\mathbf{W}_{(\cdot)}$ are the previous weight matrices; $\mathbf{W}^{'}_{(\cdot)}$ are the new weight matrices; $\mathbf{b}_{(\cdot)}$ are the previous bias vectors; and $\mathbf{b}^{'}_{(\cdot)}$ are the new bias vectors.

6.3.2 DNN-3: Multi-branch Multimode DNN Architecture

The proposed multi-branch MM DNN architecture presented in Algorithm 6 is comprised of two branches. The branch shown on the left of Fig. 6.2 is a replica of the SM DNN proposed in Section 5.2, which analyses the multi-dimension thermal image data. This branch is comprised of a flatten layer and two dense hidden layers. The flatten layer transforms the dimension of a sample of $\mathbf{X}_\mathrm{I}$ from $1 \times c \times d \times e$ into $1 \times g$ as stated in Eq. 6.14.

$$\mathbf{f} = \beta\left(\mathbf{X}_\mathrm{I}\right), \tag{6.14}$$

where $g = c \times d \times e$; and $\beta(\cdot)$ is the flatten function.

Algorithm 5: DNN-2: Single-branch Multimode DNN Algorithm

$\mathbf{f} = \beta(\mathbf{X}_I)$
for $i = 1$ *to* u **do**
  for $k = 1$ *to* n **do**
    $\mathbf{c}_{ci,k} = \mathbf{f}_k + \mathbf{x}_{Nk}$
    $\mathbf{h}_{1i,k} = \sigma_1 \left(\mathbf{c}_{ci,k} \cdot \mathbf{W}_{1i} + \mathbf{b}_{1i}\right)$
    $\mathbf{h}_{2i,k} = \sigma_2 \left(\mathbf{h}_{1i,k} \cdot \mathbf{W}_{2i} + \mathbf{b}_{2i}\right)$
    $\tilde{\mathbf{y}}_{i,k} = \phi \left(\mathbf{h}_{2i,k}\right)$
    $L_{i,k} = \theta(\mathbf{y}_{i,k}, \tilde{\mathbf{y}}_{i,k})$
  end
  $L_i = \sum_{k=1}^{n} L_{i,k}$
  $\mathbf{W}'_{i(\cdot)}, \mathbf{b}'_{i(\cdot)} = \psi \left(L_i, \mathbf{W}_{i(\cdot)}, \mathbf{b}_{i(\cdot)}\right)$
end
$L = \sum_{i=1}^{u} L_i$

The hidden dense layer, denoted $\mathbf{h}_2$, transforms the flattened batch of IR image data using a weight matrix, $\mathbf{W}_2 \in \mathbb{R}^{g \times g}$, a bias vector, $\mathbf{b}_2 \in \mathbb{R}^{1 \times g}$ and an activation function, $\sigma_2(\cdot)$, as stated in Eq. 6.15.

$$\mathbf{h}_2 = \sigma_2 \left(\mathbf{f} \cdot \mathbf{W}_2 + \mathbf{b}_2\right), \tag{6.15}$$

The next hidden dense layer, denoted $\mathbf{h}_3$, transforms the output of the previous hidden layer using a weight matrix, $\mathbf{W}_3 \in \mathbb{R}^{g \times g}$, a bias vector, $\mathbf{b}_3 \in \mathbb{R}^{1 \times g}$ and an activation function, $\sigma_3(\cdot)$, as stated in Eq. 6.16.

$$\mathbf{h}_3 = \sigma_3 \left(\mathbf{h}_2 \cdot \mathbf{W}_3 + \mathbf{b}_3\right), \tag{6.16}$$

The concurrent branch, that is shown on the right, is a MLP branch which analyses j single dimensions of numeric data. The MLP is comprised of a single dense hidden layer denoted $\mathbf{h}_1$, which transforms a sample of $\mathbf{X}_\mathrm{N}$ using a weight matrix, $\mathbf{W}_1 \in \mathbb{R}^{j \times j}$, a bias vector, $\mathbf{b}_1 \in \mathbb{R}^{1 \times j}$ and an activation function, $\sigma_1(\cdot)$, as stated in Eq. 6.17.

$$\mathbf{h}_1 = \sigma_1(\mathbf{x}_N \cdot \mathbf{W}_1 + \mathbf{b}_1) \tag{6.17}$$

The outputs of the DNN branch and the MLP branch have been transformed into single dimension vectors. The concatenation layer, denoted $\mathbf{c}_c$, combines

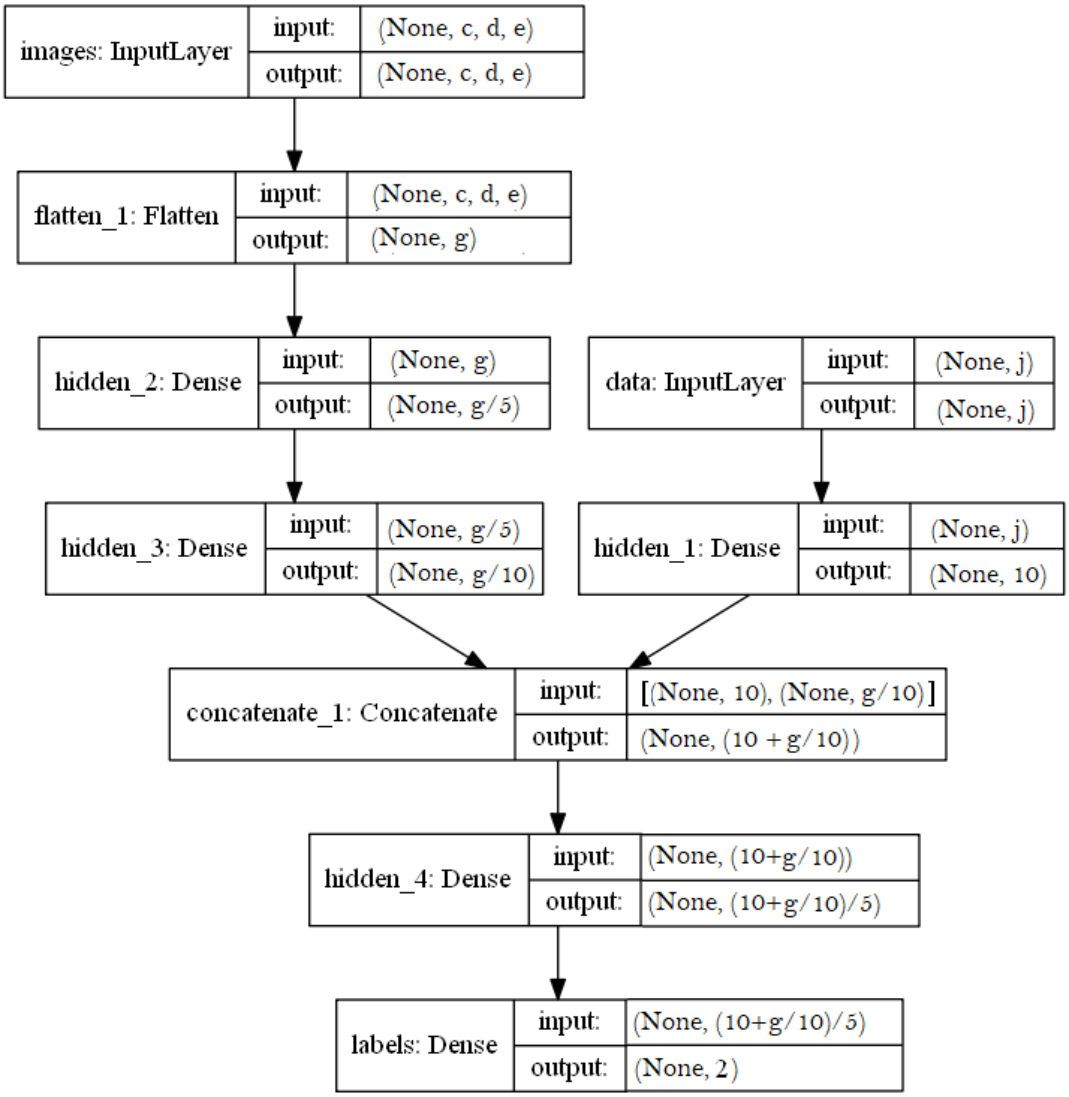


FIGURE 6.2: DNN-3: Multi-branch Multimode DNN Block Diagram

these vectors as stated in Eq. 6.18.

$$\mathbf{c}_c = \mathbf{h}_1 + \mathbf{h}_3 \tag{6.18}$$

The next hidden dense layer transforms the concatenated data using a weight matrix, $\mathbf{W}_4 \in \mathbb{R}^{q \times q}$, a bias vector, $\mathbf{b}_4 \in \mathbb{R}^{1 \times q}$ and an activation function, $\sigma_4(\cdot)$, as stated in Eq. 6.19.

$$\mathbf{h}_4 = \sigma_4 \left(\mathbf{c}_c \cdot \mathbf{W}_4 + \mathbf{b}_4 \right), \tag{6.19}$$

where $q = g + j$. The transformed concatenated data is further transformed by the dense output layer using an activation function, $\phi\ (\cdot)$, as stated in

Eq. 6.20.

$$\tilde{\mathbf{y}} = \phi\left(\mathbf{h}_4\right), \tag{6.20}$$

where $\tilde{\mathbf{y}}$ is the vector of labels predicted by DNN model. The loss, updated weight and bias remain the same as for DNN-2 and are stated in Eqs.6.12 to 6.13.

Algorithm 6: DNN-3: Multi-branch multimode DNN Algorithm

$\mathbf{f} = \beta(\mathbf{X}_I)$
for $i = 1$ *to* u **do**
  for $k = 1$ *to* n **do**
    $\mathbf{h}_{1i,k} = \sigma_1\left(\mathbf{x}_{Nk} \cdot \mathbf{W}_{1i} + \mathbf{b}_{1i}\right)$
    $\mathbf{h}_{2i,k} = \sigma_2\left(\mathbf{f}_k \cdot \mathbf{W}_{2i} + \mathbf{b}_{2i}\right)$
    $\mathbf{h}_{3i,k} = \sigma_3\left(\mathbf{h}_{2i,k} \cdot \mathbf{W}_{3i} + \mathbf{b}_{3i}\right)$
    $\mathbf{c}_{ci,k} = \mathbf{h}_{1i,k} + \mathbf{h}_{3i,k}$
    $\mathbf{h}_{4i,k} = \sigma_4\left(\mathbf{c}_{ci,k} \cdot \mathbf{W}_{4i} + \mathbf{b}_{4i}\right)$
    $\tilde{\mathbf{y}}_{i,k} = \phi\left(\mathbf{h}_{4i,k}\right)$
    $L_{i,k} = \theta(\mathbf{y}_{i,k}, \tilde{\mathbf{y}}_{i,k})$
  end
  $L_i = \sum_{k=1}^{n} L_{i,k}$
  $\mathbf{W}'_{i(\cdot)}, \mathbf{b}'_{i(\cdot)} = \psi\left(L_i, \mathbf{W}_{i(\cdot)}, \mathbf{b}_{i(\cdot)}\right)$
end
$L = \sum_{i=1}^{u} L_i$

The MM DNNs are both specified with the Sigmoid activation function in all dense hidden layers and are trained over 5 epochs. The MM DNNs are evaluated with the `Data.i5` train and test data set, which was defined in Section 5.3.2 to evaluate the SM DNN model. The numeric data, $\mathbf{X_N}$, has $j = 3$ dimensions, such that the numeric data includes three multimodal inputs: captured by a PIRS; CO_2S; and temperature sensor. Before training the MM DNN models, the data was processed to ensure that each sample included both an IR image and the corresponding numeric data. Where either the IR image or the numeric data were not available, the sample was removed from the MM data set. The numeric data includes normalised PIR data, CO_2 data and temperature data.

Following processing, the resulting multimodal train and test data sets were significantly smaller than those used to assess the SM DNN. The

multimodal train data set comprised 1100 samples and the multimodal test data sets, collected in the Office, Meeting room and Kitchen, each comprise 770 samples. Due to the significant reduction in the size of the train and test data sets, the SM DNN is also retrained with the corresponding 1100 IR data samples. This retrained version of the SM DNN will be referred to as DNN-1.

The performance of the MM DNNs will be evaluated in Section 6.6. In Section 6.4, the reconfiguration of the sensors in the IoT node is evaluated.

6.4 Evaluation of Reconfigured Node Sensors

In Section 6.2, the reconfiguration of the sensors in the node was proposed. The aim was to increase the coverage of each node and reduce coverage gaps that occur in a WSN deployment. This section analyses five sensor configurations. Applying Eqs. 4.3 and 6.5, the node's tessellating coverage area, A_t and space coverage are determined for each configuration. It will be assumed that the nodes will be positioned on a ceiling at a height of 3m.

6.4.1 Node Sensors: Single PIRS

The first node configuration that is assessed is that of a single PIRS. The coverage of a single PIRS is shown in Fig. 6.3. The resulting coverage circle and tessellating coverage square are shown in Fig. 4.1. The coverage area and tessellating coverage area are given in Eq. 6.21 and 6.22 respectively.

$$A_{P,1} = \pi(r)^2 \tag{6.21}$$

$$\begin{aligned} A_{t\{P,1\}} &= \{A_{P,1}\} \\ A_{t\{P,1\}} &= l_p \,.\, w_p = (2r)^2 \\ A_{t\{P,1\}} &= (2h\tan(\frac{f_{\mathrm{P}}}{2}))^2 \\ A_{t\{P,1\}} &= (6\tan 55)^2 \\ A_{t\{P,1\}} &= 73.4m^2 \end{aligned} \tag{6.22}$$

where l_p and w_p are the length and width of the PIRS coverage square, $l_p = w_p = 2r$, r is the radius of the PIRS coverage area,

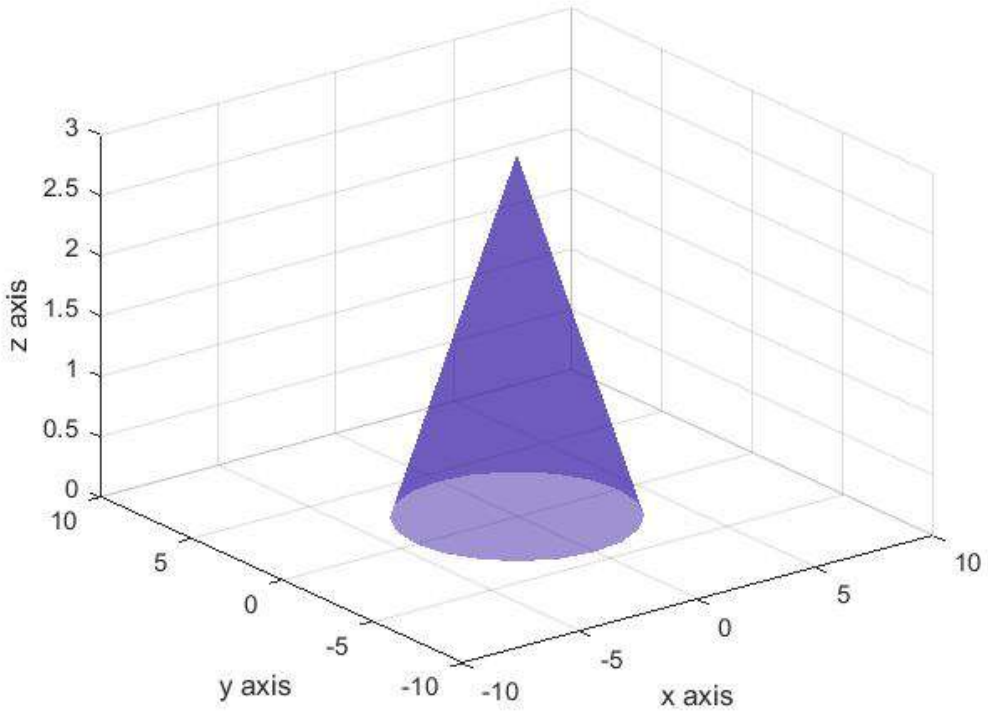


FIGURE 6.3: 3D coverage of a single PIRS

$r = (h \tan \frac{f_p}{2})$, h is the height at which the node is positioned above the ground, $h = 3$, f_{P} is the FOV of the PIRS, $f_{\mathrm{P}} = 110°$. This node configuration achieves a tessellating coverage area which is a square of dimensions $8.57 \times 8.57m$. This configuration achieves space coverage of 78.54%, resulting in 21.46% coverage gaps.

6.4.2 Node Sensors: Single IRS

The second node configuration that is assessed is that of a single IRS, which is shown in Fig. 6.4. The coverage area and tessellating coverage area are given in Eq. 6.23 and 6.24 respectively.

$$A_{I,1} = l_I \,.\, w_I \tag{6.23}$$

$$
\begin{aligned}
A_{t\{I,1\}} &= \{A_{I,1}\} = A_{I,1} = l_I \,.\, w_I \\
A_{t\{I,1\}} &= 2(h \tan \frac{f_{\mathrm{Il}}}{2}) \,.\, 2(h \tan \frac{f_{\mathrm{Iw}}}{2}) \\
A_{t\{I,1\}} &= 2(3 \tan 55) \,.\, 2(3 \tan 35) \\
A_{t\{I,1\}} &= 36.0m^2
\end{aligned} \tag{6.24}
$$

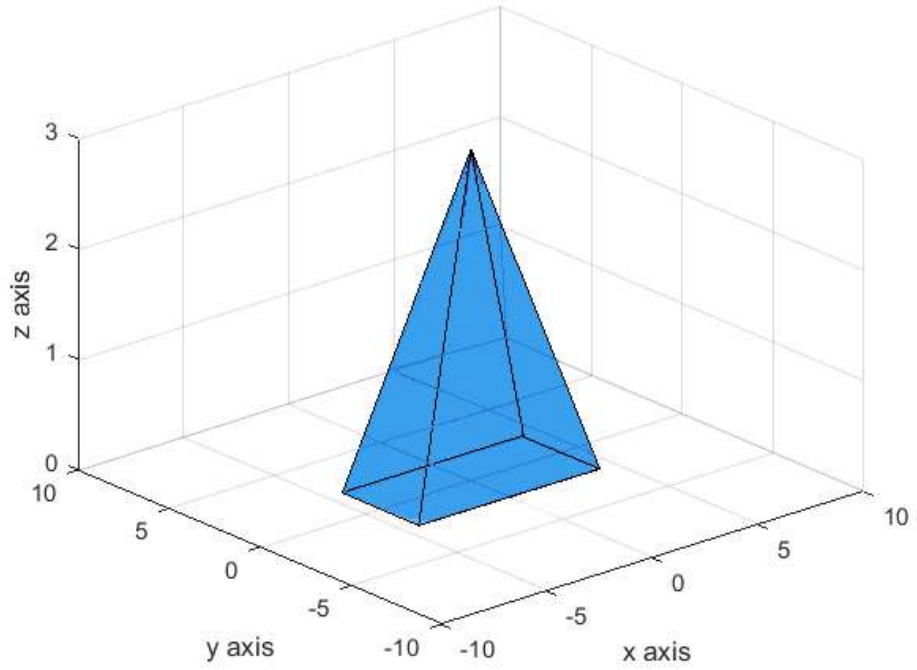


FIGURE 6.4: 3D coverage of a single IRS

where l_I and w_I are the length and width of the IRS coverage area, $l_I = 2(h \tan \frac{f_{Il}}{2})$ and $w_I = 2(h \tan \frac{f_{Iw}}{2})$, f_{Il} and f_{Iw} are the length-way's and width-way's FOV of the IRS respectively, $f_{\mathrm{Il}} = 110°$ and $f_{\mathrm{Iw}} = 70°$. This configuration achieves a tessellating coverage area which is a rectangle of dimensions $8.57 \times 4.20m$. This is less than half the tessellating coverage area of the single PIRS. This configuration achieves space coverage of 100%.

6.4.3 Node Sensors: Combine Single PIRS and Single IRS

The third node configuration that is assessed is a single PIRS and single IRS. This is the original node configuration that was deployed in Chapter 4 and 5. The coverage is shown in Fig. 6.5. The combined coverage area is given by Eq. 6.25 and the tessellating coverage area is given and Eq. 6.26.

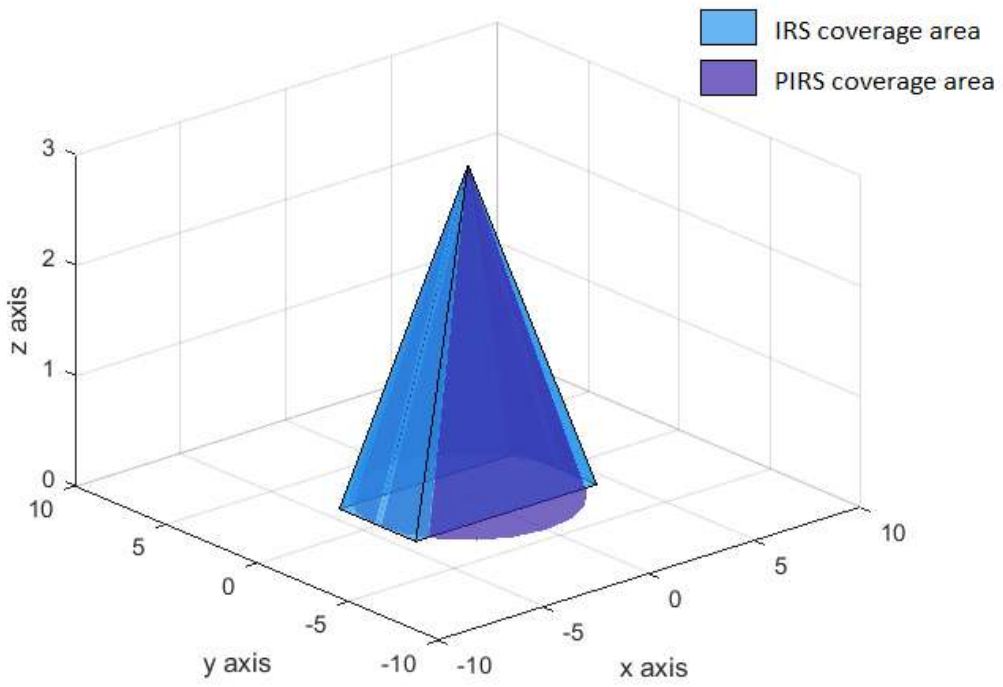


FIGURE 6.5: 3D coverage of a single PIRS and IRS

$$A_c = A_{P,1} \bigcap A_{I,1} \tag{6.25}$$

$$\begin{aligned} A_{t\{P,1\bigcap I,1\}} &= \{A_c\} = l_P \,.\, w_I \\ A_{t\{P,1\bigcap I,1\}} &= 2(h \tan \frac{f_{\mathrm{P}}}{2}) \,.\, 2(h \tan \frac{f_{\mathrm{Iw}}}{2}) \\ A_{t\{P,1\bigcap I,1\}} &= 2(3 \tan 55) \,.\, 2(3 \tan 35) \\ A_{t\{P,1\bigcap I,1\}} &= 36.0m^2 \end{aligned} \tag{6.26}$$

It can be seen from Eq. 6.24 and 6.26 that the combined tessellating coverage area of this configuration, with a single PIRS and a single IRS, is equal to the tessellating coverage area of a single IRS. This is because the tessellating rectangle that encapsulates the combined coverage area is equal to the IRS coverage area. This configuration achieves a tessellating coverage rectangle of dimensions $8.57 \times 4.20m$. This configuration achieves space coverage of 95.9%.

6.4.4 Node Sensors: Combine Single PIRS and Two IRS

The next configuration that is assessed is a single PIRS, combined with two IRS. The coverage of this configuration is shown in Fig. 6.6. To enable the coverage areas of the two IRS to be combined, the direction of the IRS is varied; one IRS is angled by 35°, the other by $-35°$ with respect to the zy axis. This results in their coverage areas being positioned side-by-side along their lengths. The combined coverage area and tessellating coverage area are given in Eq. 6.27 and 6.28.

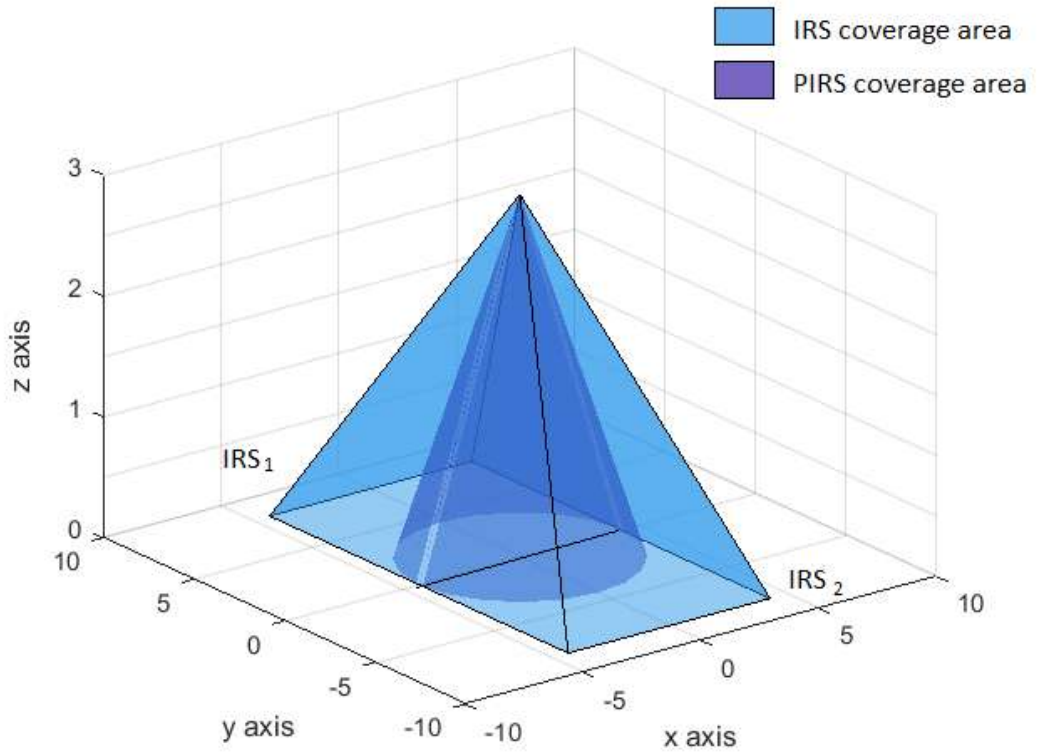


FIGURE 6.6: 3D coverage area of a single PIRS and two IRS

$$A_c = A_{P1} \bigcap (A_{I1} \bigcup A_{I2}) \tag{6.27}$$
$$A_c = \pi r^2 \bigcap ((l_{I1}.w_{I1}) + (l_{I2}.w_{I2}))$$

$$
\begin{aligned}
A_{t\{P,1\bigcap I,2\}} &= \{A_c\} \\
A_{t\{P,1\bigcap I,2\}} &= (2r)^2 = (2h\tan\frac{f_{\mathrm{P}}}{2})^2 \\
A_{t\{P,1\bigcap I,2\}} &= (6\tan 55)^2 \\
A_{t\{P,1\bigcap I,2\}} &= 73.4m^2
\end{aligned}
\tag{6.28}
$$

where l_{I1}, w_{I1}, l_{I2} and w_{I2} are the length and width of the first and second IRS's coverage area. It can be seen from Eq. 6.22 and 6.28, the combined tessellating coverage area of this configuration is equal to the tessellating coverage area of the single PIRS. This is because the region where both of the sensor coverage areas overlap is equal to the PIRS coverage area. This sensor configuration achieves a tessellating coverage square of dimensions $8.57 \times 8.57m$. It also achieves space coverage of 78.54%, resulting in 21.46% coverage gaps.

6.4.5 Node Sensors: Combine Four PIRS and Two IRS

A final node configuration, comprising 4 PIRS combined with 2 IRS, is assessed. In this configuration, rather than directing all of the sensors towards the ground, they are positioned at an angle, as shown in Fig. 6.7. Two of the PIRS sensors are angled at $\pm 55°$ along the zy axis, the other two at $\pm 55°$ along the zx axis. The two IRS are angled at $\pm 35°$ along the zy axis. The coverage of this configuration is shown in Fig. 6.8. The combined coverage area is given by Eqs. 6.29 and the tessellating coverage area is given by Eq. 6.30 to 6.34.

$$
A_c = (A_{P1}\bigcup A_{P2}\bigcup A_{P3}\bigcup A_{P4})\bigcap(A_{I1}\bigcup A_{I2}) \tag{6.29}
$$

$$
A_t = \{A_c\} = (l_{P1} + l_{P3}).l_{I1} = 2l_{P1}.l_{I1} \tag{6.30}
$$

where l_{P1} and l_{P3} are the length of coverage rectangle of PIRS_1 and PIRS_3 respectively and are equal to each other. These lengths can be determined using the dimensions of a right angled triangle positioned in the xy plane at $z = 0$[3]. As shown in Fig. 6.9, the first triangle vertex is positioned

[3] i.e. at ground level.

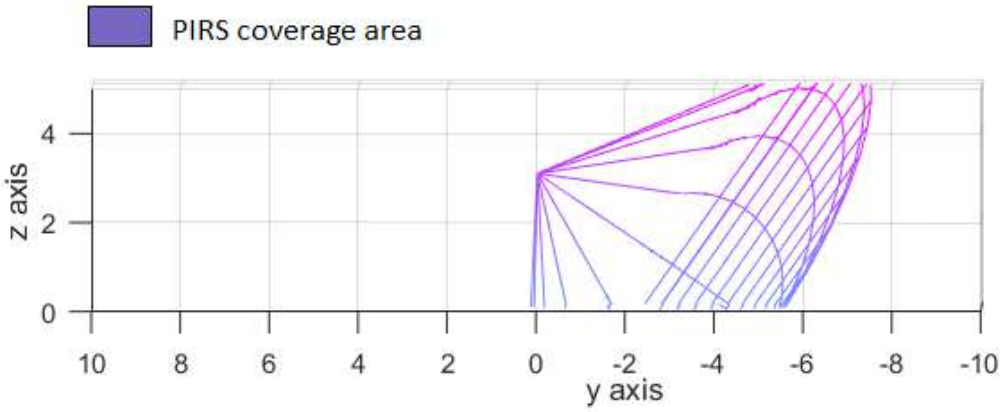


FIGURE 6.7: 3D side view of a single PIRS angled at -55° along zy axis

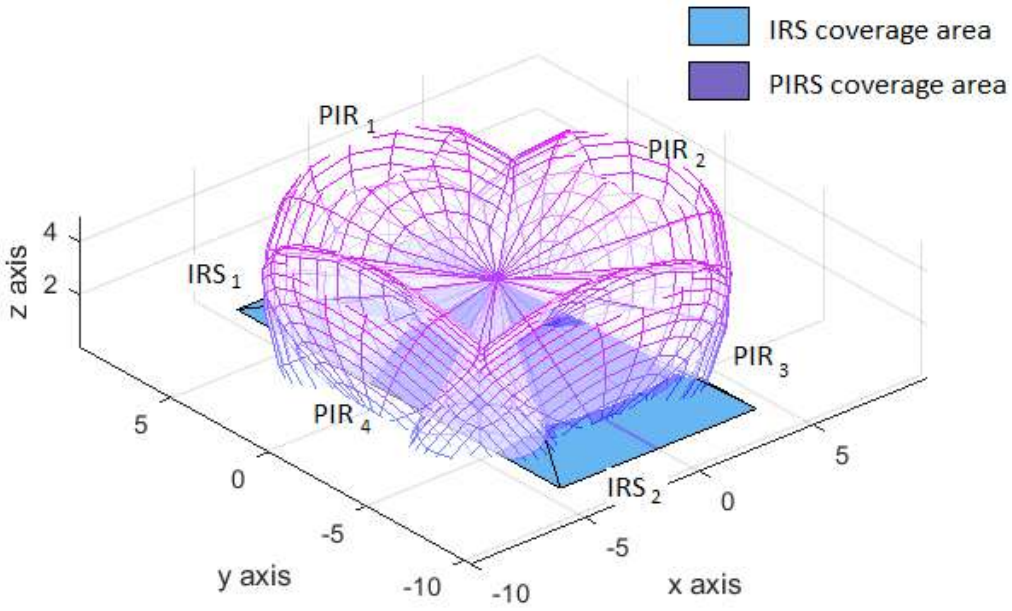


FIGURE 6.8: 3D coverage of four PIRS and two IRS (side view)

vertically below the sensors, denoted q. The next vertex is the intersection of the coverage of $PIRS_1$ and IRS_1, denoted s. The final vertex, t, is the intersection of two legs of the triangle, positioned is at x = -4.2 and y=0. Length $|qs|$ is given in Eq. 6.31, length $|qt|$ is given in Eq. 6.32, length $|st|$ is given in Eq. 6.33, $r_{\max}$ is the maximum range of the PIRS and $r_{\max} = 7$.

$$\begin{aligned} |qs| &= h \tan(cos^{-1} \frac{h}{r_{max}}) \\ |qs| &= 3 \tan(cos^{-1} \frac{3}{7}) \end{aligned} \tag{6.31}$$

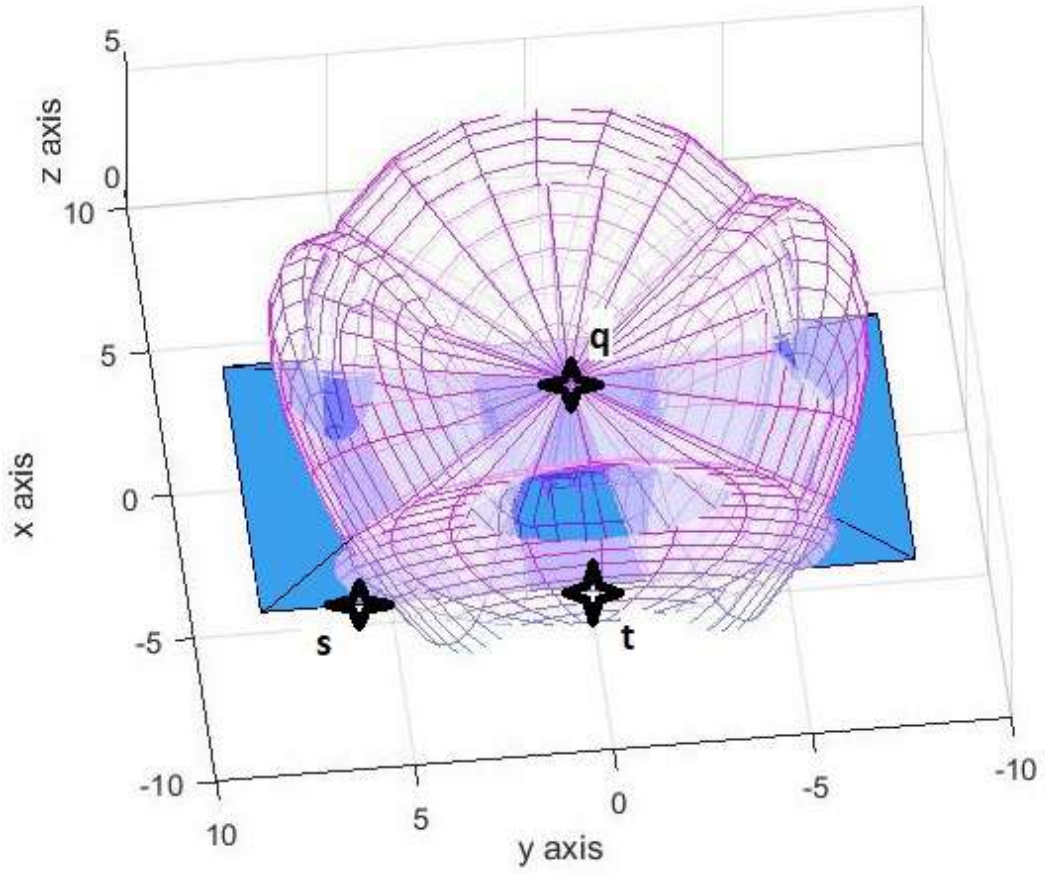


FIGURE 6.9: 3D coverage of four PIRS and two IRS (top view)

$$\begin{aligned} |qt| &= h \tan \frac{f_P}{2} \\ |qt| &= 3 \tan 55 \\ l_{I1} &= 2|qt| \end{aligned} \tag{6.32}$$

$$\begin{aligned} |st| &= l_{P1} = l_{P3} \\ |st| &= |qs| \cos\left(\sin^{-1} \frac{|qt|}{|qs|}\right) \end{aligned} \tag{6.33}$$

The tessellating coverage area is given in Eq. 6.34.

$$\begin{aligned} A_t &= 2l_{p1} \,.\, l_{I1} \\ A_t &= 2(3tan(cos^{-1}\frac{3}{7}) \,.\, cos(sin^{-1}\frac{3tan55}{3tan(cos^{-1}\frac{3}{7})}) \,.\, 2(3tan55) \\ A_t &= 9.30 \times 8.57 = 79.73m^2 \end{aligned} \tag{6.34}$$

This configuration, comprising four PIRS and two IRS, achieves the maximum tessellating coverage area of $79.73m^2$, a rectangle $9.30m \times 8.57m$. It also achieves space coverage of 100%, resulting in no coverage gaps. In Section 6.5, a number of WSN deployments will be simulated to assess three of these sensor configurations.

6.5 WSN Deployments to Evaluate the Reconfigured Node Sensors

In this section, three of the combined node configurations will be deployed as part of a WSN simulation with the aim evaluating the node configurations in terms of the hardware cost. The optimal WSN configuration is the network which has a low hardware cost but delivers a high level of space coverage. It is noted that the most expensive hardware component in each node is the processor and memory card. In comparison, the cost of the selected sensors are low. Therefore, it is more cost-effective to deploy fewer nodes containing more sensors, in comparison to deploying more nodes with fewer sensors. Based on the results achieved in Section 4.7.1, the sensor grid WSN deployment technique will be used throughout this evaluation. The coverage range of each node is defined in the previous section. Also, a range of different building layouts will be considered.

The MIoTs smart building mapping and simulation tool is used to determine the node density and the node positions. The node density and positions are based on the building size, building layout, deployment technique and tessellating node coverage area. MIoTs was developed for this research to generate WSN deployment maps. The tool is detailed in Chapter 3. The different The node configurations that are simulated include:

- config-1: 1× PIRS and 1× IRS.

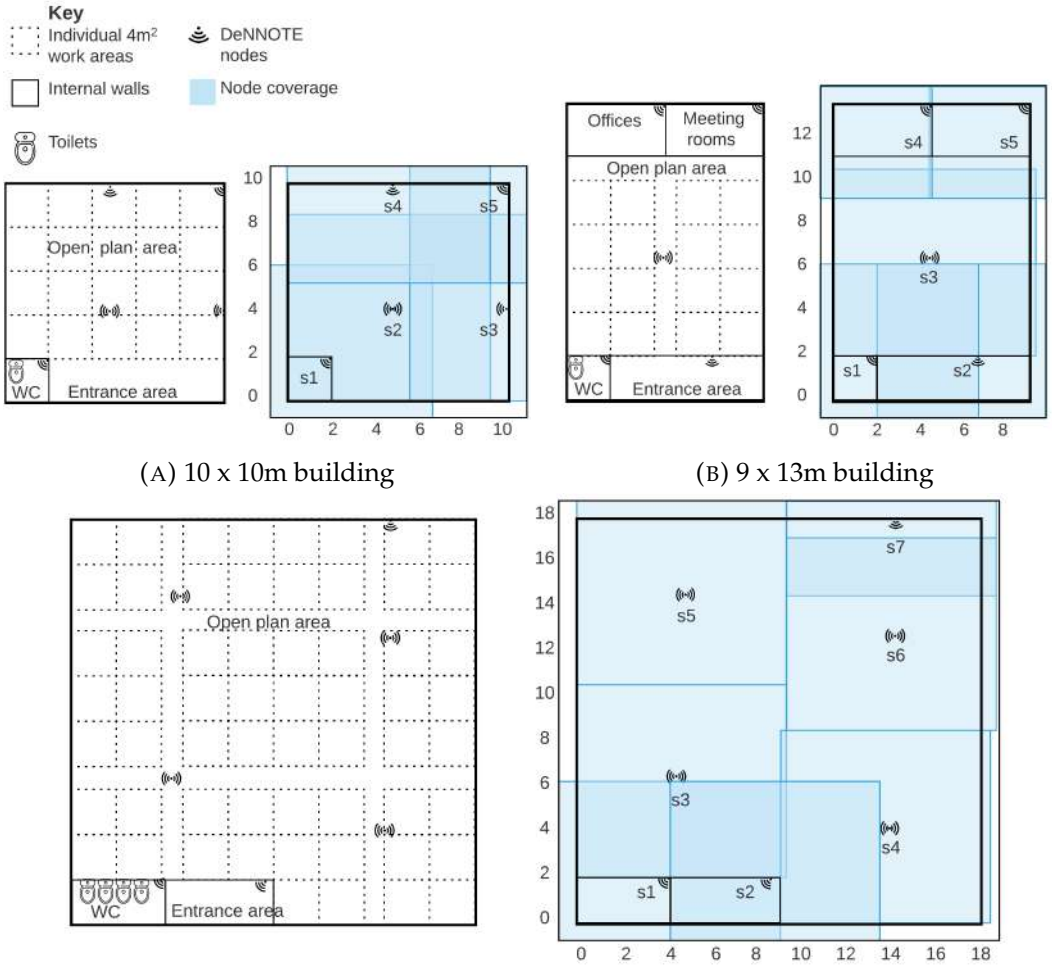
- config-2: 1× PIRS and 2× IRS.
- config-3: 4× PIRS and 2× IRS.

Three small commercial buildings will be simulated in this evaluation. Building A and B are similar sized small offices, Building C is a slightly larger small office. Building A is $10m \times 10m$ and open-plan except for the toilet facilities. Based on the HSE regulations regarding minimum working areas, this building can accommodate $20 - 25$ employees [59]. Node config-1 requires the most nodes, 8, in comparison to config-3, which only requires 5 nodes. config-2 achieves space coverage of 78.54%, config-1 and config-3 both achieve 100% space coverage. Building B is a similar size to the previous building, $9m \times 13m$, though it is configured differently. It includes a separate entrance area, an open-plan office area and a number of smaller rooms which could be used as offices or meeting rooms. Again, node config-1 requires the most nodes, 7 and config-3 requires the least, 5. Building C is a slightly larger office, approximately three times larger than the previous buildings. It measures $18m \times 18m$ and is open-plan, except for the entrance and toilet facilities. This building can accommodate approximately $60 - 70$ employees [59]. Based on the significant increase in floor area, node config-1 requires double the number of nodes, i.e. 15, compared to the previous buildings. In comparison, node config-3 only requires 7 nodes to achieve strong internal coverage. Again, config-2 achieves space coverage of 78.54%, config-1 and config-3 both achieve 100% space coverage.

Illustrations of these buildings, including their floor map and WSN deployment using node config-3, are given in Figs. 6.10a to 6.10c[4]. The node density for each node configuration is summarised in Table 6.1. Based on using the sensor grid deployment technique, the node sensor config-3, which includes 4 PIRS and 2 IRS, achieves optimal WSN since it consistently requires the minimum node density and achieves 100% coverage of the building. Config-1 also achieves 100% coverage but requires $40\% - 114\%$ more nodes than config-3, based on deployments in Building A - C.

In Chapter 7, node config-3 will be used as part an EMS which will be used to automate the reduction of energy consumption in a real commercial building.

[4]It should be noted, the sensor symbols have been manually added to the simulated deployment images to illustrate their position



(A) 10 x 10m building

(B) 9 x 13m building

(C) 18 x 18m building

FIGURE 6.10: Floor maps and simulated reconfigured node deployments for 3 small commercial buildings

TABLE 6.1: Comparison of node density based on different sensor combinations

	config-1	config-2	config-3
Building A	8	6	5
Building B	7	6	5
Building C	15	10	7

6.6 Evaluation of the Multimode DNNs

The aim of this evaluation is to compare the performance performance of the SM DNN, denoted DNN-1, against the performance of the two MM DNN models, denoted DNN-2 and DNN-3. All of the DNNs are specified with the Sigmoid activation function throughout the dense layers and trained over 5 epochs with a reduced train data set selected from `Data.i5`

and comprising 1100 multimodal samples. Tables 6.2 to 6.4 and Fig. 6.11 to 6.13 illustrate the performance of DNN-1 to DNN-3. Model DNN-S2e is the SM DNN model that was evaluated in Section 5.4.3 and demonstrated the highest generalisation ability across three different environments. This model was specified with the Sigmoid activation function in both hidden dense layers and was trained over 2 epochs with `Data.i5` comprising 9300 IR data samples. The performance of DNN-S2e is included for reference and is highlighted with a dashed yellow line on the graphs.

TABLE 6.2: Comparison of Performance of DNN-1, DNN-2 and DNN-3 with Office data (%)

Model	Precision	F1	Sensitivity	Accuracy	BAcc
DNN-1	99.00	99.05	99.10	99.17	99.16
DNN-2	99.13	99.20	99.28	99.10	99.08
DNN-3	99.16	99.19	99.22	99.08	99.07
DNN-S2e	99.32	98.92	98.51	98.6	98.63

TABLE 6.3: Comparison of Performance of DNN-1, DNN-2 and DNN-3 with Meeting room data (%)

Model	Precision	F1	Sensitivity	Accuracy	BAcc
DNN-1	51.37	64.41	86.64	90.68	88.90
DNN-2	98.39	94.18	90.32	89.84	87.63
DNN-3	97.55	94.36	91.38	90.25	86.42
DNN-S2e	99.36	94.85	90.74	91.04	92.42

TABLE 6.4: Comparison of Performance of DNN-1, DNN-2 and DNN-3 with Kitchen data (%)

Model	Precision	F1	Sensitivity	Accuracy	BAcc
DNN-1	78.67	66.27	58.04	83.24	75.98
DNN-2	94.03	90.75	87.75	84.75	77.80
DNN-3	92.69	91.19	89.86	85.77	78.95
DNN-S2e	99.32	88.97	80.57	80.57	80.56

Fig. 6.11 illustrates the performance of the SM and two MM DNN models, all trained with the small multimodal train data set and tested with the multimodal Office test data set. It can be seen that across all metrics, all three DNN models achieve a very similar level of performance, with BAcc between 99.1% to 99.2%. Both of the MM models achieve 0.1% higher precision, F1 and sensitivity metrics, compared to the SM model DNN-1. All

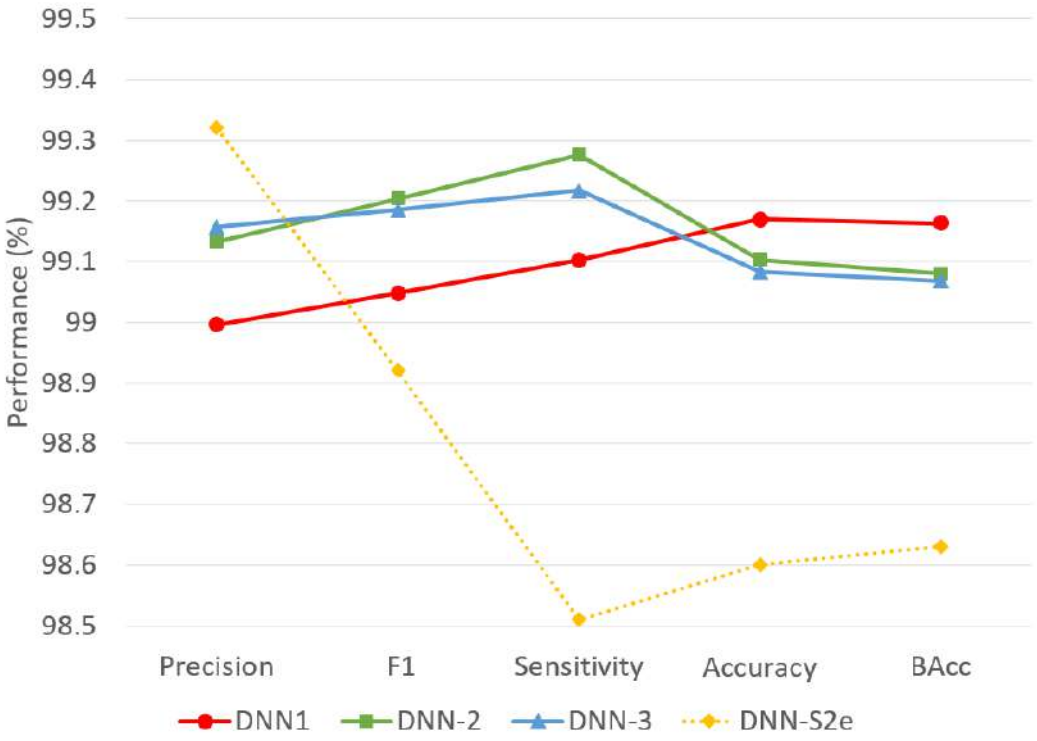


FIGURE 6.11: Test performance of single and multimode DNNs using Office test data

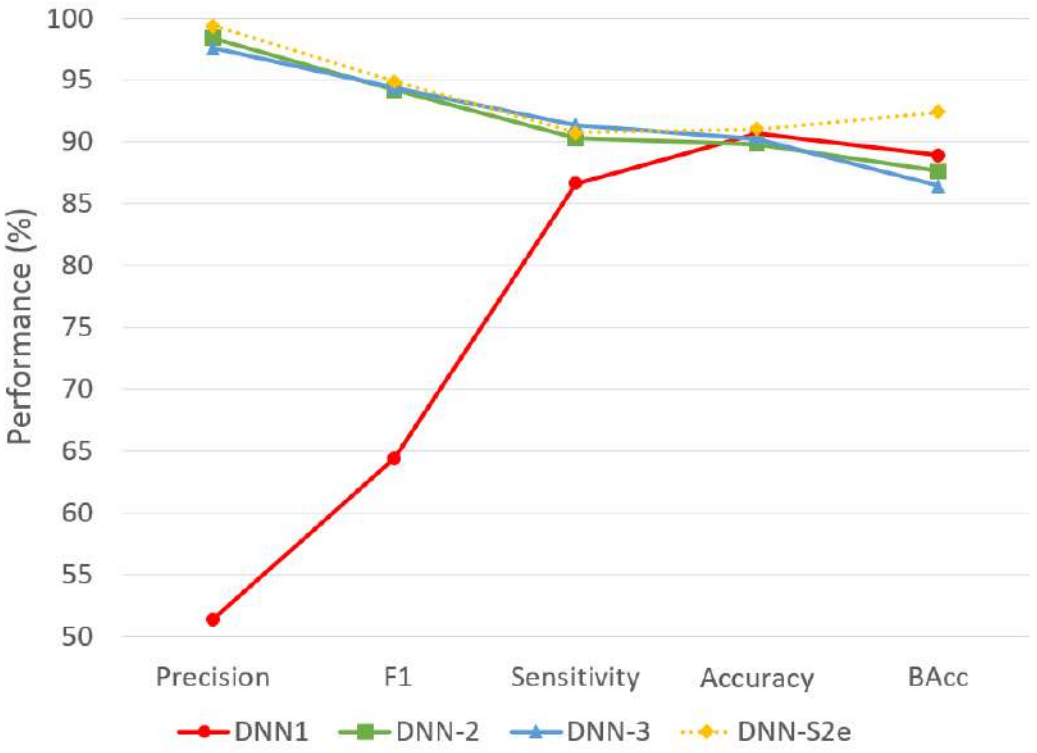


FIGURE 6.12: Test performance of single and multimode DNNs using Meeting room test data

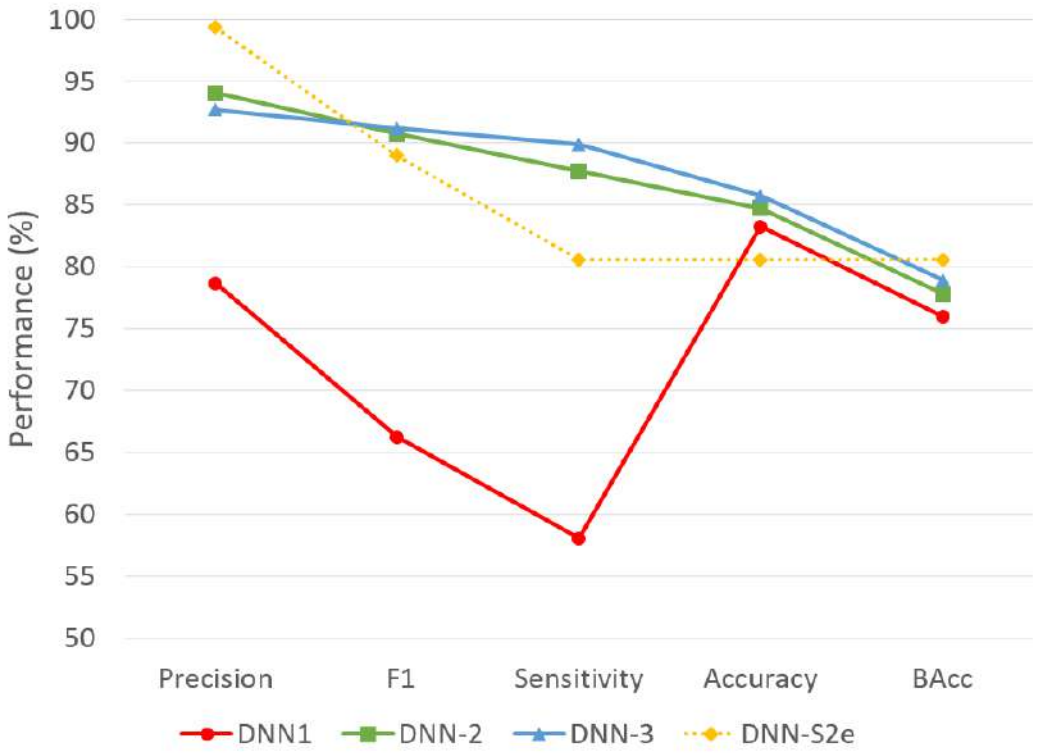


FIGURE 6.13: Test performance of single and multimode DNNs using Kitchen test data

of three of the DNN models, including the MM models and the SM DNN-1, all achieve 0.4% to 0.7% better BAcc and accuracy, compared to optimal SM DNN DNN-S2e.

Fig 6.12 illustrates the performance of the three DNN models based on the multimodal Meeting room test data. The SM DNN and both of the MM DNNs achieve similar BAcc (86.4% to 88.9%) and accuracy (88.3% to 90.7%). Considering the other performance metrics, both of the MM models achieve higher precision, F1 and sensitivity, compared to the SM model. This is particularly true for precision, where the MM DNNs improve upon the SM DNN's precision by more than 46%. Similarly, the F1 performance of the MM DNNs is almost 30% higher than the SM DNN. Considering the optimal SM DNN DNN-S2e, it achieves very similar performance to the MM DNNs, which is significantly better than the SM DNN DNN-1. A number of studies demonstrate that increasing the size of the train set also increases the performance of a DNN model. An increased train data set enables a DNN model to better learn the discriminating features of the data and reduces the effect of overfitting [157]–[159]. As such, based on training the MM DNNs with a larger data set, comparable in

size with `Data.i5`, the MM DNNs will achieve better performance in this environment than the SM DNN.

Fig 6.13 illustrates the performance of the three DNN models using the Kitchen test data. Across all metrics, the MM DNNs achieve better performance than the SM DNN. Considering the BAcc and accuracy, DNN-2 achieves 1.8% and 1.5% more respectively and DNN-3 achieves 3.0% and 2.5% more respectively, compared to the SM DNN. For precision, DNN-2 and DNN-3 achieve 15.3% and 14.0% more. For F1, DNN-2 and DNN-3 achieve 24.5% and 24.8% more. For sensitivity DNN-2 and DNN-3 achieve 29.7% and 31.8% more, all compared to the SM DNN. Due to improvements in precision, F1 and sensitivity demonstrated with the Kitchen test data, both of the MM DNNs significantly improve the performance of occupancy detection, compared to the SM DNN. Compared to the optimal SM DNN, DNN-S2e, both of the MM DNNs slightly improve upon the performance in this environment. Therefore, if the MM DNNs are trained with a larger data set, they will both achieve significantly better performance than the SM DNN [157]–[159], with DNN-3 achieving the best performance.

Both SM DNNs demonstrated lower performance in the environment with higher ambient temperature because the discriminating features in the IR thermal data were distorted and less distinguishable. In comparison, the MM DNNs are not solely dependant upon the thermal data, but also on the PIR data, CO_2 data and temperature data. This increase in data modes also increases the number of distinguishable features, which enables an improvement in the MM DNN performance. A further evaluation could investigate whether all four modes of data are required, or if less modes of data can achieve a similar high level of performance.

To summarise, considering the Office and Meeting room test data, all three DNNs achieve very similar BAcc and accuracy. The MM DNNs achieve better performance across the other metrics for these two environments. In comparison, for the kitchen test data, both of the MM DNNs perform better than the SM DNN, across all metrics. Therefore, based on training the MM DNNs with a larger data set, both of the MM DNN models will achieve similar or better performance in all environments, compared to the SM DNN. The increase in performance will be most significant in environments with a higher ambient temperature, such as a kitchen or server room. Overall when all of the DNN models are trained with a large data set, the

MM DNNs will demonstrate a higher generalisation ability than the SM DNN.

6.7 Chapter Summary

In this chapter, the sensor configuration of each node was altered by increasing the quantity of each type of sensor. The method to combine data and to determine the combined coverage area were proposed and assessed. Then the MIoTs tool was used to simulate the deployment of three of the sensor configurations. This enabled these configurations to be assessed by comparing node density and space coverage. The configuration that achieved the maximum coverage with the minimum number of sensor nodes included 4 PIRS and 2 IRS. Based on a $324m^2$ open-plan office, this configuration required less than half the number of nodes compared to the original sensor configuration with a single PIRS and a single IRS. This sensor configuration will significantly reduce the cost of EMS that is proposed in Chapter 7.

Additionally, the SM DNN was expanded to create two MM DNN models. The evaluation train data set contained less than 12% of the samples in the train data set that was used to evaluate the optimal SM DNN. For a fair evaluation, the MM DNNs needed to be evaluated against a comparable SM DNN that was trained with a similar-sized data set. To create a comparable SM DNN, the optimal SM DNN was retrained with the small data set. The evaluation of the three DNNs demonstrated that the MM DNNs performed slightly better than the comparable SM DNN in the Office and Meeting room and significantly better in the Kitchen. This improvement was due to the MM DNN using the discriminative features of the numeric data in addition to the features of the IR data. When the MM DNNs are trained with a large data set, they will demonstrate a significant improvement in their generalisation ability.

Since a large multimodal train data set is currently not available, the optimal SM DNN, DNN-S2e, will be part of the EMS that is proposed in Chapter 7.

Chapter 7

Occupant-centric IoT-based Energy Management System

7.1 Introduction

In this chapter, the contributions of Chapters 3 to 6 will be combined to inform the design of a complete real-world application. This application is an occupant-centric IoT-based energy management system which is designed specifically for SMEs based in small commercial buildings.

The **Oc**cupant-**C**entric IoT-based **EMS** referred to as OcCEMS, is a holistic solution which includes a network deployment strategy, data fusion, data analysis, data visualisation and user feedback. The network deployment strategy is the sensor grid WSN deployment technique which was analysed in Chapter 4. The sensor grid technique enables minimal IoT hardware to be deployed whilst achieving full building monitoring coverage. The DeNNOTE node, developed in Chapter 5 and improved in Chapter 6, is a high-performance IoT node which will be deployed using the sensor grid technique to monitor the building and its occupants through the use of non-intrusive thermal IR, PIR, $CO_2$and temperature sensors. This node utilises concatenation based data fusion techniques to combine multimodal data captured by the IoT sensors. The fused data are analysed by the integrated DNN model employing binary classification to determine whether occupants are present. The analysed data enables management of the building's heating and lighting systems to reduce energy consumption. In addition, energy consumption data are compared against a baseline and fed back in real-time to building users. The baseline model is proposed in Section 7.4 of this chapter. A functional representation of OcCEMS is given in Fig. 7.1, illustrating data collection, communication and management.

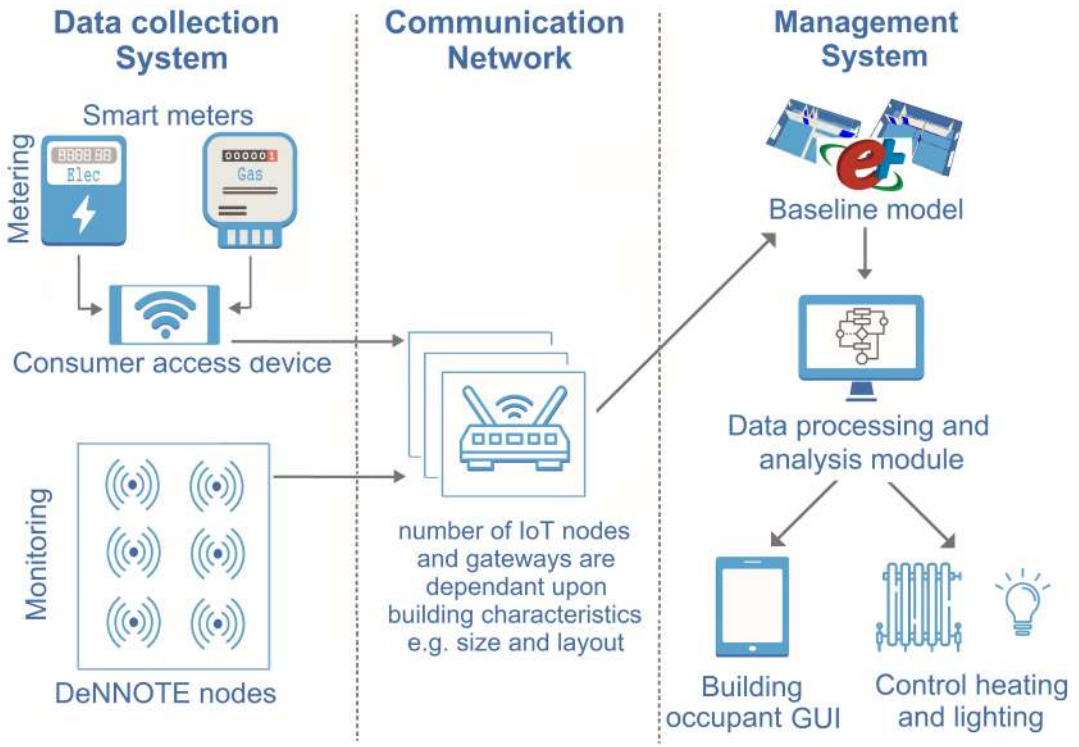


FIGURE 7.1: Graphical representation of the OcCEMS system

The remainder of this chapter includes a system overview in Section 7.2 and the integration of the constituent parts of OcCEMS in Section 7.3. In Section 7.4, the concept of the energy baseline is introduced and methodology to create a baseline is proposed. In Section 7.5 a range of IoT nodes are proposed and in Section 7.6, the deployment of OcCEMS is discussed. The evaluation of the baseline model and the OcCEMS deployment are in Sections 7.7 and 7.8. Finally, the chapter is summarised in Section 7.9.

7.2 OcCEMS Overview

Fig. 7.2 is a flow chart illustrating the OcCEMS energy management process, applying the information it generates regarding the presence of occupants and building conditions to increase the temporal match between the heating and lighting systems. This increased temporal match will reduce the duration that these energy-consuming systems are active, causing a reduction in energy consumption. The energy consumption data which is fed back to the building users will enable further long-term reductions in energy consumption.

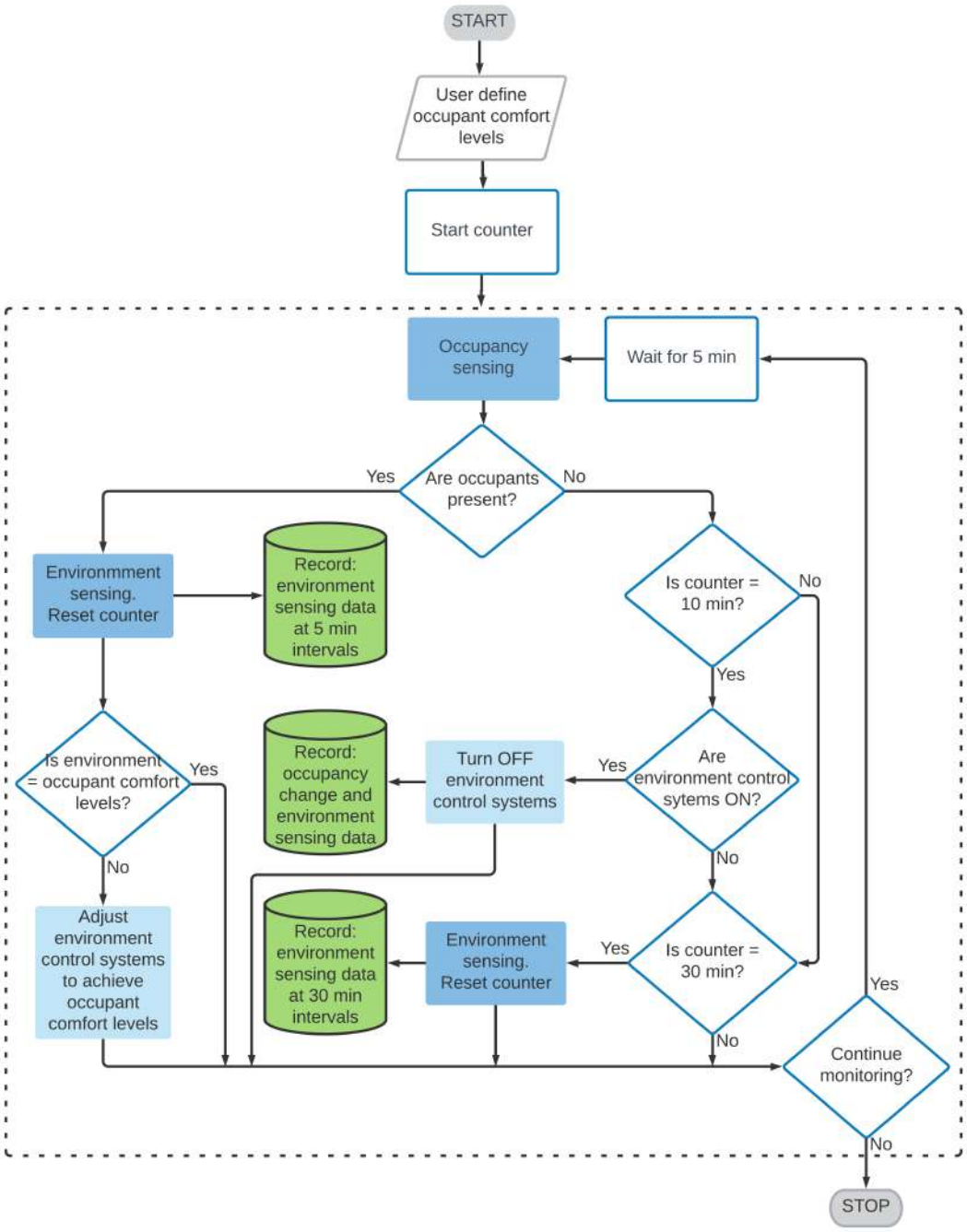


FIGURE 7.2: OcCEMS Process

Learning from existing works [4] [5] and existing systems [81] [83], OcCEMS has been developed to overcome barriers faced by SMEs. These barriers include high costs, lack of expertise and significant time requirements. OcCEMS has been developed to be low cost by combining low-cost sensors so that each node can achieve a large coverage area, integrating a high-performance DNN in each node to enable local data analysis; utilising smart meters which are available free of charge to SMEs; connecting all

components through the SME's preexisting WiFi network to an Open Source EMS integration platform with a user interface (UI). To enable OcCEMS to be quickly and easily installed and managed by non-experts, it is designed as a plug-and-play system which can be installed straight out of the box. The system will be configured and managed through an Open Source EMS integration platform which will be accessed through an intuitive UI. Time requirements are minimised by the inclusion of a baseline model which requires a short monitoring period of 3 months to determine the energy consumption baseline.

To further minimise costs, the system will be hosted locally, isolated from the Internet. Where users want a cloud-based system to enable remote access, this would be available for an additional cost. The cloud-based system will have the necessary security module embedded. The building data will be presented to the building users in real-time in the form of a reporting application which users can view on a computer, tablet or smartphone within the local area network.

7.3 OpenEMS Integration Platform

The pre-existing Open Source platform OpenEMS [160] is used to integrate the various components of OcCEMS. OpenEMS is a platform developed by a number of research institutes and corporations. OpenEMS is available with an Eclipse Public License which allows the platform to be modified and redistributed [161]. It is modular, highly adaptable and compatible with a large number of protocols and existing devices. It also includes a user-friendly UI and an optional cloud-based backend.

OpenEMS comprises three Java-based modules, OpenEMS Edge, OpenEMS UI and OpenEMS Backend. OpenEMS Edge is the local management module which integrates the smart meters, DeNNOTE nodes and the lighting/heating systems. Additionally, it houses the baseline model. This module can enable many additional compatible components to be connected with OcCEMS, including additional sensors, smart sockets, electric vehicle charging point, energy storage systems, renewable energy systems, the energy grid and many more. OpenEMS UI is the front end which enables the user to configure and manage OcCEMS. OpenEMS UI can be accessed locally within the monitored building, or remotely if the

OpenEMS Backend cloud-based module is activated. At this stage, OcCEMS is only configured for local access and the OpenEMS Backend is not utilised.

A mock-up of the homepage of OcCEMS UI is shown in Fig. 7.3. On the homepage, the DeNNOTE nodes, gas and electricity meters, temperature and CO_2 gas level are displayed in the input panels. The other panels include control of the lighting, heating and manual override. On the 'Current Usage' page users can view their recent energy consumption and compare current consumption against the baseline.

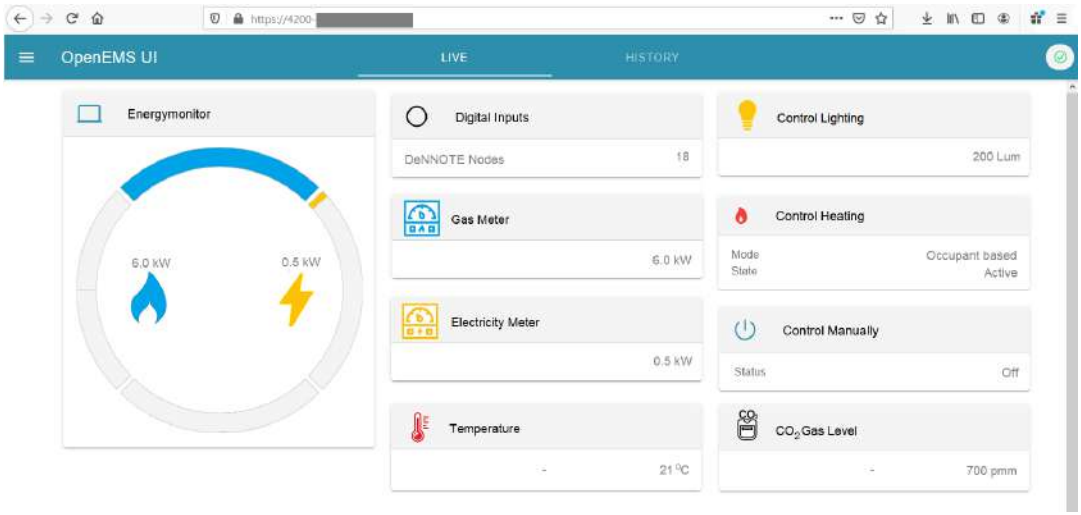


FIGURE 7.3: Mock-up of OcCEMS homepage built on OpenEMS UI

In Section 7.4, the concept of an energy baseline is introduced and a baseline model is proposed.

7.4 Energy Consumption Baseline

For an energy system to indicate a reduction in energy consumption, a baseline is required. The baseline is the measurement of normal energy consumption before energy reduction actions are applied. Traditionally, the creation of a baseline comprises collecting 18+ months of metering and monitoring data. Where possible, the same system should be used to collect baseline energy consumption data and consumption data after this baseline period. The advantage of the traditional method is data completeness since the data are collected over two winter and summer seasons are generated using the same method as the data that it will be compared against. The disadvantage, particularly for SMEs, but also for the environment, is the long lead time before the enterprise can begin to reduce their energy

consumption. Research has highlighted that this significant lead time can cause SMEs to disengage, particularly due to a change in company interests or finances [61], [67].

Instead, the proposed baseline strategy reduces this lead time to 3 months. The strategy is to develop a building model to simulate the energy baseline data. The baseline building model will be incorporated into the management system and will use the metering/monitoring data that is collected over a period of three months in combination with additional indirect building data. Research shows building models can be inaccurate due to significant differences between the building model and the real building [162]. Additional data are used to reduce the differences and improve the performance of the model so it can deliver predictions within 3% of actual [162]. The indirect building data that will be used to improve the model will include:

- Preexisting building energy performance specifications e.g. post-occupancy evaluation (POE) and display energy certification (DEC)
- 12+ months' historical billing data.
- 12+ months' weather data corresponding to the billing period.
- Normal building usage information e.g. operating hours, occupancy levels.

Additionally, during the three month baseline data collection period, it is recommended that an energy consumption profiling survey is carried out. This survey will help the enterprise to understand where and how energy is consumed and to be informed with respect to other relevant energy-saving measures [4].

7.4.1 Baseline Model

Using the EnergyPlus building simulation tool [163], a model has been built to simulate the baseline energy consumption. The model is based upon an existing Manchester-based SME building comprising $130m^2$ across two storeys. The model is shown in Fig. 7.4. The first iteration of the building model, denoted Model1, includes internal and external dimensions, generic building materials, generic lighting, heating system and energy-consuming devices. A pre-defined EnergyPlus 'open office occupancy' schedule is

selected to simulate when the building is occupied, which in turn defines when heating, lighting and electrical devices are switched on. Local weather data is also included.

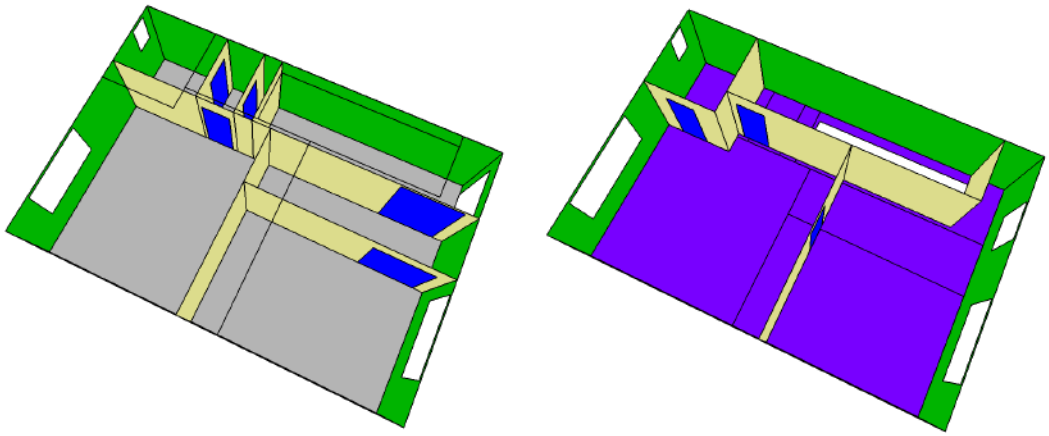


FIGURE 7.4: Ground (left) and first (right) floor baseline building model

Following the plan for the baseline model, the second iteration of the model, denoted Model2, is developed with additional information. This information includes a 2017 DEC certificate with information about building's insulation, heating and environment conditioning. Also, improved glazing, energy-saving light bulbs and information regarding the energy-consuming devices are added, to match the systems in the real building. The third iteration of the model denoted Model3, includes 12 months of billing data and more accurate occupancy, heating and lighting schedules. These schedules highlight the minimal use of the downstairs areas, which includes the operations room and meeting room. Additionally, the heating schedule is changed from the pre-defined 'open office occupancy' schedule to a time-based schedule, such that the whole building is heated between $9am - 5pm$ to meet occupancy comfort levels. Occupancy comfort levels are set to 23°C and 500 lux. EnergyPlus can simulate energy consumption for a range of time periods including daily, monthly and annually. The performance of the model is assessed by comparing existing historical data collected from the period 1st January to 31st December 2019, against the model's predicted baseline energy consumption. Throughout 2019, the actual gas consumption was 6564 kWh and electricity consumption was 6721 kWh.

The baseline model will be evaluated in Section 7.7. In the following Section, a number of IoT nodes with similar specifications are proposed.

7.5 Range of DeNNOTE Nodes

To further reduce system costs, a range of DeNNOTE nodes is proposed. The range is based on node config-3 that was assessed in Section 6.4 and each node has different specifications. Node config-3 will now be referred to as `DeNNOTE1.0` and includes a Raspberry Pi 3B+ processor [149], a $128GB$ SD memory card, $4\times$ HC-SR501 PIRS [144], $2\times$ MLX90640 IRS [146], $1\times$ K30 CO_2 & temperature sensor [148] and $1\times$ SI1132 light sensor (LS) [164]. The SI1132 LS has been added to enable the node to measure the ambient light level. This node has a coverage area of $9.30 \times 8.57m$.

`DeNNOTE0.9` includes all of the same components as `DeNNOTE1.0`, except without the K30 CO_2S. The K30 CO_2S has a coverage range of approximately $116m^2$ [154]. It is unnecessary to deploy multiple K30 sensors within an area smaller than this. This node is ideal for deployment alongside the `DeNNOTE1.0`.

`DeNNOTE0.5` is similar to `DeNNOTE1.0`, but it includes less PIRS and IRS and achieves a coverage area just under half that of `DeNNOTE1.0`, i.e. $4.20 \times 8.57m$. This node is ideal for monitoring smaller spaces.

`DeNNOTE0.4`, is similar to `DeNNOTE0.5`, but without the CO_2S. This node is ideal for deployment alongside the `DeNNOTE1.0`, e.g. around the perimeter of the room to reduce the coverage overlap that would occur if multiple `DeNNOTE1.0` or `DeNNOTE0.9` nodes were deployed instead.

`DeNNOTE0.1` is a minimal version of the node and does not include AI. It includes just the Raspberry Pi 0 W, $8GB$ memory card, $4\times$ PIRS and $1\times$ LS. This node has been developed for areas such as toilet facilities which are accessed for very short periods of time. Since it does not include the DNN, it determines if occupants are present based on detecting their movement. When this node detects an occupant, it will automate turning the light on for 5 minutes. This time period can be modified through the OcCEMS UI.

Table 7.1 states the cost of each component in the DeNNOTE nodes. These costs are based on purchasing small quantities of each component. If the system is manufactured on a large scale, the component costs will be substantially reduced. Table 7.2 details a summary of each node's specifications and cost.

In the next section, the deployment of the OcCEMS system is discussed.

TABLE 7.1: Cost of DeNNOTE node components

Component	Function	Cost
Raspberry Pi 3B+	Processor	£33 [165]
Raspberry Pi 0W	Processor	£12 [166]
128GB SD Memory card	Memory	£14 [165]
8GB SD Memory card	Memory	£5.10 [165]
83-17540	Raspberry Pi 3 case	£3.66 [25]
MC-RP001	Raspberry Pi 0 case	£1.66 [25]
MLX90640	IRS	£24 [165]
K30	CO_2 and temperature sensor	£64 [167]
HC-SR501	PIRS	£0.42 [168]
SI1132	LS	£1.46 [165]

TABLE 7.2: DeNNOTE node specifications and cost

Node	Specifications / sensors	Cost
`DeNNOTE1.0`	Coverage area: $9.30 \times 8.57m$ Monitors: Occupants, CO_2, temperature, light Sensors: 4× PIRS, 2× IRS, 1× CO_2S, 1× LS	£165.80
`DeNNOTE0.9`	Coverage area: $9.30 \times 8.57m$ Monitors: Occupants, light Sensors: 4× PIRS, 2× IRS, 1× LS	£101.80
`DeNNOTE0.5`	Coverage area: $4.20 \times 8.57m$ Monitors: Occupants, CO_2, temperature, light Sensors: 3× PIRS, 1× IRS, 1× CO_2S, 1× LS	£141.38
`DeNNOTE0.4`	Coverage area: $4.20 \times 8.57m$ Monitors: Occupants, light Sensors: 3× PIRS, 1× IRS, 1× LS	£77.80
`DeNNOTE0.1`	Coverage area: $9.30 \times 9.30m$ Monitors: Occupants, light Sensors: 4× PIRS, 1× LS	£20.64

7.6 Deployment of OcCEMS

Due to Covid-19 restrictions throughout 2020, the SME that has been participating in this research has not been open for normal operation. This means it is not possible to deploy OcCEMS in this building. Instead, the

MIoTs simulation tool, detailed in Chapter 3, is used to simulate the deployment of OcCEMS in this same building. The simulation is based on the sensor grid WSN deployment technique and the `DeNNOTE1.0` node. MIoTs is used to simulate the upstairs and downstairs of the SME office, based on working hours $9am - 5pm$, a maximum of 6 occupants and natural ventilation. The sensor data that is simulated for each node is processed by the single-mode DNN, DNN-S2e, to create an occupancy schedule for each region inside the building.

Next, the EnergyPlus Energy Management System (EnergyPlus-EMS) is used to create sensor and actuator-based control modules [169]. A new EnergyPlus model denoted OcCEMS-EP, is created by installing the sensor and actuator control modules in the existing EnergyPlus model of the SME (Model3). Using EnergyPlus-EMS, EnergyPlus is able to simulate when and where the occupants are in the building model. When occupants are present in any part of the building, the OcCEMS-EP model will simulate heating the whole building to meet occupant comfort levels of 23°C. When there are no occupants, the model will turn the heating off. In comparison to the heating, the management of the lights is based on the occupancy of each room. When one or more occupants are present in a room, the OcCEMS-EP model will turn the lights on in the respective room to meet comfort levels of 500 lux. When there are no occupants in a room, the OcCEMS-EP model will turn the lights off. The 2019 local weather data is part of this model. The OcCEMS-EP EnergyPlus model is used to simulate the energy consumption for a period of 1 year.

In Section 7.8, this deployment will be evaluated. In the next section, the baseline model that was proposed in Section 7.4 is evaluated.

7.7 Evaluation of Baseline Model

The baseline models that were developed in Section 7.4 are assessed by comparing their simulated energy consumption against the actual 2019 energy consumption, based on monthly utility bills. In Fig. 7.5 and 7.6 the actual and simulated baseline energy consumption are shown.

The simulated energy consumption of Model1 varies significantly from the actual energy consumption of the building. The annual simulated gas consumption is 49% lower than the actual gas consumption and follows an inverted bell curve, with some similarities to the actual month-on-month

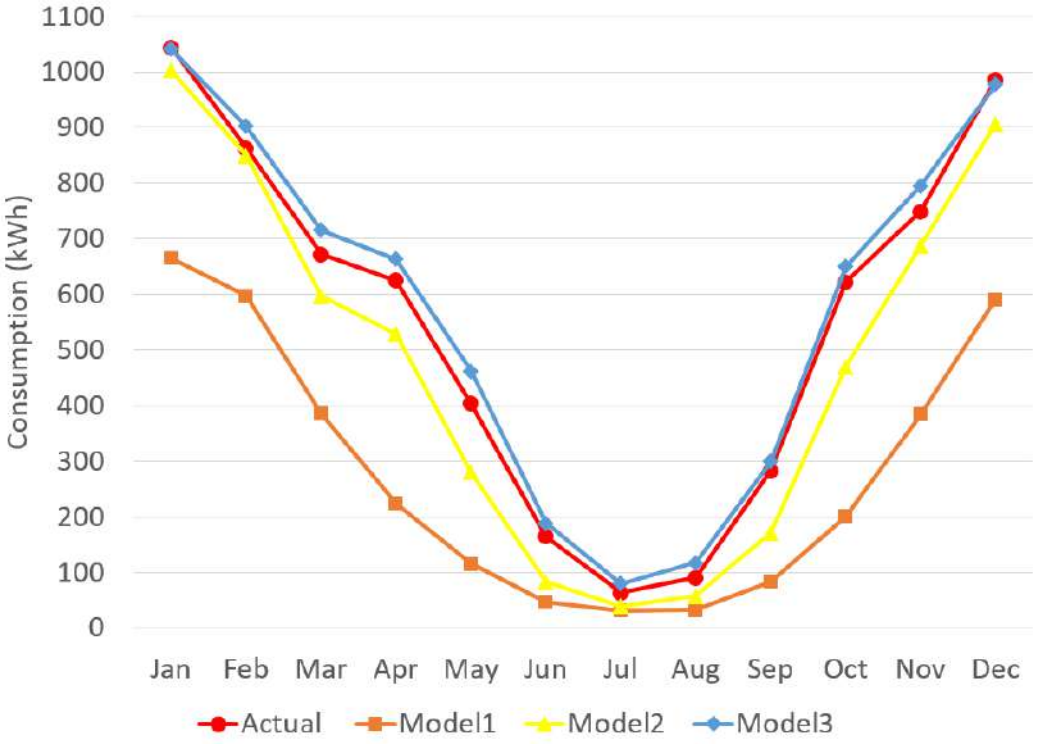


FIGURE 7.5: Gas consumption simulated by 3 iterations of the energy consumption baseline model

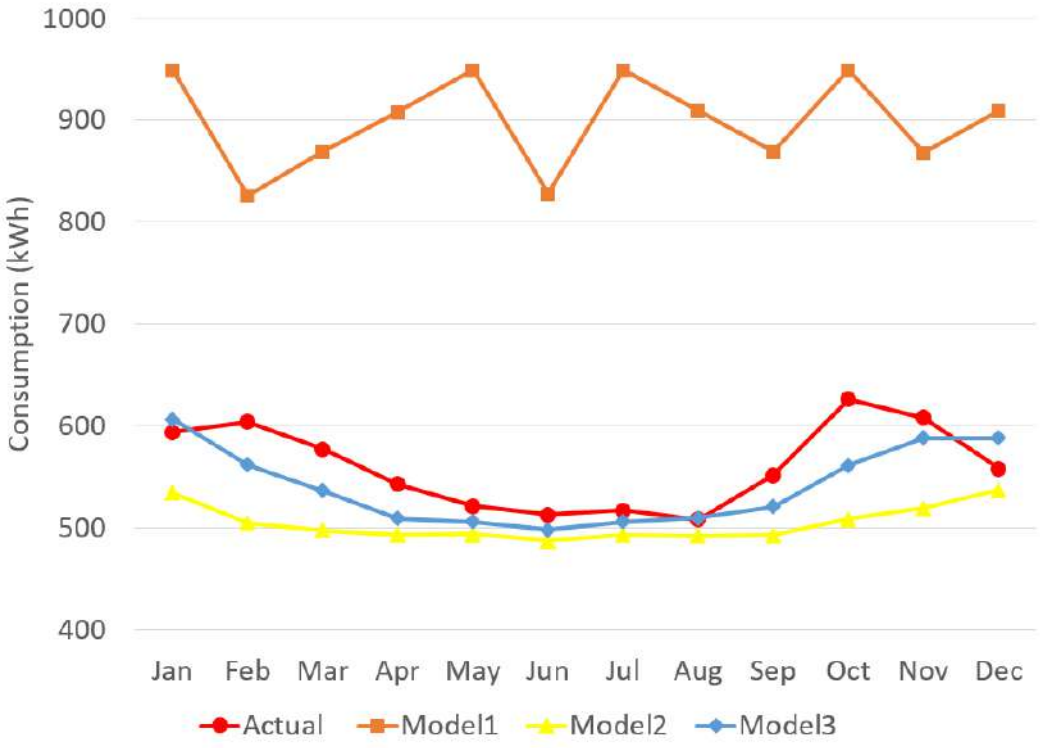


FIGURE 7.6: Electricity consumption simulated by 3 iterations of the energy consumption baseline model

consumption levels. In comparison, the electricity consumption is 60% higher than the actual electricity consumption and does not correlate with the actual, showing significant and random fluctuations in its monthly consumption.

The next iteration of the model, Model2, demonstrates an improvement compared to Model1. The total simulated gas is reduced to 5671 kWh, which is 14% less than actual and correlates much more closely with the month-on-month trends in the actual consumption. The simulated electricity consumption is reduced to 6054.31 kWh, 10% lower than actual. The electricity no longer models significant month-on-month fluctuations, but the consumption also does not correlate with the actual consumption. Instead, the simulated electricity consumption is almost constant across the year, with a slight reduction over the summer months.

Model3 shows a significant improvement in reflecting the actual gas and electricity consumption and month-on-month trends. It achieves annual gas consumption of 6889 kWh, 5% above actual and closely following the trends in the actual consumption. The simulated electricity consumption is 6492 kWh, 4% below actual and includes some similarity with the actual month-on-month consumption. These improvements in the model result from the changes in the occupancy schedule and the reduction in utilisation of the downstairs areas. Additional granular data captured hourly by the SMET2 electricity and gas meters and minutely by the IoT nodes will enable this model to demonstrate further improvement.

EnergyPlus Model3 will be the baseline model that will be used in further analysis of the OcCEMS system. In Section 7.8, a deployment of OcCEMS will be simulated and its performance will be assessed.

7.8 Evaluation of OcCEMS Deployment

It Section 7.6, it was proposed that deployment of the OcCEMS system should be simulated by the MIoTs tool. Figs. 7.7 and 7.8 show the mapping of the OcCEMS deployment within the Manchester-based SME[1]. In both the downstairs and upstairs floor plans, there are regions which do not have sensor coverage. This is intentional, two of the regions are storage, one on each floor. The third region, shown in grey in the first floor deployment, is

[1]It should be noted, the sensor symbols have been added manually to the simulated deployment images to illustrate the nodes' positions

the stairs. The stairs region is common to both the ground floor and first floor. For this region, the sensors are illustrated on the ground floor plan. Across both of the floors, a total of 8 DeNNOTE nodes are deployed, 5 on the ground floor and 3 on the first floor.

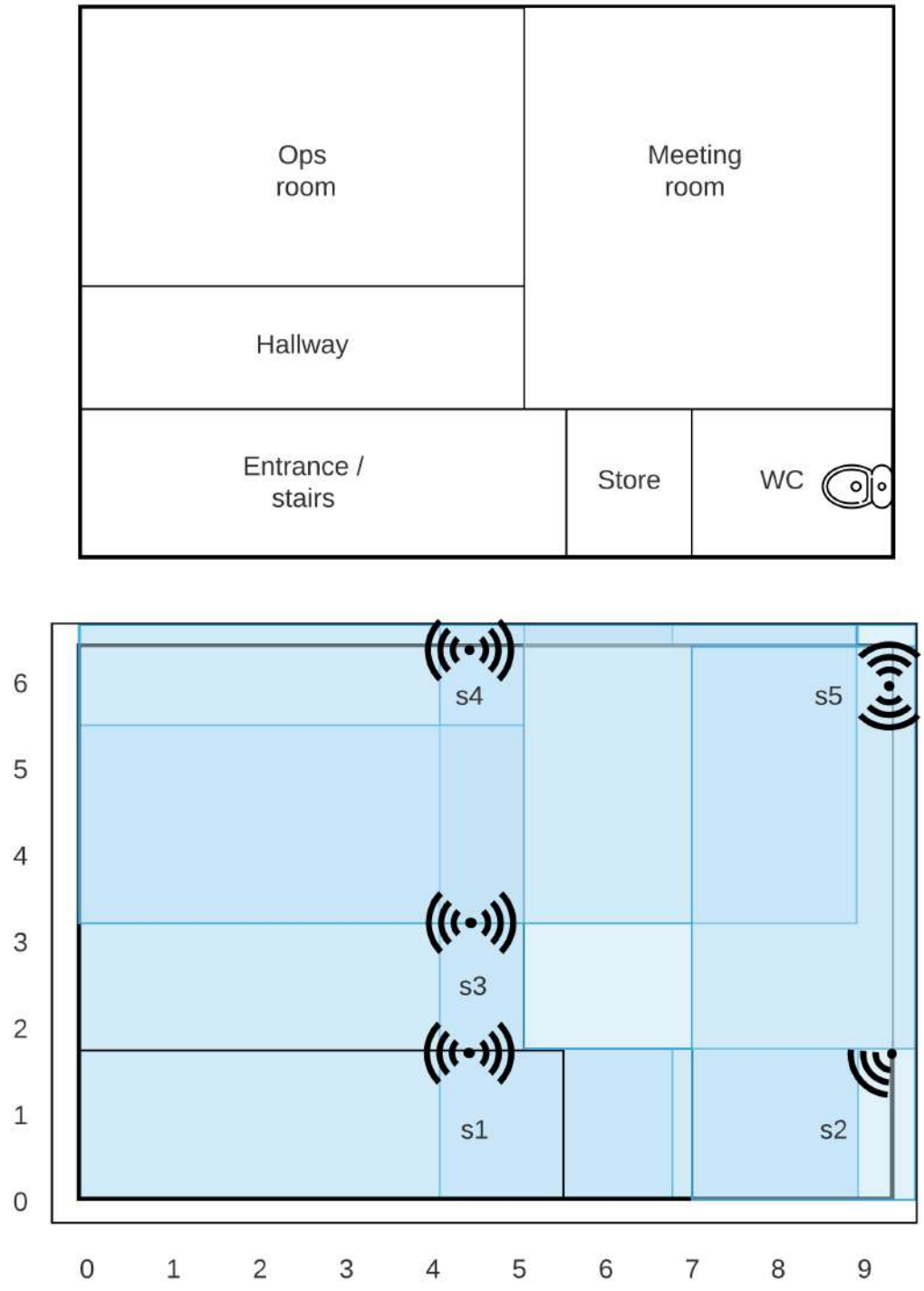


FIGURE 7.7: Floor map and simulated OcCEMS deployment in ground floor of Manchester-based SME

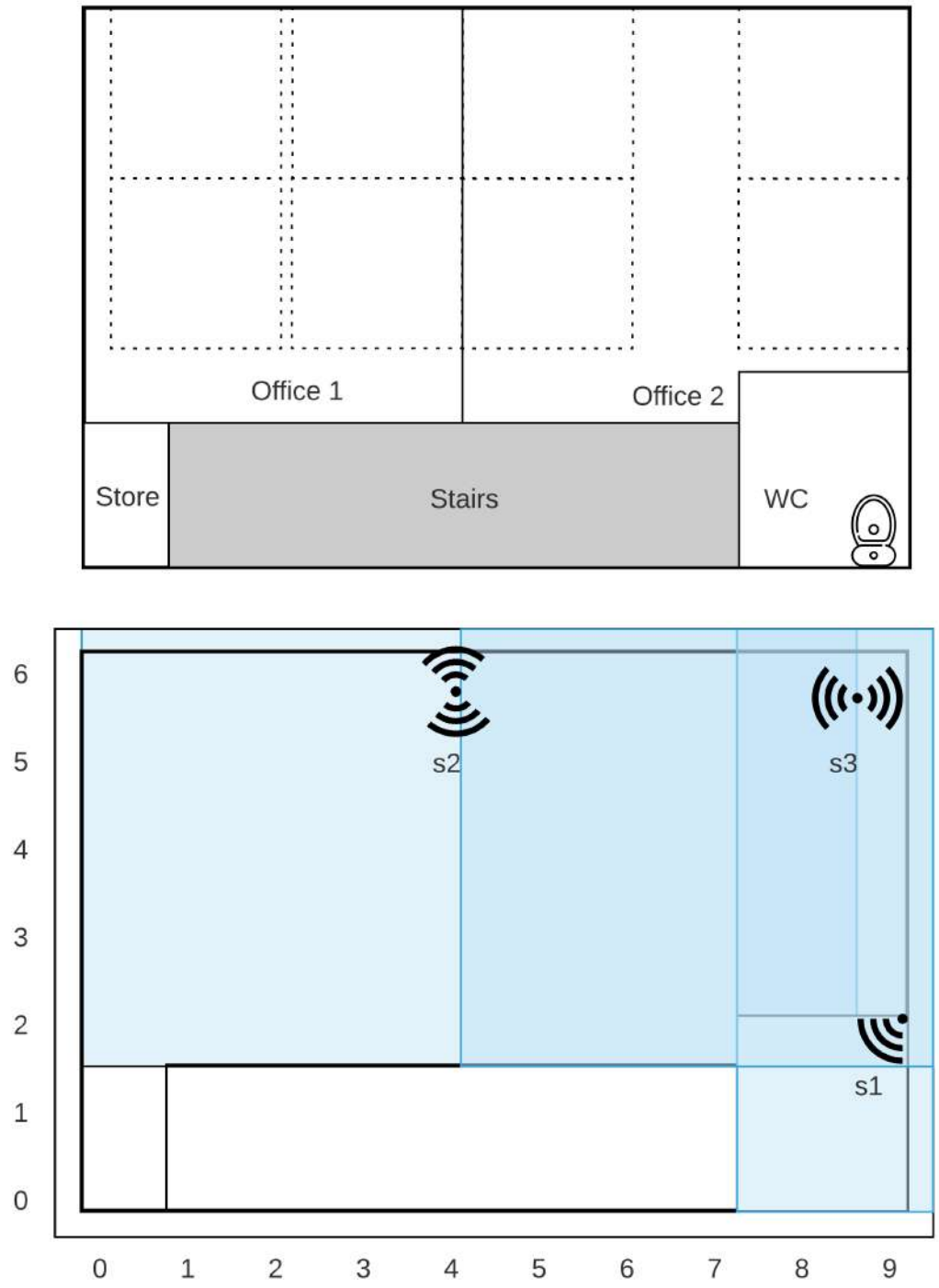


FIGURE 7.8: Floor map and simulated OcCEMS deployment in first floor of Manchester-based SME

7.8.1 Evaluation of OcCEMS Hardware Cost

In the MIoTs simulation, the IoT nodes are all the `DeNNOTE1.0 node`. The total hardware cost to deploy `DeNNOTE1.0` throughout the building is £1326.40.

For a real deployment in this building, to further reduce system costs and network redundancy, a selection of nodes from the DeNNOTE range would be used. The full range is defined in Section 7.6. Both Tables 7.3 and 7.4 summarise which nodes would be deployed where on the ground floor and the first floor respectively. The deployment would include a single `DeNNOTE1.0` in each room and `DeNNOTE0.1` in the toilet areas. Based on using these specific DeNNOTE nodes, the total hardware cost would be reduced to £1036.08, which is 22% less than using `DeNNOTE1.0` throughout. For further assessment of OcCEMS, only `DeNNOTE0.1` is used in the simulations.

TABLE 7.3: Key of DeNNOTE nodes for a real-life deployment on the ground floor

Node Configuration	Deployed node label	Total cost
`DeNNOTE1.0`	s1, s3, s4, s5	£663.20
`DeNNOTE0.1`	s2	£20.64

TABLE 7.4: Key of DeNNOTE nodes for a real-life deployment on the first floor

Node Configuration	Deployed node label	Total cost
`DeNNOTE1.0`	s2, s3	£331.60
`DeNNOTE0.1`	s1	£20.64

7.8.2 Evaluation of OcCEMS Simulated Performance

EnergyPlus has been used to simulate the annual energy consumption of a Manchester-based SME with the OcCEMS system installed. The building occupancy levels used by the EnergyPlus model were generated from the OcCEMS sensor data that was simulated using the MIoTs simulation tool. The actual 2019, baseline and OcCEMS energy consumption are compared in Figs 7.9 and 7.10 for gas and electricity respectively.

Both the gas and electricity consumption show a significant reduction over the period of a year. In comparison to the actual energy consumption in 2019, the simulated gas consumption is reduced by 10.4% and the electricity consumption is reduced by 10.6%. The trends in the OcCEMS gas consumption data are very similar to the trends in the actual and the baseline energy consumption data. This shows that with OcCEMS installed, the heating system continues to be used in a very similar way, but for

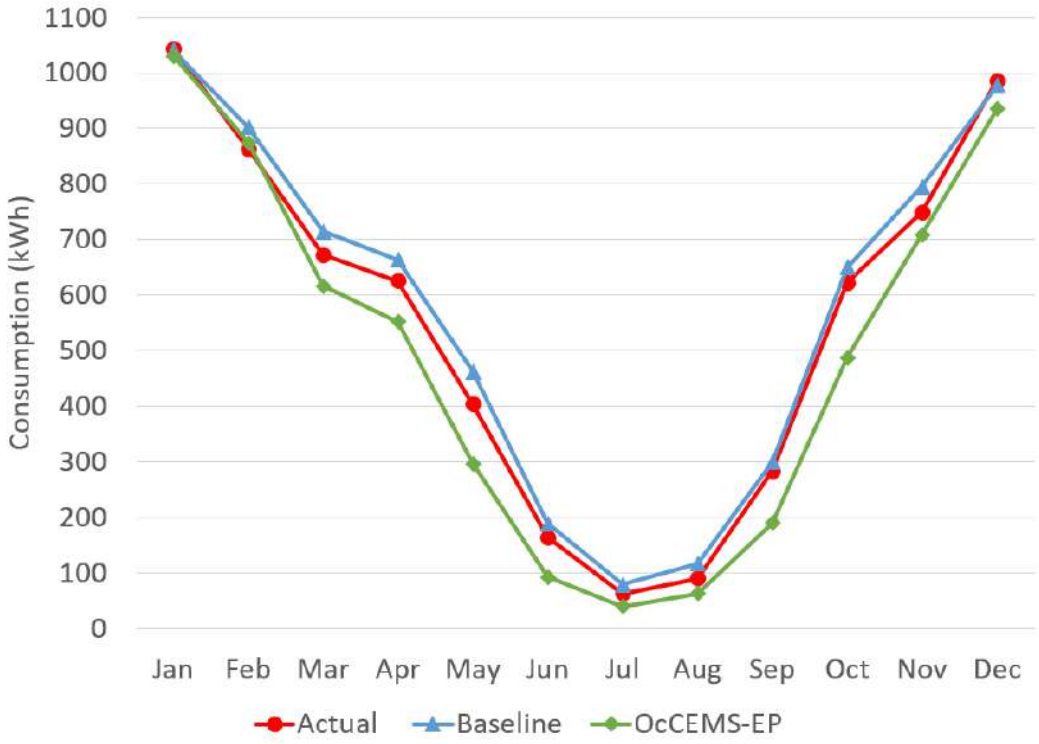


FIGURE 7.9: The actual gas consumption compared to the simulated gas consumption of Model3 and OcCEMS

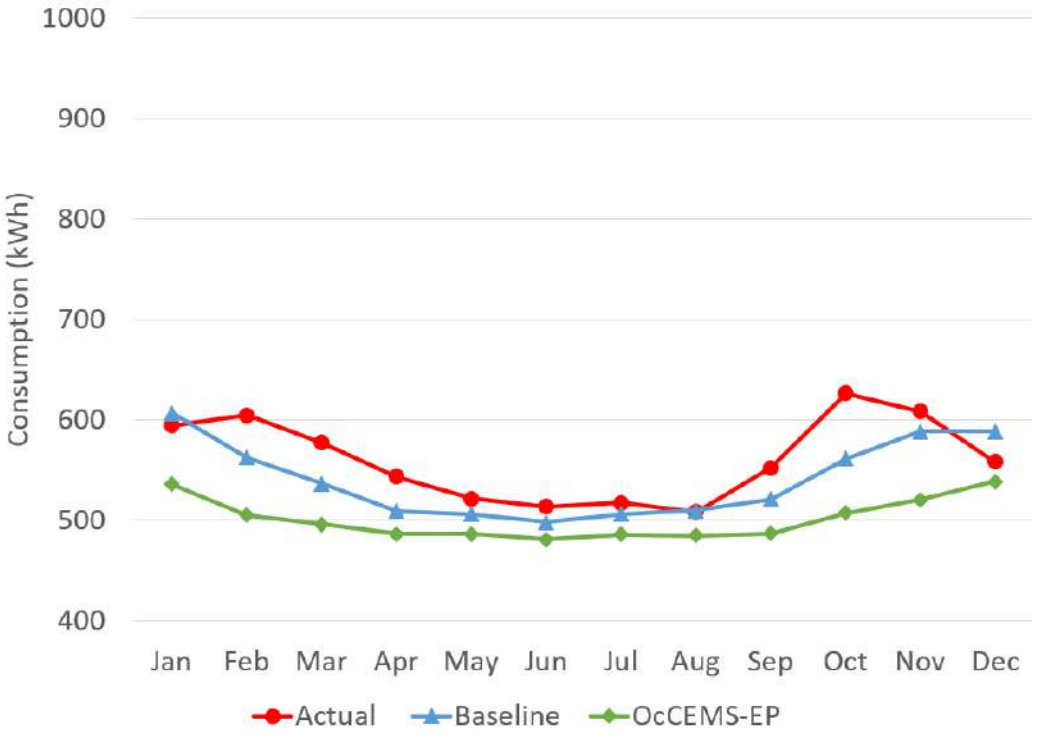


FIGURE 7.10: The actual electricity consumption compared to simulated electricity consumption of Model3 and OcCEMS

shorter periods. In comparison, the OcCEMS electricity consumption data does not follow the same trend as the actual energy consumption data. Instead, it shows a more significant reduction of up to 20% in the winter and Autumn months, particularly February, March and October. There is also a reduction across the summer months, but it is closer to 5% in May to August. These different levels of reductions are a result of two things, firstly a closer temporal match between occupants and light usage throughout the year and secondly the monitoring of ambient light levels which has more impact on the energy reduction in February, March and October. The result of monitoring ambient light levels is when the natural light level meets the occupant comfort level, OcCEMS will automatically turn the lights off. Throughout winter, due to darker mornings, building users generally turn the lights on upon arrival to work. But they may unaware of the natural light level increasing throughout the day [170]. As such, during winter months, OcCEMS enables a more significant reduction in electricity consumption by turning the lights off when the natural light level meets the occupant comfort level.

A real-life deployment of OcCEMS can achieve this energy reduction. In addition to automating switching off the heating and lighting systems, the system will use the smart metering data to generate real-time consumption data which will be fed back to building occupants. This feedback will enable further energy reductions to be achieved and maintained for the long term by encouraging positive behaviour changes and without any further cost to the SME [23]. Using the baseline and current data consumption, building occupants can trial other low-cost energy-saving measures to assess if they can further reduce their energy consumption, for example, reducing the occupant comfort level down from 23°C to 21°C, changing to energy-saving bulbs, or using smart sockets.

7.9 Chapter Summary

An occupant-centric IoT-based EMS, OcCEMS, was proposed using the contributions of earlier chapters. The OcCEMS components were combined using the pre-existing OpenEMS integration platform and a WiFi network. A baseline model was created using indirect building data including energy bills, building construction materials and approximate occupancy schedules. The baseline model will be integrated into the OcCEMS system

to enable current energy consumption to be compared to the baseline. This comparison will enable building users to assess whether there is a reduction in energy consumption. The baseline model was also used in the evaluation of OcCEMS. A range of DeNNOTE nodes with different specifications were proposed. This range of nodes enabled a further reduction in the system cost by not over-specifying the system's hardware.

The MIoTs tool was used to simulate a deployment of OcCEMS in a real SME building. The hardware cost for the OcCEMS system is £1036.08, which is 16% of the cost of comparable existing commercial systems, which could cost between £2600 and £6500 for a building spanning $130m^2$ [79], [80]. This cost is based on achieving strong coverage with no coverage gaps and minimal network redundancy. The MIoTs simulation data and the EnergyPlus baseline building model were combined to model how OcCEMS would reduce energy consumption in an SME. The simulations demonstrated that OcCEMS increased the temporal match between occupants and energy-consuming systems. It also demonstrated that OcCEMS monitored occupant comfort levels more accurately that occupants, such that when occupants were present, it correctly switched off the lights when the natural light level was sufficient. In summary, OcCEMS demonstrated a reduction of 10.4% in annual gas consumption and a reduction of 10.6% in annual electricity consumption. OcCEMS also enables further long term energy reductions through the feedback of real-time energy consumption data to building users.

Chapter 8

Conclusion and Recommendations

This chapter concludes the thesis, presents the technical limitations and outlines possible areas of future research activities.

8.1 Conclusion

Chapter 2 considered the motivation for this work, that worldwide energy consumption is increasing, causing an increase in GHG emissions and damaging the environment. Significant efforts are being made across most nations to reduce this impact, but assessments of the UK's progress are damning. Small commercial buildings are responsible for 7.1% of the UK's energy consumption and research has shown that low-cost energy-saving measures could reduce this by 25%. But the SMEs that are largely responsible for these small commercial buildings are experiencing significant barriers when trying to reduce their energy consumption, i.e. high costs, lack of expertise and time constraints. IoT was introduced in Chapter 1 as a low cost, versatile and ideal technology for building monitoring and three enabling technologies (WSN, ML and simulation tools) were introduced and the existing work was reviewed.

A number of existing WSN deployment techniques were assessed. For each technique, algorithms were developed to determine the sensor density and space coverage. The deployments were simulated and assessed in terms of their ability to accurately capture building monitoring data. It was shown that for a range of different environments, the sensor grid deployment required the minimal number of sensors and achieved a good level of data capture accuracy, demonstrating 81.20% accuracy in occupancy monitoring based on utilising the blob detection function. This deployment did create sensor coverage gaps, resulting in 21.80% of the building to be

unmonitored. The hardware in the IoT node was optimised by increasing the quantity of each multimodal sensor and using concatenation-based data fusion techniques to combine the data. This enabled a significant increase in node coverage range and reduced the coverage gaps when the nodes were deployed in a sensor grid WSN. Based on the sensor grid deployment technique and a constant number of IoT nodes, the building coverage was increased from around 78.54% to 100% by increasing the number of sensors in each node and varying their configuration.

Building upon the IoT-based WSN, machine learning was utilised to improve its performance. A deep learning, feed-forward DNN architecture was developed and integrated into the IoT node. The DNN utilised IR data that was captured by the IoT node and was processed locally to determine whether occupants were present. The inclusion of the DNN improved the performance of the IoT node from 81.20% based on a WSN deployed in sensor grid and utilising blob detection to determine occupancy, to 98.63% based on the same WSN deployment technique, but using the single mode DNN DNN-S2e and binary classification to determine occupancy. The hyperparameters of the DNN model were studied to determine the optimal hyperparameters to create a model with a high generalisation ability; demonstrating binary classification performance of $80.56\% - 98.63\%$ across three different environments, two of which were unseen environments, different from the training environment.

In an effort to improve the generalisation ability of the DNN model by improving the sensitivity of the DNN model on unseen data in a more challenging environment where the ambient temperature and thermal occupancy sensor data were too similar, the DNN architecture was expanded. The DNN was already achieving a high level of accuracy in office-based environments, so the focus of this development was on environments where ambient room temperature was slightly higher and the performance of the DNN was reduced. Such environments could include a server room or kitchen. As such, the architecture of the DNN was expanded to enable it to process multimodal sensor inputs. Based on a smaller multimodal train and test data set, a SM and two MM DNNs were evaluated. The MM DNNs both demonstrated comparable behaviour to the SM DNN in the office-based environments, but a significant improvement in the environment with a higher ambient temperature due to the inclusion of CO_2, PIR and temperature data in the DNN model. When the MM DNN

is trained with a larger multimodal data set, the generalisation ability of the MM DNN will be improved compared to the SM DNN.

All of these technical contributions were brought together to deliver a real-life application that can be deployed within a small commercial building to reduce energy consumption. This application is a non-intrusive, occupant-centric IoT-based EMS. All of the components were integrated using a pre-existing energy management application OpenEMS. The proposed system integrated smart meters, IoT nodes, machine learning and control of building heating and lighting systems. The building occupants are provided with an intuitive UI to manage the system and monitor real-time energy usage. The EMS is ISO50001 compliant since a baseline is integrated, enabling building occupants to view their real-time consumption and compare it with the baseline consumption. The IoT node was further developed to create a range of nodes with different specifications. The EMS was simulated within a real SME building and the cost of the system to be determined. The building comprised 2 floors, 8 separate rooms and a total floor area of $130m^2$. The total hardware cost of the EMS was £1036.80 which very low for such a high performance system, deployed within a building of this size. The deployment included 8 IoT nodes, connectivity with a smart gas and electricity meter and a user application for system configuration and management. The simulation demonstrated that the EMS will automate a significant reduction of energy consumption, achieving 10.5% reduction compared to the actual energy consumption.

In summary, this work has utilised IoT, WSN and machine towards resolving the barriers faced by SMEs when trying to reduce their energy consumption. The presented system is low cost, demonstrates a high performance regarding occupancy and building monitoring, increases the temporal match between building occupancy and energy consuming systems, maintains occupant comfort levels and automates a 10.5% energy reduction.

8.2 Limitations of Work

The most significant limitation of this work is that the final system was not deployed in a real environment to evaluate its ability to reduce energy consumption. This was mainly due to the current Covid-19 restrictions which have significantly effected all UK enterprises. As part of these

restrictions, enterprises had to enable employees to work from home where possible [122]. This meant for the majority of 2020, occupancy levels within many commercial building was significantly effected. To reduce the impact of this limitation, the single DeNNOTE node was deployed and evaluated in a number of real environments. Additionally, each stage of the work was modelled using either EnergyPlus [163] or MIoTs[1] simulation tools.

8.3 Further Work

When the current Covid-19 restrictions are eased and the utilisation of small commercial buildings returns to normal, further work could include the deployment of the OcCEMS system in a number of small commercial buildings. These deployments would demonstrate whether OcCEMS can improve upon its simulated performance. Further developments could include using machine to evaluate occupancy patterns to predict occupancy usage schedules [171] to further reduce energy consumption by detecting device faults and automating turn-off of idle devices such as computers, IT peripherals, industrial machinery, etc.

Further work could be carried out to capture a larger multimodal train and test data set. This will enable the two MM DNNs to be evaluated against the optimal SM DNN. Additionally, optimisation of the MM DNN hyperparameters should be studied. The hyperparameters that were used were optimal for an image-based SM DNN. In comparison, the MM DNN also processes numeric data so its hyperparameters, particularly the Sigmoid activation functions, may no longer optimal. This analysis will enable improvement of the model's generalisation ability delivering better performance across a wider range of environments.

The MIoTs simulation tool is relatively limited in its current modelling capabilities. This includes 2D building modelling, a limited range of sensor models, no consideration of network protocols and no graphical user interface (GUI). Further work could be carried out to develop this tool to enable 3D building modelling and more sensor models: e.g., temperature; light; humidity; and sound. The addition communication protocols would significantly improve the scope of the tool. The addition of a GUI would make the tool accessible to other users.

[1] Developed as part of this research and described in the Chapter 3

Appendix A

Appendix

A.1 Published Work

A.1.1 Journal Paper

Elsevier Sustainable Cities and Society Journal, Vol. 54, pp.101728-101743

Internet of Things: Evolution and Technologies from a Security Perspective

Abstract

In recent years, IoT has developed into many areas of life including smart homes, smart cities, agriculture, offices, and workplaces. Everyday physical items such as lights, locks and industrial machineries can now be part of the IoT ecosystem. IoT has redefined the management of critical and non-critical systems with the aim of making our lives more safe, efficient and comfortable. As a result, IoT technology is having a huge positive impact on our lives. However, in addition to these positives, IoT systems have also attracted negative attention from malicious users who aim to infiltrate weaknesses within IoT systems for their own gain, referred to as cyber security attacks. By creating an introduction to IoT, this paper seeks to highlight IoT cyber security vulnerabilities and mitigation techniques to the reader.

The paper is suitable for developers, practitioners, and academics, particularly from fields such as computer networking, information or communication technology or electronics. The paper begins by introducing IoT as the culmination of two hundred years of evolution within communication technologies. Around 2014, IoT reached consumers, early products were mostly small closed IoT networks, followed by large networks such as smart cities, and continuing to evolve into Next Generation Internet; internet systems which incorporate human values. Following this evolutionary introduction, IoT architectures are compared and some of the technologies that are part of each architectural layer are introduced. Security threats within each architectural layer and some mitigation strategies are discussed, finally, the paper concludes with some future developments.

1. Introduction

The Internet of Things (IoT) is a network of everyday things, connected together through the Internet. The function of an IoT system is to monitor the world around itself, to enable and assist, or to automate a response to changes in the system's environment [118, 25]. In comparison, the purpose of an IoT system is to improve the quality of life by enabling the best response to an environmental change [79] by

providing responsive services which are specific to the end-users' needs [88]. An IoT device can be 'any thing' in the world that includes the technological components to enable the Thing to connect to the Internet through a wired or wireless network. IoT users can be a human, or machine, or a combination [20]. IoT is not a specific device or technology, instead, IoT is the inter-working of different technologies enabling the connectivity of many Things.

Generally, IoT networks comprise of many connected Things connected together through a management platform. The platform has a number of roles including managing the connected Things, system threats and security, data analysis, processing and storage, and managing the response of any Things [46]. IoT Things can either have all of their electronic components included in them at conception, or added later. Examples of systems where smart functionality is added after conception [81] include a pet with a tracking tag, external or implanted human biometric systems, or older high value legacy vehicles such as an aircraft. Smart conceptualised systems include smart home heating and self-driving vehicles.

2. Evolution of ICT Culminating in IoT

Figure 1 is a time-line showing the evolution of Information Communication Technologies (ICT) starting from the 1830s, highlighting some of developments and culminating in IoT. The telegraph is considered as the first major invention of wireless communication technology. Following this hugely significant invention, comes the creation of the telephone, closely followed by the birth of computers. The invention of computers in the 1920s enabled the solving of complex computations, including the breaking of previously unbreakable codes and calculations including code breaking at Bletchley Park during WWII. The architecture of this early machine became the foundation for the computing theory that followed [92]. This led on to the development of the Personal Computer (PC) in the 1970s, and their unprecedented uptake in the 1980s. The PC totally revolutionised the lives of individuals in the home and workplace due its reduction in size and cost, and the addition of new software such as word processing and spreadsheet tools [75]. Computer technology has developed relatively slowly over 90 years to the computers we recognise today, from large powerful servers, to PCs and more versatile mobile computers, including the laptop, tablet and smart phone.

The next development along the time-line was the networking of computers, including US Defence Department project ARPNET, through to the WWW, developed by Tim Bernes-Lee and launched in 1991 as a tool to share documents. The combination of the Internet and WWW are the most significant milestone within the IoT

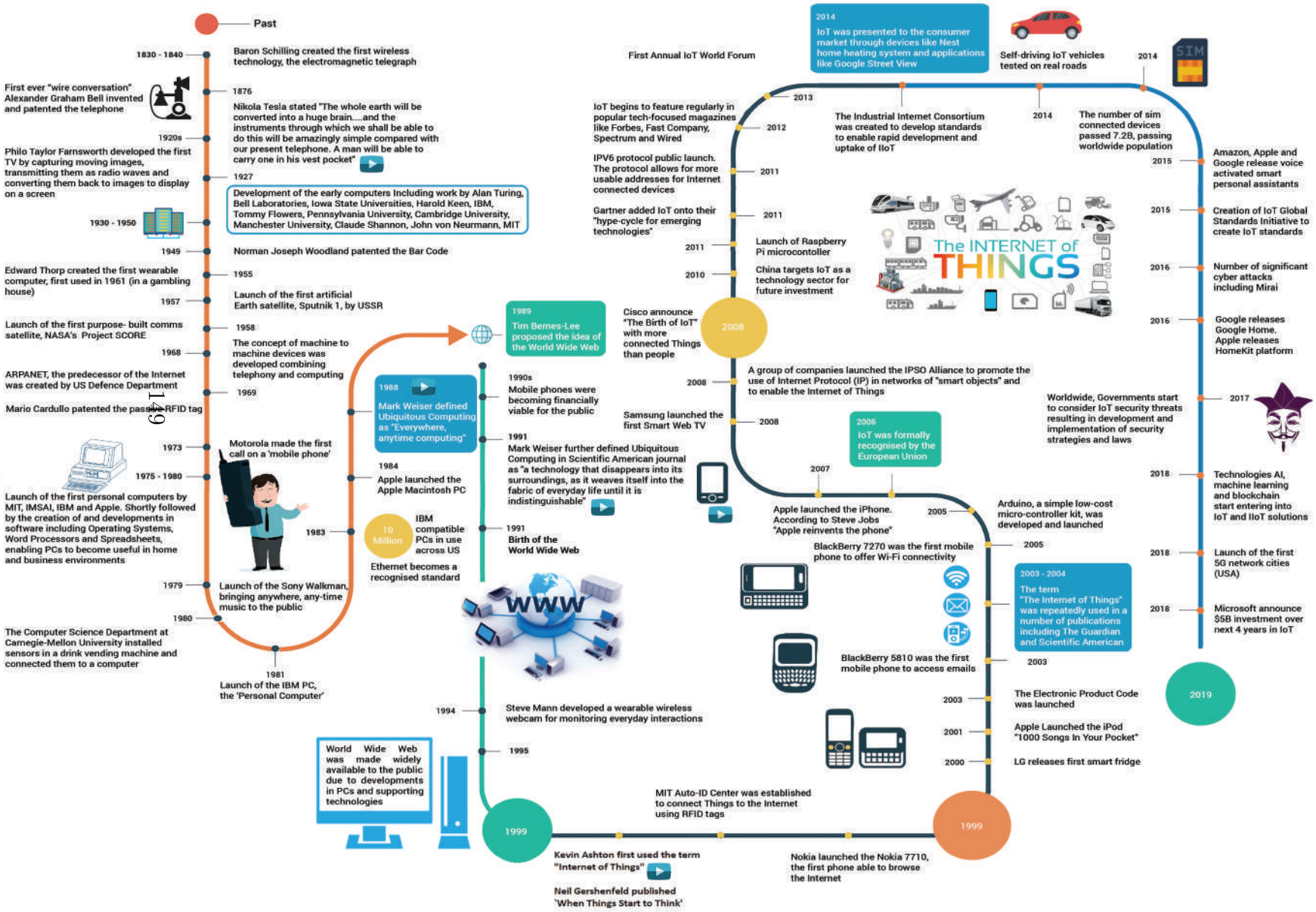


Figure 1: Time-line: The Evolution of ICT Culminating in IoT

time-line. Similar to computers, the Internet has developed relatively slowly, over 50 years from closed connectivity projects to the powerful tools that we all know and use today.

The release of the Raspberry Pi microcontroller in 2011 was another major IoT breakthrough and was in-part responsible for the swift uptake of IoT technology. This low-cost, versatile microcontroller suddenly opened up IoT to hobbyists and end users. Other microcontrollers did exist, but due its low cost, low complexity and relatively large processing power and significant amount of free on-line support, including tutorials, videos, educational DIY websites, blogs and forums, IoT was no longer limited to commercial projects. The time-line continued onto 2014 when IoT technology was widely presented to the consumer market by companies like Google, as they acquired Nest, Apple introducing the Apple Watch, and Siemens as they introduced SmartThings, an affordable smart home starter kit and software platform.

Since this consumer introduction, IoT has continued to develop significantly, including the creation of more than 400 IoT platforms, many commercial IoT developers and thousands of products. In 2016, large tech companies including Amazon, Apple and Google have release voice activated personal assistants. In addition to consumer IoT, there is Industrial IoT (IIoT) enabling the automation of many industrial processes. The concept of IoT has also continued to evolve; initially IoT systems comprised of lots of small closed networks, but this concept has evolved to incorporate larger more connected networks, for example smart cities with smart transport infrastructures. But large infrastructure is not the end of this evolutionary journey, the concept of IoT is currently evolving into the Next Generation Internet (NGI). NGI is the vision of IoT which seeks to encompass human values within Internet based systems, enabling "human potential, mobility and creativity at the largest possible scale while dealing responsibly with our natural resources... we shape a value-centric, human and inclusive Internet for all" [60]. These NGI concepts are being integrated into existing and new IoT systems through the inclusion of advanced technologies such as artificial intelligence, machine learning, augmented reality and virtual reality, are others, whilst "making the future internet more human-centric". In many areas, NGI concepts match the smart sustainable city concept, the main difference is NGI includes human values of well being, rather than just environmental and economical well-being.

Alongside of this evolutionary journey, in 2016 and 17, there were a number of very significant security attacks, particularly the Mirai Dyn attack and WannaCry NHS attack. In response, world wide Governments have begun developing strategies, initiatives, and in some instances, laws to strive to reduce IoT security vulnerabili-

ties. Other technologies including Artificial Intelligence (AI), machine learning and Blockchain are being combined with IoT to produce more powerful tools. Similarly, Augmented Reality (AR) and Virtual Reality (VR) technology are being combined to with IoT to create a more interactive user experience.

This evolution of ICT technology, which culminated in IoT has taken 200 years. In comparison, from the initial concepts of IoT in 1999, through to introduction of IoT to the consumer around 2014, which has led onto widespread adoption of IoT technology. The IoT development cycle is just 15-20 years and as a result of this rapid development, IoT faces major issues, the most significant of which are security vulnerabilities. The severity of IoT security vulnerabilities are because security has been a developmental afterthought. Designers, developers and policy makers worldwide are now looking for ways to reduce this issue. In 2018 the British Government released the world's first IoT code of practice entitled 'Secure by Design' [36]. This code aims to "remove the burden from consumers to securely configure their devices and instead ensure that strong security is built into IoT devices and services by design" [35], also the British Government is currently consulting over whether to mandate security laws for IoT consumer products [37]. Other related methodologies, strategies and technologies are also being researched and developed [121, 104, 23, 80]. Additionally, the British Government is investing 30.6 million into 'Security of Digital Technology as part of the the Periphery' (SDTaP) research program. This investment has included the opening in March 2019 of The PETRAS (privacy, ethics, trust, reliability, acceptability, and security) National Centre of Excellence for IoT Systems Cybersecurity. The national centre of excellence is a collaboration between a number of universities, including Imperial College London, Bristol University and 150 industrial partners.

3. Related Work

There is a large number of tutorials, surveys and research studies in the area of IoT. Significant surveys [16, 5, 77] consider IoT concepts and technologies as a whole, including the architectures, technologies and principal applications of IoT. Atzori et al. [17], develops his earlier survey [16], challenging the popular idea that IoT can be used to solve any issue. Many real IoT smart city deployments are detailed and analysed, including these works [2, 126, 83].

Further works focus on technologies or challenges within IoT systems, these include works comparing IoT architectures [127, 85, 5, 125]. Specific communication technologies [5] are defined and compared. Hejazi et al. [54] compare IoT cloud platforms defining strengths, weakness and where they each fit within the IoT sector.

Similarly, IoT Operating Systems (OS) are detailed and compared [24, 30]. Bujari [26] considers current challenges including interoperability, security, privacy, and business models. Security challenges are studied [124, 44, 7] and Alaba [7] surveys existing security solutions. Stergiou [115] surveys IoT and Cloud Computing from the perspective of security, Yang et al. [124] and Granjal et al. [47] both study IoT security vulnerabilities and analyse the effectiveness of security strategies. Gupta et al. [50, 51] have created a number of security books including a practical and detailed handbook which surveys security across a range of ICT including wired and wireless systems, ad-hoc networks, human wearables and cloud computing. The second [51] is a comprehensive book covering security trends, cyber risk, vulnerability assessments, the human factor, smart phone protection, critical infrastructure protection. It also introduces security policies and techniques including cryptography, standards and modelling.

The contributions of this paper relative to existing literature can be summarised as this paper is written with:

- Consideration of practitioners and researchers from neighbouring fields.
- A brief history and overview of the evolution of IoT, demonstrating where IoT sits within ICT and current trends including Industrial IoT, Smart Cities and Next Generation Internet.
- Consideration of IoT security vulnerabilities.
- Recommendations to reduce security threats.
- Architecture and technologies are considered from the perspective of designing, developing and securing large Next Generation Internet IoT systems, this includes quick reference technology comparison tables (Tables Table 1, Table 2, and Table 3).

4. IoT Technologies

IoT is not a single technology, but a system or framework comprised of many technologies. This section will begin by considering some definitions of IoT and then introduce three IoT architectures, before looking in more detail at one specific architecture, considering the the technologies and security vulnerabilities within each of its layer. Some security attacks that can be applied to IoT systems including node capture, eavesdropping, malicious control, IP Spoofing, Ping of death, sniffing, malicious code injection and denial of service. These attacks will be discussed and

mitigation techniques suggested. Technologies within IoT systems including Things, communication technologies, management platforms, data management tools and user applications will also be introduced.

4.1. Definitions and Concepts

The IEEE has developed two definitions of IoT [87, 56], the first is with respect to simple IoT networks, and states:

"An IoT is a network that connects uniquely identifiable "things" to the Internet. The "things" have sensing/actuation and potential programmability capabilities. Through the exploitation of unique identification and sensing, information about the "thing" can be collected and the state of the 'thing' can be changed from anywhere, anytime, by anything."

The second definition is specific to larger networks, for example smart cities, and it states:

"Internet of Things envisions a self-configuring, adaptive, complex network that interconnects 'things' to the Internet through the use of standard communication protocols. The interconnected things have physical or virtual representation in the digital world, sensing/actuation capability, a programmability feature and are uniquely identifiable. The representation contains information including the thing's identity, status, location or any other business, social or privately relevant information. The things offer services, with or without human intervention, through the exploitation of unique identification, data capture and communication, and actuation capability. The service is exploited through the use of intelligent interfaces and is made available anywhere, anytime, and for anything taking security into consideration."

From both of the definitions, a number of characteristics can be highlighted:

1. IoT is a system that incorporates and connects 'Things'
2. Things sense or monitor their environment
3. Things connect to the Internet to communicate
4. Things are uniquely identifiable
5. The system can potentially compute data, for example use, process, store or transmit data onward
6. The system should present information to a user or multiple users
7. The system responds to input from connected Things and, or users

.

The National Institute of Standards and Technology (NIST) do not define IoT, instead they state some of the characteristics of IoT. These characteristics resemble the IEEE definitions. NIST state that IoT systems "involve sensing, computing,

communication, and actuation" [120]. Inline with these definitions [56], and others [120, 12, 43, 59, 16, 42, 91, 40, 33], Internet connected enterprise infrastructures, PCs, laptops, tablets and smart phones will be considered part of IoT.

IoT has use-cases in many areas of life, but in the last few years the focus of IoT has been moving away from small, independent and unconnected networks towards more joined-up infrastructures and networks, particularly with a focus on smart cities. In 2016, the ITU-T SG20 IoT working group changed its name to 'ITU-T SG20: Internet of things (IoT) and smart cities and communities (SCC)'. This group is currently developing 83 smart city standards, each focused on a different aspect of smart city infrastructure such as architecture [65, 66, 69], data sharing [64] or security [67, 68]. Similarly, the IEEE P2413 IoT working group has created a smart city group, IEEE P2413.1. They are currently developing 'P4213.1 Standard for a Reference Architecture for Smart City (RASC)' [13].

The concept of smart cities and sustainable cities have been around since the mid 1990s [70]. Initially, smart cities simply referred to cities with economic improvement strategies, next the concept included use of ICT within city infrastructures, later the concept became more citizen centric. Today, most stakeholders would agree that smart cities include technology in their infrastructures, enabling them to serve their citizens, providing "more efficient services to citizens, to monitor and optimize existing infrastructure, to increase collaboration amongst different economic actors and to encourage innovative business models in both private and public sector" [82]. In addition to the well being of citizens and city infrastructure, the environmental sustainability of a city's operations has also become an important feature, "Cities become smart sustainable when smart ICT is employed for making them (the cities) more sustainable" [22]. A more recent concept is that of Next Generation Internet, in combination with smart sustainable cities, resulting in 'Next Generation Sustainable Cities'. There are a number of important characteristics of these cities. Firstly, next generation sustainable cities seek to "shape the future internet as an interoperable platform ecosystem that embodies the values .... openness, inclusivity, transparency, privacy, cooperation, and protection of data" [60]. Secondly, next generation sustainable cities are 'next generational', the technology they are comprised of is developed based upon the analysis of previous and existing generations of the technology [95]. Thirdly, next generation sustainable cities are built with next generation technologies, for example, 5G telecommunications, intelligent technologies including machine learning and AI. Thirdly, NG sustainable cities are are citizen centric, promoting the health and well-being of all citizen, the city and its environment impact and sustainability.

4.2. Architectures

A technical architecture is a framework created to allow designers and developers to consider the system as a whole and also to break it down into sections. According to Global Standards 1 (GS1),"a reference architecture is an essential foundation to enable integrating the diverse technologies into IoT applications" [48]. There are many groups working on developing IoT architectures and other standards. These groups include IEEE P4213 Working Group, IEEE 802.24 Technical Advisory Group (TAG), IEEE P4213.1 Working Group, The National Institute of Standards and Technology (NIST) IoT Working Group, International Standard Organisation / International Electrotechnical Commission (ISO/IEC) 30141 JTC1 IoT Working Group 10, International Electrotechnical Commission Strategic (IEC) Group8, International Telecommunication Union Telecommunication (ITU-T) Group, oneM2M Consortium, Open Connectivity Foundation (OFC), Industrial Internet Consortium (IIC), and Internet Engineering Task Force (IETF). Though currently, none of these architectures are universally accepted, so below three architectures will be considered and compared.

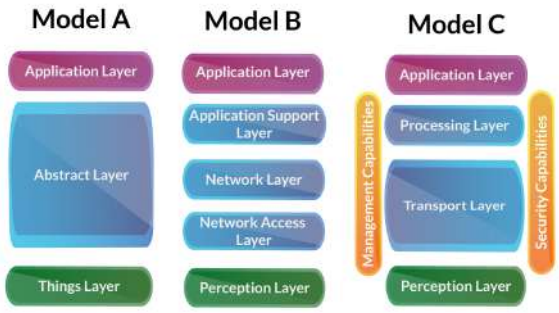


Figure 2: IoT architectures: (A) IEEE P4213 Three Layer Architecture [14]. (B) Zhong [127], Miao Wu [85] and Montagero's [90] Five Layer Architecture. (C) ITU-T Y.4000 Four Layer Architecture [41].

Historically, the most common IoT architecture is the Three Layer Architecture, illustrated in Figure 2-A. This framework is currently being developed further as part of the IEEE P4213 IoT architecture standard [14] which is based on the SO/IEC/IEEE 42010-2011 systems and software engineering architecture description standard [63]. The same framework is expanded upon within the IIC's industrial internet of things reference architecture [78]. Similar architectures are discussed [16, 85, 5]. This architecture comprises of the Things, Abstract and Application Layers [14, 107, 10]. The IEEE P4213 standard is open source and is a widely

accepted and supported standard which has been in development for a number of years, and is still being developed. The standard is very detailed, comprising over 100 pages with the aim of creating an architectural framework which is relevant to any industry or use case. Though, some researchers [127, 6] suggest the Three Layer Architecture is too high level and does not allow the different components of the IoT system to be separated out sufficiently to enable system development or protection. For the purpose of this paper, considering IoT from a security perspective, the authors agree with this conclusion, that further division of the IoT system will enable easier consideration of the security vulnerabilities. The next architecture considered is the ITU-T Y.4000 Overview of the IoT [41]. This standard comprises of a 4 layer architecture, with separate over-arching management and security capabilities, as shown in Figure 2-C. Again, for the purpose of this paper, it is more helpful to consider security within each architectural layers, rather than separating security out. In 2015 a group of researchers, Zhong et al. [127], introduced a Five Layer Architecture comprising of the Perception, Network Access, Network, Application Support and Application Presentation Layers, illustrated in Figure 2-B. Similarly in 2017, Montagero et al. [90] and Miao Wu et al. [85] developed an architecture based on the computer networking Open System Interconnect (OSI) technology architecture. Montagero et al.'s architecture resembled that developed by Zhong et al. Miao et al.'s architecture comprised of the same five layers as Zhong et al.'s, plus an additional Business Layer. From this point forwards, this architecture will be referred to as Zhong's Five Layer Architecture

Summary of the Architecture Layers:

1. The lowest layer, commonly referred to as the Perception Layer is the same across all three architectures and comprises of the physical layer that interfaces between the physical and information world, monitoring the environment and collecting data. The layer comprises of hardware devices including sensors and actuators. In this layer, the collected data is converted into digital data, ready for transmission up, to the next layer.
2. As can be seen from Figure 2, the Abstract Layer in Model A is subdivided in Model B and C. The Network Access Layer (Model B) is concerned with moving the digital data from the perception layer to an access node or gateway. The data is transmitted using access technologies like Ethernet, Wi-Fi, Bluetooth or Zigbee. The data is then ready to be used in the Network Layer.
3. The Network Layer (Model B) is concerned with transmitting data received at the access node throughout the whole IoT network, including the Application Support Layer and the Application Presentation Layer. These transmission technologies can include wired and wireless Internet protocols, for example

HTTP, MQTT and CoAp. In ITU-T's architecture (Model C), the Transport Layer is a combination of the Network Access Layer and Network Layer of Zhong's architecture (Model B).

4. The Application Support Layer (Model B), also referred to as the Processing Layer (Model C), is responsible for processing data. Dependant upon the size of the IoT system, this layer can be very complicated as it is responsible for processing and combining data from many different sensors and other devices and presenting it ready for use in the Application Layer. Technologies within this layer can include management platforms and technologies responsible for data processing, analysis and storage, this can include cloud technologies.
5. The Application Layer allows the end-user to make use of the collected data. The layer comprises of tools to develop and manage end-user applications. These applications are commonly referred to as 'Apps' which deliver IoT end-user services, for example health monitoring tools, smart homes or smart city applications.
6. Miao Wu et al.'s Architecture differs slightly from Zhong's Architecture because it includes an additional layer, the Business Layer. This layer can be considered as the "manager of the Internet of Things" [85], concerned with business and profit models, management of data sharing [5], software updates and system interoperation. This layer is very important and must be considered in the design and development of an IoT system, particularly within large systems such as smart cities. Miao Wu et al. explain that the success of a technology does not only depend on development of the technology, but also the innovation and development of the business models that manage how the technology will be used. Based on this point, the Internet of Things may not have long-term future without the significant development of its business models [85]. This layer is outside of the scope of this paper.

.

Throughout the rest of the paper, when an architecture is referred to, it is Zhong's Architecture, labelled Model B in Figure 2. When considering the movement of data throughout all of the layers of the architecture, it is important to highlight that data can move in the opposite direction to that explained above, from the Application Layer back down to the Perception Layer, enabling actuators to respond to collected data, system instructions or end-user instructions.

Other well referenced architectures include the oneM2M IoT architecture [117], cloud centric architecture [49], software stack architecture [113], TCP/IP architecture, OSI reference architecture [85] and Representational State Transfer Services

(RESTFUL) architectural style linked to HTTP [72]. Next, the paper will look in more detail at the technologies that exist within each of layer of Zhong's architecture.

4.3. IoT Technologies within the Perception Layer

As defined earlier, the end point of an IoT system is the 'Thing' that interacts with itself, other Things or its environment [14]. Generally, the Thing is collecting, responding data from the physical world or responding to instructions from the IoT system. The technology in is this layer is shown in Figure 3 and comprises of the hardware and software components that enable any physical object to act as an IoT Thing.

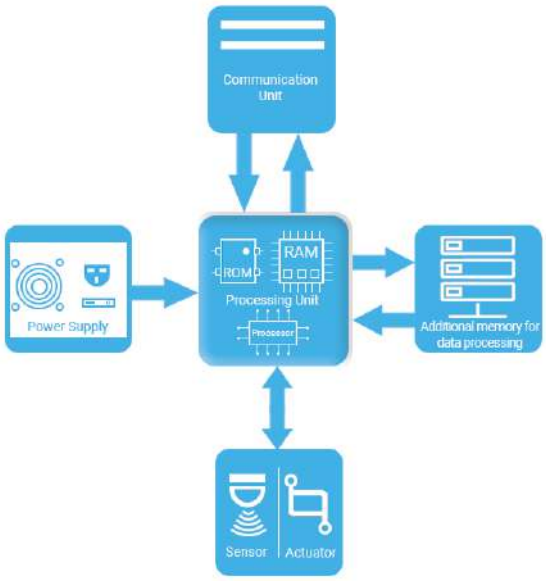


Figure 3: Components of an IoT Thing

4.3.1. Sensor & Actuator Unit

Sensors and Actuators are electrical components that connect the real and digital world by monitoring or responding respectively. Sensors are input components that monitor environmental characteristics and convert changes in this characteristic into an electrical signal. The relationship between environmental characteristics and electrical signal outputs can be linear or non-linear, this relationship is defined within the sensor's specifications.

Actuators are output components. They create a physical response to a change in their electrical input. Generally actuators fall into four categories based on the nature of their physical response:

- Hydraulic - moving under pressure liquids through a defined space
- Pneumatic - using gas stored under pressure
- Electric - generating electricity
- Mechanical - operating machinery

Similar to sensors, the relationship between electrical signal and actuator output is defined within its specifications. Generally, IoT sensors and actuators are small in size, low complexity and low unit cost. In some system, particularly critical infrastructure systems, higher specification sensors or actuators may be required, this can increase the cost of these components. Examples of component specifications include fault tolerance, sensing resolution, sensing rate, response rate, sensing range, magnitude of response, accuracy, security, storage or processing capabilities. When designing a system, component specifications and cost are important considerations.

4.3.2. Processing Unit

The processing unit is comprised of hardware and software that manage the behaviour of the Thing. This unit is commonly a microcontroller; a small, simple, programmable, self-contained computer on a single integrated circuit comprising of a processor, ROM, RAM and IO. Some microcontrollers that are commonly used with IoT projects and have significant amounts of development support.

A microcontroller is usually built and programmed to carry out simple tasks related to a single simple function, for example, controlling a smart fridge. They can be built into more complex devices, for example smart phones, then they are referred to as 'embedded'. Due to their constrained specifications, microcontrollers can not run traditional operating systems, instead they use constrained operating systems, for example mBed, TinyOS or rasbarian [102]. A microcontroller can have additional memory added to enable it to execute additional processing tasks, for example, to carry out data processing or data analysis at the sensor to reduce the amount of data transmitted from the sensor to the IoT network. This is sometimes referred to as 'edge' analysis. Additionally, the microcontroller can include security features such as data encryption and endpoint authentication.

4.3.3. Communication Unit

This unit is responsible for transferring or receiving data. If data is collected at the sensor, the communication unit will transfer the data to the IoT network. Alternatively, if data is received from the IoT network, it will then be acted upon by the actuator. Communication technologies are considered in more detail in Section 4.4. Generally microcontrollers have communication unit inbuilt, so when choosing a microcontroller, the communication and security requirements of the IoT system must be considered, for example, the desired data transfer rate and transfer distance. These requirements will also affect which communication technology is most suitable, for example for a contactless card payment system, the most suitable technology is Near-Field Communication (NFC) RF technology with a transfer distance of $10cm$. Within the IoT endpoint, communication consumes the largest proportion of power, therefore the communication technology also affects the power supply requirements.

4.3.4. Power Supply

Endpoint power requirements vary significantly dependant upon the energy requirements of all of the above components. If the Thing is connected to a mains power supply, the power requirements of the Thing can be considered nearly negligible in comparison. If the Thing is fixed in one location, for example a smart fridge, the power supply would generally be wired, for example to the mains power supply. Often though, IoT endpoints are mobile, meaning a wired power supply may not be suitable. Instead for mobile endpoints, power could be supplied from a portable source, for example batteries or a renewable source, for example solar panels or an energy harvesting unit. According to Andersen, for a wireless IoT system to be commercially viable, each Thing should have a power supply lifetime of $5 - 10$ years [11].

The criticality of an IoT system must be considered when selecting a power supply. For a highly critical system, for example a warning system within a power plant, the reliability, redundancy and security of the power supply and its connection must be considered. In this example, solar power is unlikely to be selected as the only power supply. Instead, multiple power supplies may be connected via multiple techniques to provide sufficient confidence.

4.4. IoT Technologies within the Network Access Layer and Network Layer

The movement of data within a network is referred to as communication, similarly, terms 'connect', 'transfer' or 'transmit' can be used. Within Zhong et al's architecture, the communication of data is be broken into two categories; access communication technology which sits within the Network Access Layer, and network

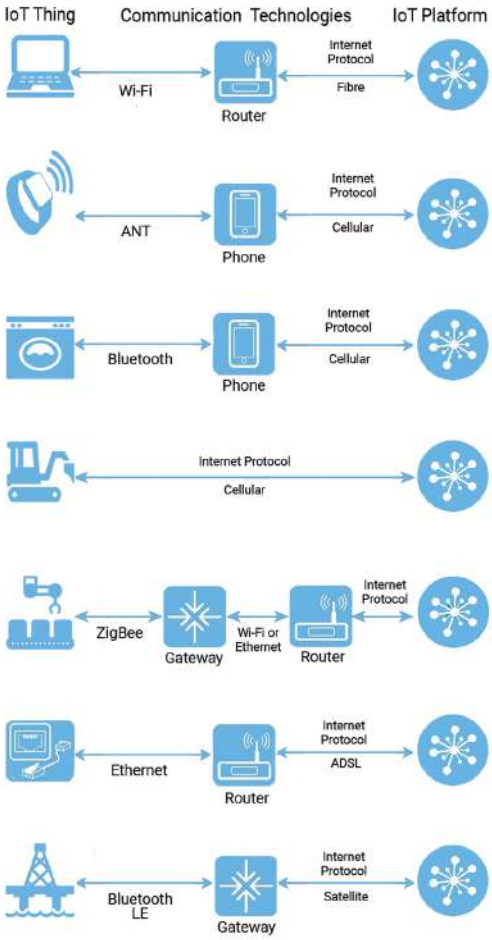


Figure 4: IoT Short to Medium Range Communication Configurations

communication technology which sits within the Network Layer. Access communication technologies connect the IoT Thing to the IoT network, usually achieved by transmitting data from the IoT endpoint to a network access gateway. The network communication technologies transmit data around the rest of the IoT network, from the access gateway all the way to the Application Layer. These communication technologies are the where most IoT security vulnerabilities exist. Figure 4 is an illustration of some short to medium range communication configurations, demonstrating technologies from the Perception, Network Access Layer and Network Layer. This illustration is not an exhaustive list of technologies or configurations.

4.4.1. Access Communication Technologies

Sometimes IoT networks are oversimplified to only include wireless networks, but this is wrong, IoT networks can be wired too. Wireless networks are commonly used in hard to reach or hard to install environments, or where wired network installation is more costly. Conversely, wired networks may be used where the wired infrastructure already exists, higher data throughput or increased security is required.

Due to power supply restrictions, wireless networks should be designed to have low power requirements. This means wireless networks are well suited to systems with short transmission distances and low transfer rates. The choice of which communication technology and protocol to choose is generally based on the system requirements, including:

- Wired / wireless
- Data rate
- Data reliability
- Data security
- Proximity of the Thing to receiving node
- Nature of environment
- Ease of access / maintenance
- Number of connected Things
- Number of simultaneously active Things
- Overall system size and complexity

	TECHNOLOGIES						
	Bluetooth	Bluetooth Low Energy	ANT	Wi-Fi	NFC	Zigbee	Z-Wave
Range (m)	100 m	50-100 m	30 m	30-50 m	5-10 cm	10-100 m	30 m

Table 1: Wireless Short to Medium Range Communication Technologies (Continued over page)

Bwidth (Hz)	2.4 GHz	2.4 GHz	2.4 GHz	2.4 / 5 GHz	13.6 MHz	2.4 GHz	2.4 GHz
Data Rate (bps)	1-3 Mbps	125 Kbps - 1 Mbps	12-60 Kbps	150-200 Mbps	100-420 Kbps	250 Kbps	9.6, 40 or 100 Kbps
Battery Lifetime	0.6 Ah: Standby: 3 mnth. Mixed: 5 dy	1 Ah: Mixed: 1-2 yr. 2xAA: 14 yr	1 Ah: Mixed: 15 yr	2xAA: Listening: 2 dy	Initiator trf: 15 mAh. Passive: 0 mAh	2xAA: Mixed: 5 yr	2xAA: Mixed: 1 yr
Authentic^{n}	Yes	Problematic	Yes	Yes	Yes	No	Yes
Encrypt^{n}	Yes	Yes	No	Yes	Yes	Yes	Yes
Standard	Based on IEEE 802.15.1	Bluetooth 4.2	Proprietary	IEEE 802.11	ISO/IEC 14443, 18092	IEEE 802.15.4	Z-Wave Alliance Proprietary
Scalability	Yes	Yes	Yes	No	No	Yes	Yes
Topology	P2P, Star (Piconets)	P2P, Star (Piconets)	P2P, Star, Tree, Mesh	P2P, Star	P2P	Mesh	Mesh
N^{o} Nodes (Mst : Slv)	8 (1:7), (200 inactive slv)	8 (1:7) (32K inactive slv)	65533 (per 8 channels)	255	2	232	232

Table 1: Wireless Short to Medium Range Communication Technologies [21] [1] [114] [123]

Defining a communication technology typically defines two elements; firstly, the nature of the transmitted data signal, and secondly the material through which the signal is transmitted. For example, Ethernet technology comprises of an electrical

signal transmitted along an Ethernet cable. For Bluetooth technology, an electromagnetic RF signal is transmitted through air at 2.4GHz. Communication technologies that do not require a wired medium are referred to as wireless.

Communication protocol refers to the rules that define how the data should be transmitted, for example, the number of bits in a data packet, which bits are real data and which are management data. Management data is generally transmitted as a data header or footer, and used to control the flow of data, including, defining the destination address, and how the data should travel to the destination.

	TECHNOLOGIES					
	Cellular 3G	Cellular 4G	SigFox	LoRa	NB-IoT	LTE-M
SPECS						
Urban Range (Km)	5 - 8 km	15 km	3 - 10 km	2 - 5 km	9 km	11 km
Rural Range (Km)	50 - 70 km	45 km	30 - 50 km	15 km	unavailable	unavailable
Transm^{n} Bandwidth (Hz)	800 MHz - 2.4 GHz	800 MHz - 2.6 GHz	868 MHz	850 MHz - 1 GHz	200 KHz, 700 - 900 MHz	700 - 900 KHz, 1.4 MHz
Data Rate (Kbits/s)	Mobile: 128, 144 Kbps. Fixed: 2 Mbs	Mobile: 20 - 100 Mbps	0.3 Kbs	0.3 - 50 Kbs	150 Kbs	64 - 128 Kbs
Power (mAh)	460 mAh	600 mAh	32 - 51 mAh	40 mAh	unavailable	80 mAh

Table 2: Wireless Wide Area Communication Technologies (Continued over page)

Standards	UMTS, HSPA, W-CDMA, WLAN, WiMAX,	OFDM, CMDA, WiMAX, LTE, LTE Adv	Proprietary	Proprietary	SC-FDMA, PRACH	OFDM, PRACH
Uses	Messaging, Internet, VoIP, IPTV	Messaging, Internet, VoIP, Games, Cloud	IoT systems	IoT systems	IoT systems	IoT systems
Battery Life	hours - days	hours - days	5 - 10 years	10 years	>10 years	>10 years

Table 2: Wireless Wide Area Communication Technologies [15] [76] [97] [94] [122] [119] [31]

For an IoT network spanning a few meters, across a single building, or collection of buildings, short to medium-range communication technologies should be selected. Communication technologies can be categorised based on their network range and use. These categories include the body area network (BAN), personal area network (PAN), local area network (LAN) and wide area network (WAN) technologies. BAN technologies include Ant, PAN technologies include NFC, Bluetooth, Bluetooth Low Energy, Zigbee and Z-wave. LAN technologies include Wi-Fi and Wi-Fi Low Energy, additionally Ethernet is a wired LAN technology. Some wireless short to Medium range technologies are compared in Table 1.

Many IoT systems, particularly larger networks, utilise numerous access communication technologies to connect multiple Things to the network. For an IoT system that covers a larger area, for example a Smart City, short and medium-range communication technologies may not be suitable to connect all of the endpoints to the network [29, 110]. Wired and wireless WAN technologies can be used. Significant work by Centenaro et al. considers cellular and low power WAN technologies, creating networks that span $10 - 50km$ in rural areas and $3 - 5km$ in urban areas. Their work demonstrates these technologies can be suitable for relatively harsh outdoor environments. Some Wireless WAN communication technologies, including Cellular (3G and 4G), Sigfox, LoRa, NB-IoT and LTE-M, are compared in Table 2. Other WAN technologies include Neul, NWave, PLC, Ethernet, Weightless -N, Weightless -P.

4.4.2. Network Communication Technologies

Once data is transferred from the Thing to the access node, the data is available to the Internet to be transported throughout the top three layers of the network using Internet protocols (IP). HTTP and its secure variant HTTPS are the most well known IP, but due to large control data overheads ensuring data reliability, HTTP and HTTPS may not be the most suitable protocols for constrained IoT systems. So alternative and lighter protocols have been developed that are more suitable for constrained IoT systems. Examples include Constrained Application Protocol (CoAP), Advanced Message Queuing Protocol (AMQP), Extensible Messaging and Presence Protocol (XMPP), Message Queue Telemetry Transport (MQTT) and Data Distribution Service (DDS). Some of these protocols offer security feature similar to HTTPS.

Platform	Focus / Tools	Local/ Cloud	Language	Cost ($): Free Vs 10,000 Connected Devices
Ayla Network	E2E. Compatible: AMAP. Tools: embedded agents, Phone-as-a-Gateway, ADP	PaaS	C, Java	Custom pricing
Arm MBED IoT Platform	Compatible: MBED OS. Support: ARM/ARMcommunity. Tools: security E2E, easy integration, open standards	PaaS, local OS	C/C++	Custom pricing
AWS IoT	E2E. Focus: extreme scalability, many partners. Tools: recognition registry for Things, device SDKs, rules engine - message evaluation	IaaS	NET, Java, JVM, Node. js, Python, Ruby, PHP	**Free**: 50 Devices. Daily: 300 msg, 130 registry actions, 150 exceptions **10,000 Devices**: 1 KB msg/min = $560/month [18]
Bosch IoT Suite	E2E. Focus: cost, local&cloud, security. Tools: analytics, open standards	PaaS, local	Unknown	Custom pricing

Table 3: Enterprise IoT Management Platforms (Continued over page)

Platform	Focus / Tools	Local/ Cloud	Language	Cost ($): Free Vs 10,000 Connected Devices
Carriots	Focus: customer access hierachy, easy tool/app integration. Tools: debug/logs, data export, SDKs, API design	PaaS	Groovy	**Free**: 2 Devices, 500 msg/day, 5 KB/msg **10,000 Devices**: $2/Device (up to 1 MB/day) = $20,000/month [28]
Cisco IoT Cloud Connect	Focus: agriculture, customer relations. Tools: Devices connect through cellular (sim) network , voice/data connectivity	PaaS	Unknown	Custom pricing
Datav by Bsquare	Tools: predict/analyse issues, automate maintenance/repairs, max utilisation	PaaS	Unknown	Custom pricing
General Electric's Predix	Compatible: GE apps, products, partners. Focus: healthcare, transport, energy. Tools: asset digital twin modelling	PaaS	Java, Ruby, Node.js, Python	Custom pricing
Google Cloud	E2E. Tools: partnerships with device /app providers, big data analytics, Google's fast fibre network	IaaS	PHP, Java, Node.js, .Net, Ruby, Go, Python	**Free**: 50 Devices, 2800 msg/day (upto 250 MB, then charged minutely) **10,000 Devices**: 1 KB msg/min = $1940/month. [45]
Universal of Things HP	Focus: scalability. Tools: 'market place' for billing, easy app design, analytics	PaaS, local	Unknown	Custom pricing

Table 3: Enterprise IoT Management Platforms (Continued over page)

Platform	Focus / Tools	Local/ Cloud	Language	Cost ($): Free Vs 10,000 Connected Devices
IBM Watson IoT Platform	Compatible: IBM Bluemix. Focus: beginners. Tools: ADP, security, weather data, real-time data, significant storage	PaaS	Java, C, C#, mBed-C++, Python, Node.js/ RED	**Free**: 50 Devices, 1920 msg/day (100 MB/month) **10,000 Devices**: 1 K msg/min = $421.68/month [55]
Kaa IoT Platform	Focus: open source, scalability, industry, low R&D time/cost. Tools: SDKs	PaaS	Java, C, C++ ObjectiveC	Free
LTI's Mosiac	Focus: oil/gas, security/risk compliance, manufacturing. Tools: analytics, insight	PaaS	Unknown	Custom pricing
Microsoft Azure IoT	E2E. Focus: AWS competitor. Tools: rule evaluation engine, device security shadowing, real-time analytics	IaaS	C, Node.js, Java, .NET, Python	**Free**: 50 Devices, 144 msg/day (8 K/day) **10,000 Devices**: (Tier S3) 1 KB msg/min = $3726.55/month [86]
Mocana	Focus: Military level security and tools	PaaS	Unknown	Customised pricing
Oracle Integration Cloud	Focus: manufacturing, logistics, security, scalability. Tools: device virtualisation, big data analytics, fast messaging.	PaaS	Java, Java Script, Node.js	**10,000 Devices**: from $1.6129/hour = $1161.28/month [96]
PTC ThingWorx	Focus: fast develop/deploy. Tools: big data analytics, machine learning, deployable in Device/local/cloud	PaaS, local	C, Java, .NET, iOS, Android	Custom pricing

Table 3: Enterprise IoT Management Platforms (Continued over page)

Platform	Focus / Tools	Local/ Cloud	Language	Cost ($): Free Vs 10,000 Connected Devices
Salesforce IoT Cloud	Focus: capture sales leads, customer relations. Tools: CS management, automate: service request, repair, feedback	PaaS	Rubyon Rails, Java, Node.js, Python,	$4000/month
Samsung Artik	Focus: security, easy to use, optimum system performance	PaaS	PHP, Java, Swift, C++	**Free**: 50 Devices, 72 msg/day (100 K msg/month)
Cloud			Ruby, Java-Android, Python, C	**10,000 Devices**: (Small Business Tier) 1 msg/min = $6480/month [105]
Siemens Mindsphere	Focus: cost-effective, open source based, security. Tools: machine data, confidential storage, embedded agents, libraries	PaaS	Unknown	Custom Pricing comprised of Connectivity, Access and Data

Table 3: Enterprise IoT Management Platforms

4.5. IoT Technologies within the Application Support Layer

Device management, data analysis and processing is handled within the application support layer. For systems with more than a few connected Things, a management platform can handle these tasks. In 2017 more than 450 companies offered IoT Platforms [62]. Platforms can specialise in End-to-End (E2E) solutions, system security, application enablement, device management, analytics, cloud storage and back-end connectivity.

A management platform should enable the following actions or services:

- Synchronise with and monitor connected Things
- Control and retrieve data from Things
- Respond to received data

- Manage system security
- Offer device dashboards to review analytics

Table 3 compares 20 enterprise IoT platforms [111], considering their focus, tools, development languages, and what IoT system can be developed and operated using any free allowance, versus the cost to run a system with 10,000 connected devices. The cost comparison for a 10,000 device system includes each device sending a message once per minute, but excludes data processing, data analysis, device shadowing, rule triggering and other actions which can add additional costs. Developers might need to do further research and testing to determine which platform is best to manage their network.

4.6. IoT Technologies within the Application Presentation Layer

The technology within the Application Presentation Layer includes the Application Development Platform (ADP), which is a tool to enable developers to create and manage end-user software applications. The end-user applications consume the data that was collected by connected sensors and processed in the previous layers, and then present it to the user in a usable format. Some management platforms considered in Section 4.5 include ADPs, for example ThingWorx, Carriots and Kaa. Other management platforms interface with specific ADPs, for example IBM's Watson Management Platform interfaces with IBM's IoT ADP. Additionally, many management platforms also integrate with third party ADPs.

The ADP tools are briefly introduced below. A detailed review is carried out by Ray et al. [101]. Within an ADP, tools can include an Application Programming Interface (API) and Software Development Kits (SDK). In general terms, an API is a block of code acting as an interface between two different objects to enable them to communicate. The API usually comprises of commands, functions and protocols. Within IoT, an API is the code which acts as a logical connector and translator between the connected Thing and an end-user software application enabling easy integration of the Thing into the IoT system and end-user application. Essentially, the API allows the application to access useful processed data. Generally, APIs are created by the manufacture of the IoT Thing.

In some literature, the terms API and SDK are used interchangeably, but they are very different. An SDK is not just code, but instead it is comprised of a whole set of development tools for example libraries, instruction documentation, APIs, samples of code and examples of processes. It may also include ADP guides to help a developer build end-user applications on a specific platform. Additional documentation within the SDK can include industry or user-specific guides. Comparatively, if an API is

thought of as a building block, an SDK can be thought of as a complete workshop full of all of the tools, instructions and building blocks. An API, or multiple APIs can be part of an SDK. Generally, manufacturers create the initial SDK for an IoT device, and developers can contribute to the SDK. Developer contributions are particularly common in an open source environment.

Many ADPs require the developer to be familiar with some programming languages, for example Node.js, Perl, Python, Java or C. Though some platforms have been developed to encourage non-technical developers to create end-user Apps. An example of two such platform include IBM's IoT platform which uses Node-RED visual modelling layout tool employing drag-and-drop methods to connect hardware devices, APIs and on-line services [93]. Secondly, Mendix ADP also uses simple web and desktop based visual modelling tools [84]. Both platforms state that their tools reduce development complexity, time and cost.

4.7. IoT Security Throughout the Architecture Layers

Historically, IoT security has been an after-thought, rather than being considered throughout the design and development of a system. This after-thought approach has led to huge security problems within IoT networks due to no, or low security in IoT endpoints, within network gateways, and throughout the communication layers [103]. These vulnerabilities have led to attacks such as the 2016 Distributed Denial of Service (DDoS) attack against a small jewellery shop, who were under attack from more than 25000 IoT cameras. This attack was found and mitigated by security firm Sucuri [32]. Another very well known example is the Mirai DDoS attack in 2016 [39, 58] which caused Dyn, a large US network provider to temporarily cease providing IT services to its business customers including Amazon, Twitter, PayPal and Netflix, and as a result disabling customer websites. During both the Sucuri and Mirai attacks, hackers used active attack methods [53] to infiltrate a huge number of no-security, or low-security IoT devices and converted them into remotely controlled robots, known as botnets. These botnets were then used to look for other low security IoT devices before all of the botnets were then directed to carry out a DDoS cyber attack on both the Jewellers and Dyn causing the systems to overload with too much traffic. In order to understand the security challenges within IoT systems, this section will consider some common security weaknesses, where the weaknesses sits within the IoT Architecture [127] shown in Figure 2B, the nature of attacks and a range of solutions.

Two important definitions are that of passive and active security attacks. A passive attack can be defined as activity where an unauthorised user, referred to as an attacker, attempts to read data within a network. This action is passive since the

attacker does not attempt to make changes to the data. In comparison, an active attack is when an attacker makes efforts to change data within the network. In both passive and active attacks, the behaviour is unauthorised and for malicious purposes.

4.7.1. Security within Perception Layer & Network Access Layer

The Perception Layer comprises of the sensors and actuators. The Network Access Layer comprises of transmission nodes that allow the access of data into the IoT gateway. Security attacks within these layers are typically the easiest to execute and are generally focused on the acquisition of data. The purpose of attacks in these layers are (1) to snoop on and collect data, (2) to stop sensor from functioning, which can cause a partial denial of service (3) replace sensor data with false data. Sensor snooping is generally a passive attacks, for example employing node capture and eavesdropping techniques [77]. In comparison, active attacks such as hardware jamming can be applied to stop sensors from functioning, or false data injection [53] can be used to replace the sensor data with false data, this in turn may affect the response of the IoT system.

Tools such as Attify can be used to intercept data that is collected by a sensor and transmitted to its node, or from the node to gateway. The attacker may use this data for reconnaissance enabling them to learn about the environment that the sensor is monitoring, or to enable the attacker to perform attacks in other layers of the network.

Another attack method is hardware jamming. Constrained IoT sensors are particularly susceptible to this type of attack which can be achieved in two ways; firstly by remotely injecting the sensor with code or secondly, by physically attaching unauthorised hardware to the sensor to jam it. Hardware jamming is applied for two purposes, to permanently damage the hardware sensor which will reduce or remove its computational power and stop the sensor from collecting data or converting its analogue data into digital data, known as actuating. In this way, hardware jamming can effectively remove sensors from the network, resulting in a sensor DoS. Alternatively, hardware jamming can be used to get vital data such as the cryptographic key or routing table, or to insert false sensor data into the system to affect the behaviour of the IoT system.

A sensor battery-depletion attack is another attack method which is similar to hardware jamming. An attacker can purposely reduce a sensor or actuator's power level, enabling the attacker to reduce its computational power, this can affect the sensor or actuator's ability to function, for example affecting reliable sensing, actuation or communication [108, 109] enabling an attacker to create a sensor or actuator DoS or. Alternatively, if a battery-depletion attack is applied to a sensor, the attacker

can insert false code, which could in turn affect the behaviour of the IoT system.

A relay attack is when an attacker eavesdrops on the communication between the sensor and its node or node and connected gateway. The compromised data is then relayed to another system, a victim system, to make the victim system carry out actions defined by the attacker [99]. Due to the significant growth of constrained IoT devices, this type of attack is increasing in frequency.

Security techniques and strategies to defend against all of these attacks include changing default passwords, device/system authentication, strict firewall rules, static code analysis (SCA) executed within the IoT system or applications and network intrusion detection mechanisms. Authentication of all connected IoT devices is a mitigation method used to reduce the likelihood of malicious devices infiltrating the network. Similarly, safe booting is the technique of checking the integrity of the different operating system (OS) in connected IoT devices, it uses cryptographic hash algorithms. For IoT devices with limited power and computation power, WH and NH cryptographic algorithms are the most appropriate for safe booting [9].

4.7.2. Security within Network Layer

The Network Layer routes data around the IoT network. This layer is embedded deeper than the Perception and Network Access Layer so infiltrating this layer is more difficult [73]. Within this layer, the purpose of attacks is to breach the network to intercept the data within it, this is generally done with active attacks [53] and can include gateway attacks, Man-in-the-Middle, ARP cache poisoning, ICMP attacks, Ping of death, Pong attacks and IP spoofing [57, 52].

A gateway attack is similar to a relay attack applied in the perception layer, it can be used to block the connection between the sensors and the internet infrastructure, thus deleting sensor data or redirecting the sensor data, causing damage to the system and causing a DoS [100]. A Man-in-the-Middle attack is widely used to secretly intercept system data and then alter this data, giving the attacker the ability to capture and manipulate data in real time [34]. A sinkhole attack is related to a Man-in-the-middle attack, the attacker employs a vulnerability within the network layer to cause the dropping of delivery packets, thus preventing the packets from reaching their destination. These dropped packets can then be destroyed or redirected to a different destination which is harmful in an IoT environment resulting in a system wide DoS.

In addition to the interception attacks above, malicious control of this layer can enable sophisticated attacks on services within the next layers the Application Support Layer and the Application Layer, including attacking end-user services or applications.

Security techniques to mitigate these attacks include using firewall rules to instigate device white and black lists, enabling randomized algorithms for TCP sequence numbers, using short time to live (TTL) durations for the DNS cache, blocking applications with weak authentication features or forged packet discovery mechanisms [57]. Secure routing is the technique of routing data via multiple paths securely, this can which reduces the error exposure and acts as a network mitigation technique.

4.7.3. Security within Application Support Layer

The Application Support Layer is the brains of the IoT network because it is responsible for the management of devices and data. Many of the attacks within this layer are as a result of security attacks in the lower layers, also attacks not related to the layers below are sometimes similar in nature to the attacks described in lower layers, but some attacks are also independent of other layers. This layer is vulnerable to a broad range of attacks, including sniffing, malicious code injection [116] and particularly denial of service.

Denial of service is the most common attack here due to the significant number of network resources being used in this layer. This means there are many different types of DoS attacks that can be applied to prevent genuine users from being able to access IoT devices, the complete system or specific applications. DoS attacks typically occur by the attacker flooding a victim device, or multiple victim devices, with redundant requests or null sessions, making it impossible for genuine users to access the victim device, just like the famous Mirai attack [98].

This layer is also known to be vulnerable to malicious insider attacks which are performed by an authorised system user who tries to access information from other users or other devices in the IoT network [106]. Once the insider has access to other user accounts or devices, they can carry out unauthorised actions, for example issue unauthorised commands or access system credentials and vital system information, thus enabling the malicious insider to carry out more higher level attacks.

The Application Support Layer contains many of the shared resources, for example routing tables which can serve as an attack vector for attackers to observe shared resources and get the required information to enable them to carry out attacks on other areas of the IoT system. Similarly, third-party tools such as a Platform as a Service (PaaS) based management platform, or cloud computing data processing tools, they provide a third-party web service component which can be used by attackers to breach the IoT environment remotely.

Malware attacks are executed in this layer. Malware is a security program which is secretly placed inside a network to monitor the traffic without the system administrator being aware of it presence. There are a number of different techniques that

an attacker can use to install malware, including phishing emails, but once malware is in place, the data it collects generally enables further higher level attacks.

Many of these different attacks can be mitigated by enforcing robust security features such as strong authentication, intrusion detection mechanisms, traffic encryption and regularly checking that all of the technologies within the system have up-to-date software, and particularly that patches are applied where required [89]. Data fragmentation is also useful as a mitigation technique in which data within this layer can be split into various fragments and stored on different servers or other system locations, thereby reducing the risk of data theft, or rendering the theft as useless [112]. The hyper-safe lock-down of the write memory files and device boot-up and configuration files mitigates against the unauthorised customisation of the files that control the behavior of the IoT system. This lock-down is achieved by using point indexing which constrains changes in data into the pointer indexes [74]. When all of these measures are implemented and regularly checked, this should act as a good security barrier against external attacks.

4.7.4. Security within Application Layer

The Application Layer manages the user applications and provides services from the rest of the IoT system to the user applications. It is through these applications that an IoT system interacts with its users. These applications include smart home systems, smart healthcare, smart tracking and logistics, and smart city infrastructures and applications, to name a few. All of the services provided by this layer are dependent on the data actuated from sensors, communicated by the Network Layer, and, managed and processed by the Application Support Layer. Lots of data is handled by this layer, so vulnerabilities and threats exist from both within the IoT system and from the applications. Most of these threats are focused on manipulating the IoT application for the attacker‘s purposes. Many of the attacks that can be experienced in the previous layers can occur in this layer, but in addition to the previous attacks, there are also client-side application attacks.

Cross Site Scripting (XSS) is an injection attack during which attackers insert a client-side script such as java script to modify the application‘s web interface, enabling the attacker to trigger unwanted behavior and actions in the application, for example the attacker has the ability to completely change the content and behaviour of the user application [8].

A malicious code attack [53] is another attack vector used by attackers to disrupt the services provided by the application layer. It is sent by an attacker and can sometimes be executed by itself or triggered by the victim through another medium, for example through a phishing emails. The purpose of this attack can be to change

the data within a user application or to gain application credentials or other vital system information.

Intentional data loss is a vulnerability that can affect the application layer. Due to the large amount of data transmitted between the devices, an attacker can orchestrate a disruption in the network which can lead to data loss. In these circumstances, low-cost sensors and actuators are most affected as they generally do not have storage, error checking or redundancy features due to their constrained nature.

Another kind of phishing attack is prevalent in this layer. The attack is distributed through infected user emails with the purpose of tricking a victim into revealing their login credentials, or to tricking the application into accepted spoofed user credentials.

Many of the attacks experienced within the Application Layer occur due to non-standard application code which is written by the application programmer. Generally application programmers are concerned with application functionality rather than security, therefore secure coding techniques may not be employed. This non-standard code increases application vulnerabilities allowing malicious attackers to take advantage and cause damage to the IoT system.

The majority of these attacks can be mitigated with 'user validation' using integrity and encryption mechanisms to authenticate user interactions. The use of system anti-virus, firewalls and anti-malware programs are crucial. Finally, incoming and outgoing network traffic can be monitored, also, for a large scale IoT system, all of the sensor and actuator connections within the system can be monitored. All of these interactions can be monitored using systems such as a Network Intrusion Detection (NID) system. For most networks and IoT systems there are 'normal traffic' patterns, and the NID system can be trained to recognise normal traffic and detect outliers or anomalies.

5. Future Developments

As discussed throughout this paper, IoT can be used to target challenges and improve quality of life. In the opinion of the authors, one of the largest international challenges that we face today is this that of reducing energy consumption. Energy has become a very important human commodity. Yet, it is widely recognised that our main energy resources, that of coal, gas and oil are limited. So, there is the need to find alternative sustainable energy resources and to reduce energy consumption. Instead though, Worldwide, energy consumption is currently increasing, particularly in the world's emerging markets. Since the 1970s Asia and Africa's energy demands have increased approximately 7 fold [61]. In addition to these factors, at present,

"most people spend 90% of their daily lives indoors relying on mechanical heating and air conditioning, thus leading to buildings becoming the largest energy consumers worldwide" [27]. Within the US and the EU, buildings account for a staggering 40% of all of the energy that is consumed this those regions.

Existing Building Energy Management Systems (BEMS) generally measure and monitor energy usage. Some systems also offer automated control of the Heating, Ventilation and Air Conditioning (HVAC) Systems. These BEMS have been demonstrated to reduce energy consumption by up to 30%. An area of further future development of IoT technology is within advanced BEMS using the concepts of Next Generation Internet, to achieve further energy reductions of $30-40\%$ [19]. Such systems could include additional control, for example to automate opening or closing of windows, doors and other building assets, control of appliances or machinery, for example, turning off domestic appliances when energy consumption passes a usage threshold, or turning industrial machinery from 'stand-by' to 'off' outside working hours. Developments could also include data fusion techniques [3] to combine different IoT data sets, including weather data, zoned heating linked to room occupancy [38, 71], or lighting systems which respond to external daylight conditions [27]. Such system could manage windows and blinds causing windows to tilt slightly to reflect away sunlight during warm conditions, or blinds to open fully to make optimal use of external light conditions. Also, the consideration of the people using these buildings; their comfort and their building interaction expectations [4] AI, machine learning and gamification techniques could be employed to make these systems more intelligent, human-centric and energy conscience, allowing us to reduce energy consumption further. AI and machine learning are methods of increasing system intelligence, including human ethics and improving user experience. Gamification is the mechanics of gaming, applied in a real-life context to improve a user's experience of a system and increase their engagement with it.

Many different and often unrelated IoT BEMS are currently being developed, but future developments that focus on user engagement by including system resilience, delivering sustainability and combining more of these different techniques are most likely to result in consumer and industrial uptake, enabling ae significant reduction in energy consumption within buildings, which will in turn result in a significant reduction in worldwide energy consumption.

6. Conclusion

This paper seeks to be an introduction, overview, and reference guide for IoT systems, particularly considering security issues. Within this paper the authors

demonstrate that IoT is the culmination of advances within computing, communication technologies and the Internet, all combined with the human drive to improve our quality of life. Next, IoT architectures and technologies are introduced including a number of quick reference technology comparison tables. Following this, the significant IoT security vulnerabilities, which have appeared as a result of the rapid development of IoT are described and some mitigation techniques are discussed. Finally, a future area of development is introduced.

[1] Thomas Aasebo. 2017. Wireless Technology Bluetooth, Zigbee and Ant. (2017). http://its-wiki.no/wiki/

[2] E. Ahmed, I. Yaqoob, A. Gani, M. Imran, and M. Guizani. 2016. Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges. *IEEE Wireless Communications* 23, 5 (October 2016), 10–16. https://doi.org/10.1109/MWC.2016.7721736

[3] Kemal Akkaya, Ismail Guvenc, Ramazan Aygun, Nezih Pala, and Abdullah Kadri. 2015. IoT-based occupancy monitoring techniques for energy-efficient smart buildings. In *Wireless Communications and Networking Conference Workshops (WCNCW), 2015 IEEE*. IEEE, 58–63.

[4] K. Akkaya, I. Guvenc, R. Aygun, N. Pala, and A. Kadri. 2015. IoT-based occupancy monitoring techniques for energy-efficient smart buildings. In *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. 58–63. https://doi.org/10.1109/WCNCW.2015.7122529

[5] A Al-Fuqaha, M Guizani, M Mohammadi, M Aledhari, and M Ayyash. 2015. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys Tutorials* 17, 4 (Fourthquarter 2015), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095

[6] S Al-Qaseemi, H Almulhim, H Almulhim, and S Chaudhry. 2016. IoT architecture challenges and issues: Lack of standardization. In *2016 Future Technologies Conference (FTC)*. 731–738. https://doi.org/10.1109/FTC.2016.7821686

[7] Fadele Ayotunde Alaba, Mazliza Othman, Ibrahim Abaker Targio Hashem, and Faiz Alotaibi. 2017. Internet of Things security: A survey. *Journal of Network and Computer Applications* 88 (2017), 10–28.

[8] Bako Ali and Ali Awad. 2018. Cyber and physical security vulnerability assessment for IoT-based smart homes. *Sensors* 18, 3 (2018), 817.

[9] MOJTABA Alizadeh, Mazleena Salleh, Mazdak Zamani, Jafar Shayan, and Sasan Karamizadeh. 2012. Security and performance evaluation of lightweight cryptographic algorithms in RFID. *Communications and Computing* (2012), 45–50.

[10] Leonardo Albernaz Amaral, Everton de Matos, Ramão Tiago Tiburski, Fabiano Hessel, Willian Tessaro Lunardi, and Sabrina Marczak. 2016. Middleware Technology for IoT Systems: Challenges and Perspectives Toward 5G. In *Internet of Things (IoT) in 5G Mobile Technologies*. Springer, 333–367.

[11] Michael Andersen. 2015. Trends in Internet of Things platforms. *XRDS: Crossroads, The ACM Magazine for Students* 22, 2 (2015), 40–43.

[12] Kevin Ashton. 2009. That 'Internet of Things' Thing. *RFiD Journal* 22, 7 (2009), 97–114.

[13] IEEE Standards Association et al. [n. d.]. P2413.1 - Standard for a Reference Architecture for Smart City (RASC). ([n. d.]). https://standards.ieee.org/project/2413$_1$.*html*

[14] IEEE Standards Association et al. 2016. P2413-Standard for an Architectural Framework for the Internet of Things (IoT). *Institute of Electrical and Electronics Engineers, New York* (2016).

[15] AT&T. 2013. Comparing LTE and 3G Energy Consumption. (2013). https://developer.att.com/application-resource-optimizer/docs/best-practices/comparing-lte-and-3g-energy-consumption

[16] Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The Internet of Things: A Survey. *Computer Networks* 54, 15 (2010), 2787–2805.

[17] Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2017. Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks* 56 (2017), 122–140.

[18] AWS. 2017. AWS IoT Core Pricing. (2017). https://aws.amazon.com/IoT-core/pricing/

[19] Bharathan Balaji, Jian Xu, Anthony Nwokafor, Rajesh Gupta, and Yuvraj Agarwal. 2013. Sentinel: occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM, 17.

[20] John Barber. 2017. Getting Answers to Your IoT Questions. (2017). http:www.gartner.com/podcasts/getting-answers-to-your-IoT-questions/

[21] Peter Barker and Mohammad Hammoudeh. 2017. A Survey on Low Power Network Protocols for the Internet of Things and Wireless Sensor Networks. In *Proceedings of the International Conference on Future Networks and Distributed Systems*. ACM, 33.

[22] Simon Elias Bibri and John Krogstie. 2017. Smart sustainable cities of the future: An extensive interdisciplinary literature review. *Sustainable Cities and Society* 31 (2017), 183–212.

[23] JM Blythe and SD Johnson. 2018. The Consumer Security Index for IoT: A protocol for developing an index to improve consumer decision making and to incentivize greater security provision in IoT devices. (2018).

[24] Tuhin Borgohain, Uday Kumar, and Sugata Sanyal. 2015. Survey of Operating Systems for the IoT Environment. *CoRR* abs/1504.02517 (2015).

[25] M Botterman. 2009. For the European Commission Information Society and Media Directorate General. In *Networked Enterprise & RFID Unit–D4, Internet of Things: An Early Reality of the Future Internet, Report of the Internet of Things Workshop, Prague, Czech Republic*.

[26] Armir Bujari, Marco Furini, Federica Mandreoli, Riccardo Martoglia, Manuela Montangero, and Daniele Ronzani. 2018. Standards, Security and Business Models: Key Challenges for the IoT Scenario. *Mobile Networks and Applications* 23, 1 (01 Feb 2018), 147–154. https://doi.org/10.1007/s11036-017-0835-8

[27] Xiaodong Cao, Xilei Dai, and Junjie Liu. 2016. Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade. *Energy and buildings* 128 (2016), 198–213.

[28] Carriots. 2017. Carriots Pricing. (2017). https://www.carriots.com/pricing

[29] Marco Centenaro, Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi. 2016. Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios. *IEEE Wireless Communications* 23, 5 (2016), 60–67.

[30] Tej Bahadur Chandra, Pushpak Verma, and A. K. Dwivedi. 2016. Operating Systems for Internet of Things: A Comparative Study. In *Proceedings of the*

Second International Conference on Information and Communication Technology for Competitive Strategies (ICTCS '16). ACM, New York, NY, USA, Article 47, 6 pages. https://doi.org/10.1145/2905055.2905105

[31] Khyati Chourasia, Anubhuti Khare, Manish Saxena, and Roshan Singh Thakur. 2012. Conserving Energy in 3G and Study of 4G Cellular Networks. (2012).

[32] Daniel Cid. [n. d.]. Large CCTV Botnet Leveraged in DDoS Attacks. ([n. d.]). https://blog.sucuri.net/2016/06/large-cctv-botnet-leveraged-ddos-attacks.html

[33] Cisco. 2015. Internet of Things Will Deliver $1.9 Trillion Boost To Supply Chain And Logistics Operations. (2015). https://newsroom.cisco.com/press-release-content?articleId=1621819

[34] Mauro Conti, Nicola Dragoni, and Viktor Lesyk. 2016. A survey of man in the middle attacks. *IEEE Communications Surveys & Tutorials* 18, 3 (2016), 2027–2051.

[35] British Government: DDCMS. [n. d.]. Government response to the Secure by Design informal consultation. ([n. d.]). https://www.gov.uk/government/publications/secure-by-design/government-response-to-the-secure-by-design-informal-consultation

[36] British Government: DDCMS. 2018. *Code of Practice for Consumer IoT Security*. British Government.

[37] British Government: DDCMS. 2019. Open consultation: Consultation on regulatory proposals on consumer IoT security. (2019). https://www.gov.uk/government/consultations/consultation-on-regulatory-proposals-on-consumer-iot-security

[38] S. Dharur, C. Hota, and K. Swaminathan. 2017. Energy efficient IoT framework for Smart Buildings. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. 793–800. https://doi.org/10.1109/I-SMAC.2017.8058288

[39] Dyn. 2016. Dyn Analysis Summary Of Friday October 21 Attack. (2016). https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/

[40] Ericsson. 2016. Ericsson Mobility Report 2016. *Ericsson, Stockholm, Sweden, Tech. Rep. EAB-16* (2016).

[41] Internationale Fernmeldeunion. 2012. ITU-T Y. 4000/Y. 2060 (06/2012). *Overview of the Internet of things* (06 2012).

[42] Gartner. 2018. Internet of Things Definition. (2018). www.gartner.com/it-glossary/Internet-of-things/

[43] Neil Gershenfeld and JP Vasseur. 2014. As objects go online; the promise (and pitfalls) of the Internet of Things. *Foreign Aff.* 93 (2014), 60.

[44] Ibrahim Ghafir, Jibran Saleem, Mohammad Hammoudeh, Hanan Faour, Vaclav Prenosil, Sardar Jaf, Sohail Jabbar, and Thar Baker. 2018. Security threats to critical infrastructure: the human factor. *The Journal of Supercomputing* (2018), 1–17.

[45] Google. 2017. Google Cloud Internet of Things Core Pricing. (2017). https://cloud.google.com/iot/pricing

[46] Google. 2017. Overview of Internet of Things. (2017). https://cloud.google.com/solutions/iot-overview

[47] J. Granjal, E. Monteiro, and J. S Silva. 2015. Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues. *IEEE Communications Surveys Tutorials* 17, 3 (thirdquarter 2015), 1294–1312. https://doi.org/10.1109/COMST.2015.2388550

[48] GS1. 2017. GS1 And The Internet of Things. (2017). https://www.gs1.org/sites/default/files/images/standards/internet-of-things/gs1-and-the-internet-of-things-iot.pdf

[49] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems* 29, 7 (2013), 1645–1660.

[50] Brij Gupta, Dharma P Agrawal, and Shingo Yamaguchi. 2016. *Handbook of research on modern cryptographic solutions for computer and cyber security.* IGI global.

[51] Brij B Gupta. 2018. *Computer and cyber security: principles, algorithm, applications, and perspectives.* CRC Press.

[52] Asif Habib. 2008. Sensor network security issues at network layer. In *Advances in Space Technologies, 2008. ICAST 2008. 2nd International Conference on*. IEEE, 58–63.

[53] D. He, R. Ye, S. Chan, M. Guizani, and Y. Xu. 2018. Privacy in the Internet of Things for Smart Healthcare. *IEEE Communications Magazine* 56, 4 (APRIL 2018), 38–44. https://doi.org/10.1109/MCOM.2018.1700809

[54] H. Hejazi, H. Rajab, T. Cinkler, and L. Lengyel. 2018. Survey of platforms for massive IoT. In *2018 IEEE International Conference on Future IoT Technologies (Future IoT)*. 1–8. https://doi.org/10.1109/FIoT.2018.8325598

[55] IBM. 2017. IoT Cost Calculator. (2017). https://console.bluemix.net/pricing/platform/internet$_o f_t hings$

[56] IEEE. 2014. Special Report: The Internet of Things. *IEEE The Institute* March (2014). http://theinstitute.ieee.org/static/special-report-the-internet-of-things

[57] M Imani, M Taheri, ME Rajabi, and M Naderi. 2010. Vulnerabilities in network layer at wireless mesh networks (WMNs). In *Educational and Network Technology (ICENT), 2010 International Conference on*. IEEE, 487–492.

[58] Incapsula. 2016. Breaking Down Mirai: An IoT DDoS Botnet Analysis. (2016). https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html

[59] INFSO Networked Enterprise & RFID INFSO. 2008. Internet of Things in 2020, Roadmap for the Future, Version 1.1, Authors: INFSO D.4 Networked Enterprise & RFID INFSO G.2 Micro & Nanosystems in co-operation with the RFID Working Group Of EPOSS. (2008).

[60] Next Generation Internet Initiative. [n. d.]. Next Generation Internet - Vision. ([n. d.]). https://www.ngi.eu/vision/

[61] International Energy Agency. 2016. *World Energy Balances 2016*. OECD.

[62] IoT Analytics. 2017. List Of 450 IoT Platform Companies. (2017). https://IoT-analytics.com/product/list-of-450-IoT-platform-companies/

[63] ISO/IEC/IEEE. 2011. ISO/IEC/IEEE Systems and software engineering – Architecture description. *ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)* (Dec 2011), 1–46. https://doi.org/10.1109/IEEESTD.2011.6129467

[64] ITU. 2018. ITU Work Program [2017-2020]:[SG20]:[Q1/20]. (2018). https://www.itu.int/ITU-T/workprog/wp_item.aspxisn=13670

[65] ITU. 2018. ITU Work Program [2017-2020]:[SG20]:[Q3/20]. (2018). https://www.itu.int/ITU-T/workprog/wp_item.aspxisn=14126

[66] ITU. 2018. ITU Work Program [2017-2020]:[SG20]:[Q3/20]. (2018). https://www.itu.int/ITU-T/workprog/wp_item.aspxisn=14650

[67] ITU. 2018. ITU Work Program [2017-2020]:[SG20]:[Q6/20]. (2018). https://www.itu.int/ITU-T/workprog/wp_item.aspxisn=14656

[68] ITU. 2018. ITU Work Program [2017-2020]:[SG20]:[Q6/20]. (2018). https://www.itu.int/ITU-T/workprog/wp_item.aspxisn=14318

[69] ITU. 2018. ITU Work Program [2017-2020]:[SG20]:[Q7/20]. (2018). https://www.itu.int/ITU-T/workprog/wp_item.aspxisn=14949

[70] Mike Jenks, Elizabeth Burton, and Katie Williams. 1996. A sustainable future through the compact city? Urban intensification in the United Kingdom. *Environment by Design* 1, 1 (1996), 5–20.

[71] Ming Jin, Nikolaos Bekiaris-Liberis, Kevin Weekly, Costas Spanos, and Alexandre Bayen. 2015. Sensing by proxy: Occupancy detection based on indoor CO2 concentration. *UBICOMM 2015* 14 (2015).

[72] Vasileios Karagiannis, Periklis Chatzimisios, Francisco Vazquez-Gallego, and Jesus Alonso-Zarate. 2015. A survey on application layer protocols for the Internet of Things. *Transaction on IoT and Cloud Computing* 3, 1 (2015), 11–17.

[73] Chris Karlof and David Wagner. 2003. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad hoc networks* 1, 2-3 (2003), 293–315.

[74] Sarvesh Kumar, Suraj Pal Singh, A Kumar Singh, and Jahangir Ali. 2013. Virtualization, the great thing and issues in cloud computing. *International Journal of Current Engineering and Technology* 3 (2013).

[75] LB Landauer. 2001. The History, Development, and Importance of Personal Computers. *Science and Its Times: Understanding the Social Significance of Scientific Discovery* 7 (2001), 536–540.

[76] Christian Ostergaard Laursen. 2017. Internet of Things: A Comparison of Communication Technologies. (2017). https://blog.montem.io/2017/03/10/Internet-of-things-a-comparison-of-communication-technologies/

[77] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. 2017. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal* 4, 5 (Oct 2017), 1125–1142. https://doi.org/10.1109/JIoT.2017.2683200

[78] SW Lin, B Miller, J Durand, G Bleakley, A Chigani, R Martin, B Murphy, and M Crawford. 2017. IIC: The industrial internet of things volume G1: reference architecture. *Industrial Internet Consortium* (2017), 10–46.

[79] James Manyika, Michael Chui, Peter Bisson, Jonathan Woetzel, Richard Dobbs, Jacques Bughin, and Dan Aharon. 2015. Unlocking the Potential of the Internet of Things. *McKinsey Global Institute* (2015).

[80] Johan Marconot, Florian Pebay-Peyroula, and David Hély. 2017. IoT Components LifeCycle Based Security Analysis. In *2017 Euromicro Conference on Digital System Design (DSD)*. IEEE, 295–298.

[81] IHS Markit and Sam Lucero. 2016. IoT platforms: Enabling the Internet of Things. *HIS Technology* March (2016).

[82] Maria-Lluïsa Marsal-Llacuna, Joan Colomer-Llinàs, and Joaquim Meléndez-Frigola. 2015. Lessons in urban monitoring taken from sustainable and livable cities to better address the Smart Cities initiative. *Technological Forecasting and Social Change* 90 (2015), 611–622.

[83] C. A. Medina, M. R. Prez, and L. C. Trujillo. 2017. IoT Paradigm into the Smart City Vision: A Survey. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 695–704. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2017.109

[84] Mendix. 2017. Collaborative Visual Development. (2017). https://www.mendix.com/collaborative-visual-development/

[85] Wu Miao, Lu Ting-Jie, Ling Fei-Yang, Sun Jing, and Du Hui-Ying. 2010. Research on the architecture of Internet of Things. In *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, Vol. 5. V5–484–V5–487. https://doi.org/10.1109/ICACTE.2010.5579493

[86] Microsoft Azure. 2017. Pricing Calculator. (2017). https://azure.microsoft.com/en-gb/pricing

[87] Roberto Minerva, Abyi Biru, and Domenico Rotondi. 2015. Towards a definition of the Internet of Things (IoT). *IEEE Internet Initiative* 1 (2015).

[88] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. 2012. Internet of Things: Vision, applications and research challenges. *Ad Hoc Networks* 10, 7 (2012), 1497 – 1516. http://www.sciencedirect.com/science/article/pii/S1570870512000674

[89] Alhasan A. Alharbi Mohammed Tawfik, Ali M. Almadni. 2017. A Review: the Risks And weakness Security on the IoT. *IOSR Journal of Computer Engineering* 1 (2017), 12–17.

[90] Manuela Montangero. 2017. IoT: Science Fiction or Real Revolution?. In *Smart Objects and Technologies for Social Good: Second International Conference, GOODTECHS 2016, Venice, Italy, November 30–December 1, 2016, Proceedings*, Vol. 195. Springer, 96.

[91] Bill Morelli. 2013. Internet Connected Devices: Evolving from the "Internet of Things to the "Internet of Everything. *IHS Website* (2013). https://www.ihs.com/pdf/IHS-IoT-Evolution_161384110915583632.pdf

[92] Max HA Newman. 1955. Alan Mathison Turing. 1912-1954. *Biographical memoirs of Fellows of the Royal Society* 1 (1955), 253–263.

[93] NodeRED. 2017. NodeRED. (2017). https://nodered.org/

[94] K. E. Nolan, Y. Guibene, and M. Y. Kelly. 2016. An evaluation of low power wide area network technologies for the Internet of Things. In *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*. 439–444. https://doi.org/10.1109/IWCMC.2016.7577098

[95] Adegboyega Ojo, Edward Curry, Tomasz Janowski, and Zamira Dzhusupova. 2015. *Designing Next Generation Smart City Initiatives - The SCID Framework*. https://doi.org/10.1007/978-3-319-03167-5_4

[96] Oracle. 2017. Oracle Integration Cloud Pricing. (2017). https://oracle.com/

[97] N Oyj. 2016. LTE evolution for IoT connectivity. *Nokia Corporation White Paper* (2016).

[98] S Prabhakar. 2017. Network Security in Digitalization: Attacks and Defence. *Int. J. Res. Comput. Appl. Robot* 5, 5 (2017), 46–52.

[99] Deepak Puthal, Surya Nepal, Rajiv Ranjan, and Jinjun Chen. 2016. Threats to networking cloud and edge datacenters in the Internet of Things. *IEEE Cloud Computing* 3, 3 (2016), 64–71.

[100] Tariq Aziz Rao. 2018. Security Challenges Facing IoT Layers and its Protective Measures. *International Journal of Computer Applications* 179 (2018), 8887–8921.

[101] Partha Pratim Ray. 2016. A survey of IoT cloud platforms. *Future Computing and Informatics Journal* 1, 1 (2016), 35 – 46. http://www.sciencedirect.com/science/article/pii/S2314728816300149

[102] C. Sabri, L. Kriaa, and S. L. Azzouz. 2017. Comparison of IoT Constrained Devices Operating Systems: A Survey. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. 369–375. https://doi.org/10.1109/AICCSA.2017.187

[103] Jibran Saleem, Bamidele Adebisi, Ruth Ande, and Mohammad Hammoudeh. 2017. A state of the art survey-Impact of cyber attacks on SME's. In *Proceedings of the International Conference on Future Networks and Distributed Systems*. ACM, 52.

[104] Mattia Salnitri, Mahdi Alizadeh, Daniele Giovanella, Nicola Zannone, and Paolo Giorgini. 2018. From Security-by-Design to the Identification of Security-Critical Deviations in Process Executions. In *International Conference on Advanced Information Systems Engineering*. Springer, 218–234.

[105] Samsung. 2017. Pricing Plans. (2017). https://artik.cloud/pricing/

[106] Ameya Sanzgiri and Dipankar Dasgupta. 2016. Classification of insider threat detection techniques. In *Proceedings of the 11th annual cyber and information security research conference*. ACM, 25.

[107] Pallavi Sethi and Smruti R Sarangi. 2017. Internet of Things: Architectures, Protocols, and Applications. *Journal of Electrical and Computer Engineering* 2017 (2017).

[108] Vladimir Shakhov and Insoo Koo. 2018. Depletion-of-Battery Attack: Specificity, Modelling and Analysis. *Sensors* 18, 6 (2018), 1849.

[109] Vladimir V Shakhov. 2013. Protecting wireless sensor networks from energy exhausting attacks. In *International Conference on Computational Science and Its Applications*. Springer, 184–193.

[110] Zhengguo Sheng, Shusen Yang, Yifan Yu, Athanasios Vasilakos, Julie Mccann, and Kin Leung. 2013. A survey on the ietf protocol suite for the Internet of Things: Standards, challenges, and opportunities. *IEEE Wireless Communications* 20, 6 (2013), 91–98.

[111] Santosh Singh. 2018. Top 20 IoT Platforms in 2018. (2018). https://Internetofthingswiki.com/top-20-IoT-platforms/

[112] Yashaswi Singh, Farah Kandah, and Weiyi Zhang. 2011. A secured cost-effective multi-cloud storage in cloud computing. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 619–624.

[113] Ian Skerrett. 2016. Three Software Stacks Required for the Internet of Things (IoT). (2016). https://www.linkedin.com/pulse/three-software-stacks-required-internet-things-iot-eclipse-das

[114] Phil Smith. 2017. Comparing Low-Power Wireless Technologies. (2017). https://www.digikey.co.uk/en/articles/techzone/2017/oct/comparing-low-power-wireless-technologies

[115] Christos Stergiou, Kostas E Psannis, Byung-Gyu Kim, and Brij Gupta. 2018. Secure integration of IoT and cloud computing. *Future Generation Computer Systems* 78 (2018), 964–975.

[116] Sowmya Nagasimha Swamy, Dipti Jadhav, and Nikita Kulkarni. 2017. Security threats in the application layer in IoT applications. In *I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2017 International Conference on*. IEEE, 477–480.

[117] Jorg Swetina, Guang Lu, Philip Jacobs, Francois Ennesser, and JaeSeung Song. 2014. Toward a standardized common M2M service layer platform: Introduction to oneM2M. *IEEE Wireless Communications* 21, 3 (2014), 20–26.

[118] Techtarget. 2016. The essential guide to supply chain management best practices. (2016). http://Internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

[119] Nadeem Unuth. 2017. What Is the Definition of 3G Wireless Technology? Technical Specifications of 3G. (2017). https://www.lifewire.com/what-is-3g-3426465

[120] Jeffrey Voas. 2016. Networks of things. *NIST Special Publication* 800, 183 (2016), 800–183.

[121] J Voas. 2018. NIST: Internet of Things (IoT) Trust Concerns. (2018).

[122] H Wang and A O Fapojuwo. 2017. A Survey of Enabling Technologies of Low Power and Long Range Machine-to-Machine Communications. *IEEE Communications Surveys Tutorials* 19, 4 (Fourthquarter 2017), 2621–2639. https://doi.org/10.1109/COMST.2017.2721379

[123] Wikipedia. 2017. ANT (network). (2017). https://en.wikipedia.org/wiki/ANT$_{(network)}$

[124] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao. 2017. A Survey on Security and Privacy Issues in Internet-of-Things. *IEEE Internet of Things Journal* 4, 5 (Oct 2017), 1250–1258. https://doi.org/10.1109/JIOT.2017.2694844

[125] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani. 2017. Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges. *IEEE Wireless Communications* 24, 3 (June 2017), 10–16. https://doi.org/10.1109/MWC.2017.1600421

[126] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. 2014. Internet of Things for Smart Cities. *IEEE Internet of Things Journal* 1, 1 (Feb 2014), 22–32. https://doi.org/10.1109/JIoT.2014.2306328

[127] C. L. Zhong, Z. Zhu, and R. G. Huang. 2015. Study on the IoT Architecture and Gateway Technology. In *2015 14th International Symposium on Distributed*

Computing and Applications for Business Engineering and Science (DCABES). 196–199. https://doi.org/10.1109/DCABES.2015.56

A.1.2 Conference Paper

In Proceedings of International Conference on Sustainability in Energy and Buildings, Croatia, September 2020,

Sustainability in Energy and Buildings Journal,Vol. 7.

Energy Monitoring Solution for SMEs

Ruth Ande[1], Bamidele Adebisi[1],
Mohammad Hammoudeh[2], and Austin Ikpehai[3]

[1] School of Engineering, Manchester Metropolitan University, UK,
[2] School of Computing, Mathematics & Digital Technology,
Manchester Metropolitan University, UK
[3] School of Engineering, Sheffield Hallam University, UK

Abstract. The complexity and high cost of building retrofitting for improved energy performance can be overwhelming for many SMEs. Tailor-made frameworks, are therefore, required to deliver long term energy reduction benefits, for relatively small commercial buildings. This paper presents a low cost energy monitoring and reporting solution for SMEs, which includes a system architecture, a baseline data generation strategy that significantly reduces the retrofitting timeline and a sensor network strategy that incorporates existing ICT infrastructure and minimises the number of IoT sensors. The system reports the energy monitoring data to building users in real time in an easy to understand format allowing building users to quickly analyse the affect of changes in their energy behaviour, encouraging them to try different low cost energy reduction strategies, before choosing more expensive solutions.

Keywords: IoT, data fusion, sensor, sensor node, energy consumption, energy reduction, buildings, energy monitoring

1 Introduction

This work proposes an energy monitoring solution for micro, small or medium sized enterprises (SMEs), developed to help such organisations overcome barriers faced when trying to reduce building energy consumption. According to the Department for Business, Energy and Industrial Strategy (BEIS), in 2018, the UK's energy consumption was 143MTOE, where 28.9% was consumed by the domestic sector, 39.9% by the transport sector and 31.2% by the industry and service sector [1]. Buildings consumed 40.3%, with commercial buildings responsible for 49.4%, of which, 62% was for lighting and heating. Due to the significant cost of operating commercial buildings, there is a clear link between the size of an enterprise and the size of building they occupy. This is illustrated by a correlation in the energy consumed by small commercial buildings and by SMEs. In 2015, there were 5.38M UK businesses, 99% were SMEs [2], responsible for consuming 48.4% of commercial energy. Simultaneously, the commercial building stock included 1.57M buildings, of which 92% were classified as small buildings; 1,000 m^2 or less [2], and consumed 35.6% of all commercial energy.

Generally, SMEs are defined as enterprises with 0-249 employees. The energy consumption of enterprises based on employee numbers and building size are shown in Figure 1, created with data from [1]. The graph shows that for retail, office and hospitality, there is a clear correlation between energy consumption, enterprise size and building size. For the industrial sector, building size and energy consumption is instead dependant upon industrial equipment and processes. Similarly within the leisure sector some public buildings, including theatres and art galleries, the building size and energy consumption are more dependant upon size of audience or community served.

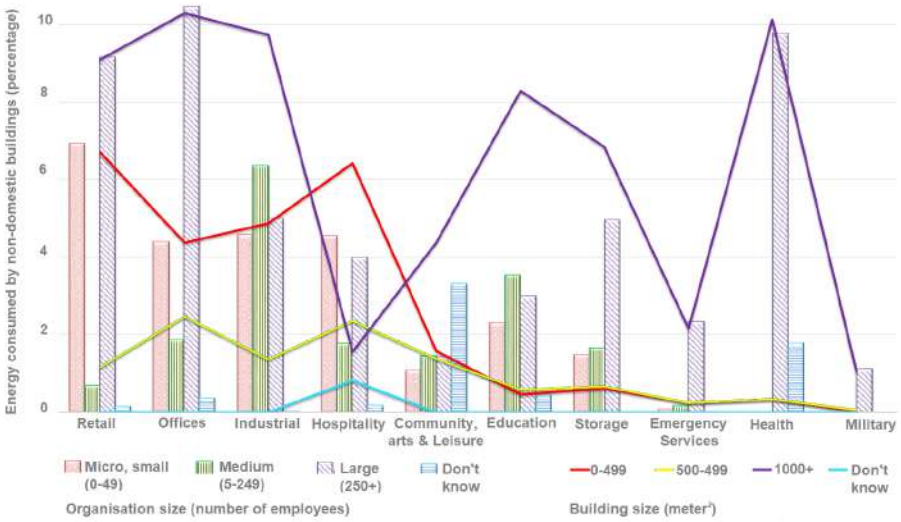


Fig. 1. Energy consumption based organisation size and building size

2 Related Work

The Health & Safety Executive's (HSE) 1992 Workplace Regulations [3] defines a minimum floor space of 3.67 - 4.58m^2 per employee. So, a micro enterprise with 1-9 employees, could occupy a building 3.67 - 33m^2, plus toilets. The Total Office Cost Survey 2019 [4] values an individual work area at £5,408 - £18,988 annually (based on Norwich and London) using an individual area 2.5 times larger than HSE requirements. Using HSE regulations, the surveyed buildings would cost £2,130 - £7,478 per person. SMEs are sensitive to overhead costs, so will tend towards supplying employees with the minimal working area and occupy commercial spaces which are as small and low cost as possible.

Worldwide, building energy consumption is rising, but research shows that energy saving initiatives that include monitoring consumption and giving feedback to users can achieve long term energy reductions [5]. Feedback from surveyed SMEs [6] [7] [8] shows that barriers to the uptake of energy reduction initiatives include high upfront or lifetime costs, significant time commitments

and expertise. Many commercial building energy monitoring and reporting systems do exist, but these systems are inaccessible to SMEs because they generally fit into three categories, being complex and expensive, affordable but requiring expertise to fit and manage, or affordable but overly simple and ineffective.

Previous research carried out in this area includes a number of surveys, such as [6] which audited 280 SMEs, [7] used data from energy audits to propose solutions to overcome SME energy reduction barriers, and [8] studied the gap between energy reduction measures and their implementation. Developmental work includes management of energy demand [9] using energy disaggregation techniques to turn off appliances, [10] developed an SME monitoring and targeting plan based on utility bill analysis, and [11] a monitoring system specifically for SMEs within manufacturing. These works highlight the barriers that SMEs face and offer a number of specific solutions but a solution suitable for a wide range of SMEs has not be offered.

Based on lesson learned, this paper presents a building monitoring and reporting solution designed specifically for SMEs based in small commercial buildings, to accurately monitor and report real-time energy usage. The system measures energy consumption and additional features including weather, occupant presence and occupant comfort levels. The data is analysed and fed back to the building users in an understandable format with the aim of encouraging users to test small changes to quickly and significantly reduce energy consumption, with no cost, or low costs to the SME.

3 Proposed System Specifications

Energy consumption is dependent upon many conditions and to create a complete picture for building users the system needs to measure energy usage, referred to as 'metering', and the additional conditions referred to as 'monitoring'.

3.1 Baseline Generation Strategy

For a monitoring and reporting system to indicate a reduction in energy consumption, a baseline is required, this is the measurement of normal energy consumption before energy saving initiatives are applied. Traditionally, this comprises 18 months of metering and monitoring data. The advantage of this method is data completeness since it is generated using the same method as the data that it will be compared against. The disadvantage, particularly for SMEs, is the long lead time before implementing ESMs. Recent research from a study carried out in Manchester on behalf of the 'Boosting access for SMEs to energy efficiency' (BASEE) competition, this significant lead time can cause SMEs to disengage, resulting in a loss of company interest or finances. Instead, the proposed baseline generation strategy reduces the lead time to 3 months. The strategy is to create baseline data using metering/monitoring data collected for a period of 3 months, and combine it with building model data. Research shows building models can be inaccurate [12] due to significant differences between the building model and the real building. To improve the model, indirect building data will be added:

- pre-existing building energy performance specifications e.g. post occupancy evaluation (POE) and display energy certification (DEC),
- 12+ months' historical billing data,
- normal building usage information e.g. operating hours, occupancy levels,
- energy performance data collected from similar smart buildings,
- data collected from pre-installed sensors, security systems or BMS

This improved model can deliver consumption predictions within 3% of actual [12]. Additionally, during the baseline monitoring period, energy consumption profiling can be completed to determine where energy is consumed to inform the SME which additional ESMs may be most effective.

3.2 Proposed System Architecture

As discussed in the introduction, the cost of monitoring systems and their complexity can be prohibitive to SMEs, some lower cost solutions are available such as the Beringar IoT building resource and occupancy tracking module [13] costing £450 per module for the hardware, with an annual subscription of £350 for software-as-a-service data reporting. The Pressac smart monitoring hardware [14] costs £750 each and the Smart Citizen starter hardware [15] costs £100 each and both solutions offer a range of IoT sensors but require additional expertise to install, configure and manage. The Pressac kits include a Wi-Fi, Ethernet and LTE gateway but do not include software for reporting, instead they are compatible with commercial cloud platforms. The Smart Citizen kits are open source and connect to a range of open source reporting tools. Alternate low cost solutions typically include installation of motion detectors to automate lights and other services but this is generally overly simple and ineffective.

Learning from previous work [6] [7] and existing systems [13] [15], the proposed solution is simple enough that a user can set the system up straight out of the box, connecting pre-installed smart meters and other pre-installed systems using the existing WiFi network. Additional IoT sensor nodes can then be fitted around the building and connected, also using the WiFi network. The objectives of the proposed architecture are to:

1. accurately measure energy consumption,
2. accurately and reliably monitor the building,
3. use pre-installed ICT infrastructures such as WiFi, smart meters, BMS etc,
4. use minimal additional communication / metering / monitoring equipment,
5. feedback real-time understandable information to building users to enable changes in energy usage behaviour,
6. keep system simple for easy installation, set-up and maintenance,
7. keep system costs to a minimum.

The proposed architecture is shown in Figure 2 and is split into 3 sections; the monitoring system, communication network and reporting system. The monitoring system is comprised of the energy metering and the building monitoring.

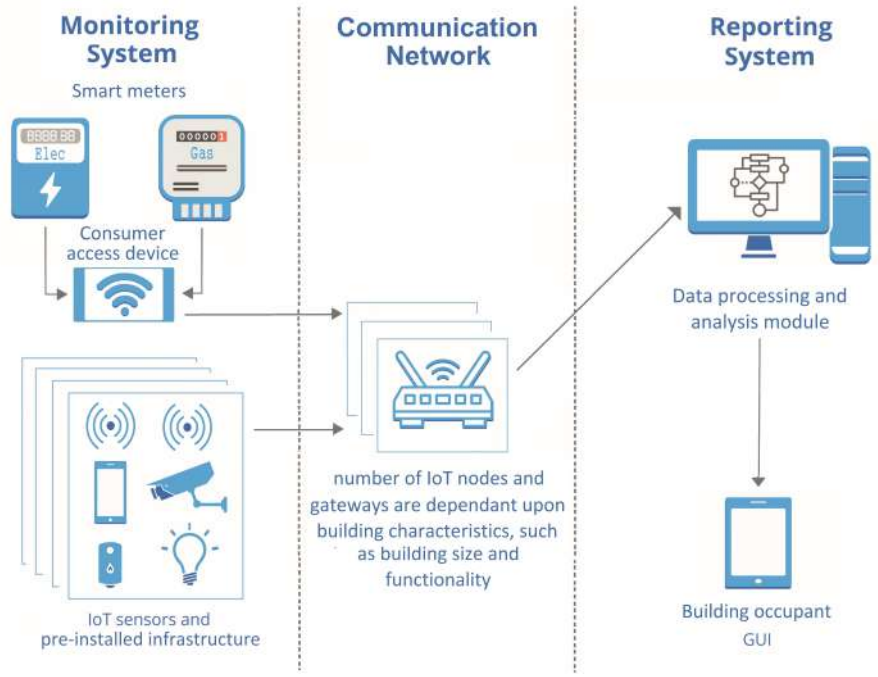


Fig. 2. Proposed System Architecture

The energy metering will be completed by the smart metering equipment technical specifications version 2 (SMET2) gas and electricity smart meters. These meters are available without charge for SMEs from their utility company. The metering data from the SMET2 meters will be collected by a consumer access device (CAD) and transmitted to the processing and analysis module.

As discussed in the introduction, traditional building monitoring systems are expensive mainly due to high specification hardware. Inline with the low cost objective of the proposed solution, it has been designed to incorporate any previously installed building monitoring systems for example camera, heat imaging or RFID based security systems, clocking-in systems or BMS etc. The proposed system will combine data from these pre-installed systems with data collected by additional low-cost IoT sensor nodes. All of the data will be transmitted using the existing building WiFi network to the data processing and analysis module. Next the data will be processed and combined using data fusion techniques and analysed to determine if occupants are within specific building zones and if occupant comfort levels are being achieved, such as room temperature and light levels. The number of WiFi gateways will be kept to a minimum, dependant upon the size and layout of the building. In order to keep system costs to a minimum, data processing and analysis will carried out locally and isolated from the internet. Where users want a cloud-based solution, this will be available for an

additional cost, the cloud-based solution will have the necessary security module embedded. The building monitoring results will be delivered to the building users in real time in the form of a reporting application which users can view on a computer, tablet or smartphone.

3.3 Sensor Network Strategy

Sensors are required to adequately monitor the features affecting energy consumption. Traditionally, building monitoring is achieved by installing a large number of IoT sensors around a building, where each IoT sensor includes a sensor, communication unit and power supply. This work proposes an alternative sensor network strategy which reduces installation time, installation complexity, the amount of repeated hardware, operating power and system costs. The strategy is implemented using accurate low cost sensors which have a large sensor range. Multiple heterogeneous sensors will be combined into a single sensor node which will also include a WiFi communication unit and power supply, resulting in a single sensor node which can monitor multiple features across a wide area.

The proposed sensor nodes have a combined circular coverage area, shown as the blue circle in Figure 3. The coverage range of the node is the coverage radius labelled c in the figure. If a square, of maximum dimensions, is drawn inside the circular coverage area, shown in black, an equilateral triangle can also be drawn, shown in green in Figure 3. Using Pythagoras' Theorem, the size of the node coverage square can be determined. Labels a, b, c denote the equilateral triangle:

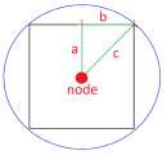


Fig. 3. IoT node coverage area

$$c^2 = a^2 + b^2 = 2a^2 \quad (1)$$

$$a = \frac{c}{\sqrt{2}} \quad (2)$$

$$L_{\text{square}} = 2a = \frac{2c}{\sqrt{2}} = c\sqrt{2} \quad (3)$$

$$A_{\text{square}} = (L_{\text{square}})^2 = 2c^2 \quad (4)$$

To create reliable monitoring data, the sensor network strategy must offer complete coverage of the building. As such, the strategy defines the sensor coverage square, shown in black in Figure 3, as its coverage area. Additionally, the strategy stipulates that the sensor nodes should be positioned such that each node coverage square should either meet a building wall or a neighbouring node coverage square. Where two neighbouring node coverage squares meet, overlap should be kept to a minimum to reduce the number of nodes required. The proposed nodes have been designed with a coverage range of 7m, and the length of the node coverage square can be calculated as 9.9m.

Using the sensor network strategy, building size and layout, the required number of sensor nodes can be calculated. Typically, for an open plan building, less nodes will be required since the functionality of the sensor nodes is inhibited by building walls. If a small open-plan commercial building 10m x 10m is considered, based on the HSE regulations [3], the building can accommodate up

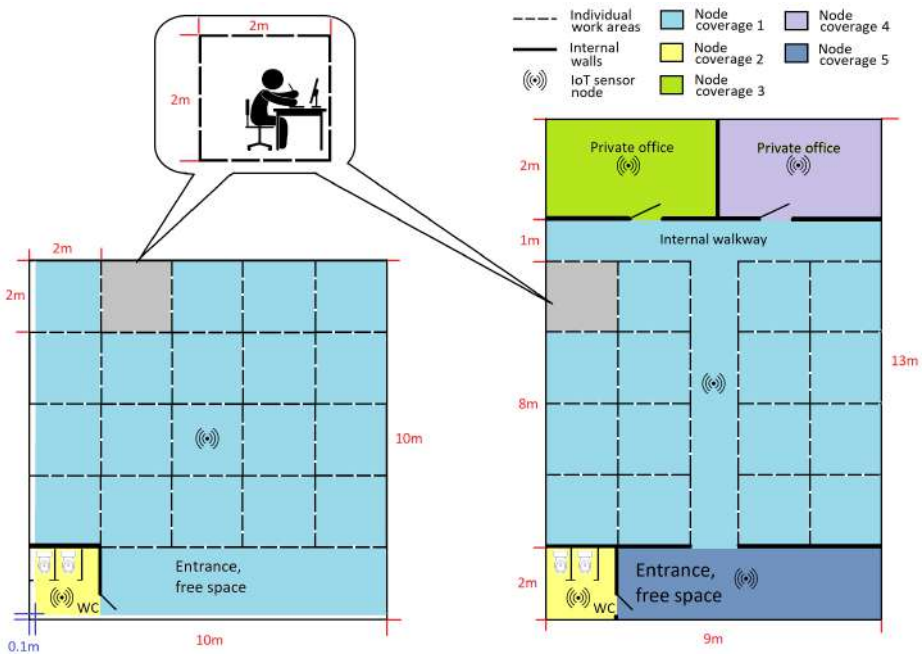


Fig. 4. A: 10m x 10m open plan building, B: 9m x 13m building with multiple areas

to 20 employees. Using the sensor network strategy, each node can cover an area up to 98m^2, so just 2 sensor nodes achieve full coverage of the building, one in the open plan area and a second for the toilet area, as shown in Figure 4a. A similar sized building, 9m x 13m, configured with multiple internal spaces would require 5 sensors nodes to achieve full building coverage, as shown in 4b.

4 Conclusion

This paper proposes a energy monitoring and reporting solution, which is specifically designed to overcome barriers SMEs face when trying to reduce energy consumption. Though monitoring and reporting systems do exist, research has shown that existing systems fit into three categories, either being too complex and expensive, affordable but requiring additional expertise to fit and manage, or affordable but overly simple and ineffective. The design of this energy monitoring and reporting solution makes it accessible to SMEs because it is low-cost, requires no expertise to install and operate and has a short lead time before reporting on improvements in energy behaviour, enabling SMEs to develop and test energy reduction strategies and quickly see the effects.

References

1. Department of Business, Energy Industrial Strategy, "Energy Consumption in the UK (ECUK) 2018 - National Statistics," July 2019.
2. House of Commons, "Briefing paper, number 06152, business statistics 2015," December 2015, accessed 2019-12-09. [Online]. Available: http://www.faseo.com/wp-content/uploads/2016/02/House-of-Commons-Business-Briefing.pdf
3. Health Safety Executive, Ed., *Workplace (Health, Safety and Welfare) Regulations 1992: Approved Code of Practice and guidance.* The Stationery Office, PO Box 29, Norwich NR3 1GN: The Stationery Offic, 2013.
4. L. S. Hampton, "Total office cost survey 2019," 2019, accessed 2019-10-10.
5. D. Violette, C. Mudd, and M. Keneipp, "An initial view on methodologies for emission baselines: Case study on energy efficiency," *IEA and OECD Information Papers*, 06 2000.
6. J. Fresner, F. Morea, C. Krenn, J. A. Uson, and F. Tomasi, "Energy efficiency in small and medium enterprises: Lessons learned from 280 energy audits across europe," *Journal of Cleaner Production*, vol. 142, pp. 1650 – 1660, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959652616319783
7. T. Fleiter, J. Schleich, and P. Ravivanpong, "Adoption of energy-efficiency measures in smes—an empirical analysis based on energy audit data from germany," *Energy Policy*, vol. 51, pp. 863 – 875, 2012, renewable Energy in China. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0301421512008166
8. K. Bunse, M. Vodicka, P. Schönsleben, M. Brülhart, and F. O. Ernst, "Integrating energy efficiency performance in production management – gap analysis between industrial needs and scientific literature," *Journal of Cleaner Production*, vol. 19, no. 6, pp. 667 – 679, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959652610004452
9. L. Hattam and D. V. Greetham, "Energy disaggregation for smes using recurrence quantification analysis," in *Proceedings of the Ninth International Conference on Future Energy Systems*, ser. e-Energy '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 610–617. [Online]. Available: https://doi.org/10.1145/3208903.3210280
10. J. Cosgrove, F. Doyle, F. Hardiman, and G. O'Farrell, "An approach to electricity monitoring and targeting (m&t) in irish precision engineering smes," in *International Conference on Sustainable Design and Manufacturing.* Springer, 2016, pp. 157–168.
11. J. Heilala, K. Klobut, T. Salonen, P. Siltanen, R. Ruusu, L. Urosevic, P. Reimer, A. Armijo, M. Sorli, T. Fatur, Z. Gantar, and A. Jung, "Ambient intelligence based monitoring and energy efficiency optimisation system," in *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)*, May 2011, pp. 1–6.
12. A. C. Menezes, A. Cripps, D. Bouchlaghem, and R. Buswell, "Predicted vs. actual energy performance of non-domestic buildings: Using post-occupancy evaluation data to reduce the performance gap," *Elsevier, Applied energy,*, vol. volume 97, pp. pg 355–364, 2012.
13. Beringar, "Creating clever space," 2019, accessed 2019-10-10. [Online]. Available: https://www.beringar.co.uk/product
14. Pressac, "Products," 2019, accessed 2019-10-10. [Online]. Available: https://www.pressac.com/buy
15. S. Studio, "Order your smart citizen kit," 2019, accessed 2019-10-10. [Online]. Available: https://smartcitizen.me

Bibliography

[1] Department of Business, Energy Industrial Strategy, *Energy Consumption in the UK (ECUK) 2019 - National Statistics*, Jul. 2019.

[2] House of Commons, *Briefing paper, number 06152, business statistics 2015*, Accessed 2019-12-09, Dec. 2015. [Online]. Available: http://www.faseo.com/wp-content/uploads/2016/02/House-of-Commons-Business-Briefing.pdf.

[3] D. Violette, C. Mudd, and M. Keneipp, "An initial view on methodologies for emission baselines: Case study on energy efficiency," *IEA and OECD Information Papers*, Jun. 2000.

[4] J. Fresner, F. Morea, C. Krenn, J. A. Uson, and F. Tomasi, "Energy efficiency in small and medium enterprises: Lessons learned from 280 energy audits across europe," *Journal of Cleaner Production*, vol. 142, pp. 1650–1660, 2017.

[5] T. Fleiter, J. Schleich, and P. Ravivanpong, "Adoption of energy-efficiency measures in smes—an empirical analysis based on energy audit data from germany," *Energy Policy*, vol. 51, pp. 863–875, 2012.

[6] K. Bunse, M. Vodicka, P. Schonsleben, M. Brulhart, and F. O. Ernst, "Integrating energy efficiency performance in production management – gap analysis between industrial needs and scientific literature," *Journal of Cleaner Production*, vol. 19, no. 6, pp. 667–679, 2011.

[7] Committee on Climate Change. (2019). "Net zero - the uk's contribution to stopping global warming," [Online]. Available: https://www.theccc.org.uk/wp-content/uploads/2019/05/Net-Zero-The-UKs-contribution-to-stopping-global-warming.pdf.

[8] J. Manyika, M. Chui, P. Bisson, J. Woetzel, R. Dobbs, J. Bughin, and D. Aharon, "Unlocking the potential of the internet of things," *McKinsey Global Institute*, 2015.

[9] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[10] Techtarget, "The essential guide to supply chain management best practices," 2016. [Online]. Available: http://Internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT.

[11] M. Botterman, "For the european commission information society and media directorate general," in *Networked Enterprise & RFID, Internet of Things: An Early Reality of the Future Internet, Report of the Internet of Things Workshop, Prague, Czech Republic*, 2009.

[12] J. Barber. (2017). "Getting answers to your iot questions," [Online]. Available: http:www.gartner.com/podcasts/getting-answers-to-your-IoT-questions/ (visited on 05/27/2017).

[13] IEA, "World energy balances 2016," [Online]. Available: http://dx.doi.org/10.1787/9789264263116-en/content/book/9789264263116-en.

[14] The International Energy Agency. (2020). "Tracking buildings 2020 iea report," [Online]. Available: https://www.iea.org/reports/tracking-buildings-2020.

[15] UNFCCC. (2016). "What is the paris agreement," [Online]. Available: https://unfccc.int/process-and-meetings/the-paris-agreement/what-is-the-paris-agreement.

[16] Department of Business, Energy Industrial Strategy, *Energy consumption in the uk (ecuk) 2020 - national statistics*, Jul. 2020.

[17] The Institute for Government. (2020). "Net zero: How government can meet its climate change target," [Online]. Available: https://www.instituteforgovernment.org.uk/sites/default/files/publications/net-zero-government-climate-change-target.pdf.

[18] The Institute for Government Think Tank. (2020). "Summary of net zero: How government can meet its climate change target," [Online]. Available: https://www.instituteforgovernment.org.uk/publications/net-zero.

[19] Committee on Climate Change. (2020). "Reducing uk emissions: 2020 progress report to parliament," [Online]. Available: https://www.theccc.org.uk/wp-content/uploads/2020/06/Reducing-UK-emissions-Progress-Report-to-Parliament-Committee-on-Cli..._-002-1.pdf.

[20] Y.-J. Zhang, Y.-F. Sun, and J. Huang, "Energy efficiency, carbon emission performance, and technology gaps: Evidence from cdm project investment," *Energy Policy*, vol. 115, pp. 119–130, 2018.

[21] E. Gelenbe and Y. Caseau, "The impact of information technology on energy consumption and carbon emissions," *Ubiquity*, vol. 2015, no. June, pp. 1–15, 2015.

[22] "Ict and environmental sustainability: A global perspective," *Telematics and Informatics*, vol. 34, no. 4, pp. 85–95, 2017.

[23] D. Minoli, K. Sohraby, and B. Occhiogrosso, "Iot considerations, requirements, and architectures for smart buildings—energy optimization and next-generation building management systems," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 269–283, 2017.

[24] B. Balaji, J. Xu, A. Nwokafor, R. Gupta, and Y. Agarwal, "Sentinel: Occupancy based hvac actuation using existing wifi infrastructure within commercial buildings," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, 2013, pp. 1–14.

[25] Premier Farnell Limited. (2020). "All products," [Online]. Available: `https://uk.farnell.com`.

[26] K. Weekly, M. Jin, H. Zou, C. Hsu, C. Soyza, A. Bayen, and C. Spanos, "Building-in-briefcase: A rapidly-deployable environmental sensor suite for the smart building," *Sensors*, vol. 18, no. 5, p. 1381, 2018.

[27] H. Zou, Y. Zhou, J. Yang, and C. J. Spanos, "Device-free occupancy detection and crowd counting in smart buildings with wifi-enabled iot," *Energy and Buildings*, vol. 174, pp. 309–322, 2018.

[28] A. Beltran, V. L. Erickson, and A. E. Cerpa, "Thermosense: Occupancy thermal based sensing for hvac control," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, 2013, pp. 1–8.

[29] G. J. Levermore, *Building Energy Management Systems: Applications to low-energy HVAC and natural ventilation control*. Taylor & Francis, 2000.

[30] W. Tushar, N. Wijerathne, W.-T. Li, C. Yuen, H. V. Poor, T. K. Saha, and K. L. Wood, "Internet of things for green building management: Disruptive innovations through low-cost sensor technology and artificial intelligence," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 100–110, 2018.

[31] X. Yu and H. Guo, "A survey on iiot security," in *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, IEEE, 2019, pp. 1–5.

[32] H. N. Rafsanjani, A. Ghahramani, and A. H. Nabizadeh, "Isea: Iot-based smartphone energy assistant for prompting energy-aware behaviors in commercial buildings," *Applied Energy*, vol. 266, 2020.

[33] H. N. Rafsanjani and A. Ghahramani, "Extracting occupants' energy-use patterns from wi-fi networks in office buildings," *Journal of Building Engineering*, vol. 26, p. 100 864, 2019.

[34] Health Safety Executive, Ed. (2020). "Human factors: Lighting, thermal comfort, working space, noise and vibration." Accessed 2019-10-10, [Online]. Available: `https://www.hse.gov.uk/humanfactors/topics/lighting.htm`.

[35] J. Ahmad, H. Larijani, R. Emmanuel, M. Mannion, and A. Javed, "Occupancy detection in non-residential buildings – a survey and novel privacy preserved occupancy monitoring solution," *Applied Computing and Informatics*, 2018, ISSN: 2210-8327.

[36] S. Bergen Henegouwen, "Create wellbeing by performance in office buildings: Permanent wellbeing with infinite material flows," 2020.

[37] K. Akkaya, I. Guvenc, R. Aygun, N. Pala, and A. Kadri, "Iot-based occupancy monitoring techniques for energy-efficient smart buildings," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2015 IEEE*, IEEE, 2015, pp. 58–63.

[38] K. Echigo, T. Petrovic, and H. Morikawa, "Detecting presence from a wifi router's electric power consumption by machine learning," *IEEE Access*, vol. 6, pp. 9679–9689, 2018.

[39] N. Li, G. Calis, and B. Becerik-Gerber, "Measuring and monitoring occupancy with an rfid based system for demand-driven hvac operations," *Automation in Construction*, vol. 24, pp. 89–99, 2012.

[40] N. Cao, J. Ting, S. Sen, and A. Raychowdhury, "Smart sensing for hvac control: Collaborative intelligence in optical and ir cameras," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 12, pp. 9785–9794, 2018.

[41] A. Coates, M. Hammoudeh, and K. G. Holmes, "Internet of things for buildings monitoring: Experiences and challenges," in *Proceedings of the International Conference on Future Networks and Distributed Systems*, ACM, 2017, p. 38.

[42] H. Zou, Y. Zhou, J. Yang, W. Gu, L. Xie, and C. Spanos, "Freedetector: Device-free occupancy detection with commodity wifi," in *2017 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops)*, Jun. 2017, pp. 1–5.

[43] R. Rana, B. Kusy, J. Wall, and W. Hu, "Novel activity classification and occupancy estimation methods for intelligent hvac (heating, ventilation and air conditioning) systems," *Energy*, vol. 93, pp. 245–255, 2015.

[44] W. Shen and G. Newsham, "Smart phone based occupancy detection in office buildings," pp. 632–636, 2016.

[45] J. Zou, Q. Zhao, W. Yang, and F. Wang, "Occupancy detection in the office by analyzing surveillance videos and its application to building energy conservation," *Energy and Buildings*, vol. 152, pp. 385–398, 2017.

[46] L. Zimmermann, R. Weigel, and G. Fischer, "Fusion of non-intrusive environmental sensors for occupancy detection in smart homes," *IEEE Internet of Things Journal*, p. 1, 2017.

[47] N. Nesa and I. Banerjee, "Iot-based sensor data fusion for occupancy sensing using dempster–shafer evidence theory for smart buildings," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1563–1570, 2017.

[48] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models," *Energy and Buildings*, vol. 112, pp. 28–39, 2016.

[49] A. Tyndall, R. Cardell-Oliver, and A. Keating, "Occupancy estimation using a low-pixel count thermal imager," *IEEE Sensors Journal*, vol. 16, no. 10, pp. 3784–3791, 2016.

[50] M. Abedi and F. Jazizadeh, "Deep-learning for occupancy detection using doppler radar and infrared thermal array sensors," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, IAARC Publications, vol. 36, 2019, pp. 1098–1105.

[51] Department of Business, Energy Industrial Strategy, *Building Energy Efficiency Survey, 2014 – 15: Overarching Report*, Nov. 2016.

[52] Local Authority Building Control. (2018). "What is the average house size in the uk?" [Online]. Available: `https://www.labc.co.uk/news/what-average-house-size-uk`.

[53] G. Liu, B. Liu, W. Wang, J. Zhang, R. A. Athalye, D. Moser, E. Crowe, N. Bengtson, M. Effinger, L. Webster, *et al.*, "Advanced energy retrofit

guide office buildings," Pacific Northwest National Lab.(PNNL), Richland, WA (United States), Tech. Rep., 2011.

[54] V. L. Erickson, M. Á. Carreira-Perpiñán, and A. E. Cerpa, "Occupancy modeling and prediction for building energy management," *ACM Transactions on Sensor Networks (TOSN)*, vol. 10, no. 3, p. 42, 2014.

[55] "Data protection act 2018, c. 12," *UK Legislation*,

[56] T. Fawcett and S. Hampton, "Why & how energy efficiency policy should address smes," *Energy Policy*, vol. 140, p. 111 337, 2020.

[57] A. Herren and J. Hadley, "Barriers to environmental sustainability facing small businesses in durham, nc," *Unpublished Master's Thesis*, 2010.

[58] E. Nilsson and J. Mähler, *Rental affordability solutions for startups and smes*, 2017.

[59] Health Safety Executive, Ed., *Workplace (Health, Safety and Welfare) Regulations 1992: Approved Code of Practice and guidance*. The Stationery Office, 2013, p. 23.

[60] . S. Hampton. (2019). "Total office cost survey 2019."

[61] R. Ande. (2020). "Survey about energy consumption within small and medium enterprises," [Online]. Available: `https://www.surveymonkey.co.uk/r/BHLPS77`.

[62] X. Zhang, M. Pipattanasomporn, T. Chen, and S. Rahman, "An iot-based thermal model learning framework for smart buildings," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 518–527, 2019.

[63] Z. Yang, A. Ghahramani, and B. Becerik-Gerber, "Building occupancy diversity and hvac (heating, ventilation, and air conditioning) system energy efficiency," *Energy*, vol. 109, pp. 641–649, 2016.

[64] A. K. Mikkilineni, J. Dong, T. Kuruganti, and D. Fugate, "A novel occupancy detection solution using low-power ir-fpa based wireless occupancy sensor," *Energy and Buildings*, vol. 192, pp. 63–74, 2019.

[65] S. Singh and B. Aksanli, "Non-intrusive presence detection and position tracking for multiple people using low-resolution thermal sensors," *Journal of Sensor and Actuator Networks*, vol. 8, no. 3, p. 40, 2019.

[66] R. Lawson, C. Alcock, J. Cooper, and L. Burgess, "Factors affecting adoption of electronic commerce technologies by smes: An australian study," *Journal of small business and enterprise development*, 2003.

[67] L. Marquez, J. McGregor, and M. Syme, "Barriers to the adoption of energy efficiency measures for existing commercial buildings," *CSIRO Energy Transformed Flagship*, 2012.

[68] Carbon Trust. (2020). "Energy efficiency starts here: Sme survey results," [Online]. Available: `https://www.carbontrust.com/news-and-events/news/energy-efficiency-starts-here-sme-survey-results`.

[69] R. A. R. Ghazilla, N. Sakundarini, S. H. Abdul-Rashid, N. S. Ayub, E. U. Olugu, and S. N. Musa, "Drivers and barriers analysis for green manufacturing practices in malaysian smes: A preliminary findings," *Procedia Cirp*, vol. 26, pp. 658–663, 2015.

[70] M. Jia, R. S. Srinivasan, and A. A. Raheem, "From occupancy to occupant behavior: An analytical survey of data acquisition technologies, modeling methodologies and simulation coupling mechanisms for building energy efficiency," *Renewable and Sustainable Energy Reviews*, vol. 68, pp. 525–540, 2017.

[71] M. P. T. Sulistyanto, K. B. Pranata, and Solikhan, "Preliminary study of utilizing internet of things for monitoring energy use in building to support energy audit process," in *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, 2017, pp. 1–7.

[72] J. Cosgrove, F. Doyle, F. Hardiman, and G. O'Farrell, "An approach to electricity monitoring and targeting (m&t) in irish precision engineering smes," in *International Conference on Sustainable Design and Manufacturing*, Springer, 2016, pp. 157–168.

[73] K. D. Dugay, A. O. Unira, and B. E. P. Visperas, "Energy monitoring and consumption reduction by automated branch circuit actuation for tertiary mainstreet establishments," in *2019 2nd World Symposium on Communication Engineering (WSCE)*, 2019, pp. 195–199.

[74] J. Heilala, K. Klobut, T. Salonen, P. Siltanen, R. Ruusu, L. Urosevic, P. Reimer, A. Armijo, M. Sorli, T. Fatur, Z. Gantar, and A. Jung, "Ambient intelligence based monitoring and energy efficiency optimisation system," in *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)*, May 2011, pp. 1–6.

[75] S. Johnson, P. Ropion, and B. Dezfouli, "Synergy: A smart and scalable energy measurement platform for electricity consumers," in *2019 IEEE Global Humanitarian Technology Conference (GHTC)*, 2019, pp. 1–7.

[76] Z. Yang, S. Zarabi, E. Fernandes, I. Rua, H. Debeda, A. Salehian, D. Nairn, and L. Wei, "Electricity monitoring system with interchangeable piezoelectric energy harvesters and dynamic power management circuitry," in *2018 IEEE 18th International Conference on Nanotechnology (IEEE-NANO)*, 2018, pp. 1–4.

[77] M. Luna, G. L. Tona, M. Carmela di Piazza, M. Pucci, A. Accetta, D. Taibi, C. Vetro, and R. L. Grassa, "A prototype of wireless sensor for data acquisition in energy management systems," in *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, 2018, pp. 1–6.

[78] E. Al-Hassan, H. Shareef, M. M. Islam, A. Wahyudie, and A. A. Abdrabou, "Improved smart power socket for monitoring and controlling electrical home appliances," *IEEE Access*, vol. 6, pp. 49 292–49 305, 2018.

[79] Midatlantic Controls. (2020). "How much does a building automation system cost," [Online]. Available: https://info.midatlanticcontrols.com/blog/how-much-does-a-building-automation-system-cost.

[80] Intel. (2016). "Cost savings roi smart building implementation," [Online]. Available: https://blogs.intel.com/iot/2016/06/20/costs-savings-roi-smart-building-implementation.

[81] Beringar. (2019). "Creating clever space." Accessed 2019-10-10, [Online]. Available: https://www.beringar.co.uk/product.

[82] Pressac. (2019). "Products." Accessed 2019-10-10, [Online]. Available: https://www.pressac.com/buy.

[83] Seeed Studio. (2019). "Order your smart citizen kit." Accessed 2019-10-10, [Online]. Available: https://smartcitizen.me.

[84] Premier Farnell Limited. (2020). "Product search: Motion sensor switches," [Online]. Available: https://cpc.farnell.com/c/electronic-electrical-components/sensors-transducers/sensors/sensor-switches/motion-sensor-switches.

[85] S. Dharur, C. Hota, and K. Swaminathan, "Energy efficient iot framework for smart buildings," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2017, pp. 793–800.

[86] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, "Wireless sensor networks: A survey on recent developments and potential synergies," *The Journal of supercomputing*, vol. 68, no. 1, pp. 1–48, 2014.

[87] V. Bhasin, S. Kumar, P. Saxena, and C. Katti, "Security architectures in wireless sensor network," *International Journal of Information Technology*, vol. 12, no. 1, pp. 261–272, 2020.

[88] V. Mihai, C. Dragana, G. Stamatescu, D. Popescu, and L. Ichim, "Wireless sensor network architecture based on fog computing," in *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, IEEE, 2018, pp. 743–747.

[89] K. Thangaramya, K. Kulothungan, R. Logambigai, M. Selvi, S. Ganapathy, and A. Kannan, "Energy aware cluster and neuro-fuzzy based routing algorithm for wireless sensor networks in iot," *Computer Networks*, vol. 151, pp. 211–223, 2019.

[90] G. Abdul-Salaam, A. H. Abdullah, M. H. Anisi, A. Gani, and A. Alelaiwi, "A comparative analysis of energy conservation approaches in hybrid wireless sensor networks data collection protocols," *Telecommunication Systems*, vol. 61, no. 1, pp. 159–179, 2016.

[91] K. Guleria and A. K. Verma, "Comprehensive review for energy efficient hierarchical routing protocols on wireless sensor networks," *Wireless Networks*, vol. 25, no. 3, pp. 1159–1183, 2019.

[92] R. E. Mohamed, A. I. Saleh, M. Abdelrazzak, and A. S. Samra, "Survey on wireless sensor network applications and energy efficient routing protocols," *Wireless Personal Communications*, vol. 101, no. 2, pp. 1019–1055, 2018.

[93] B. Bhushan and G. Sahoo, "A comprehensive survey of secure and energy efficient routing protocols and data collection approaches in wireless sensor networks," in *2017 International Conference on Signal Processing and Communication (ICSPC)*, IEEE, 2017, pp. 294–299.

[94] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, and H.-J. Kim, "An intelligent data gathering schema with data fusion supported for mobile sink in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 15, no. 3, 2019.

[95] X. Deng, Y. Jiang, L. T. Yang, M. Lin, L. Yi, and M. Wang, "Data fusion based coverage optimization in heterogeneous sensor networks: A survey," *Information Fusion*, vol. 52, pp. 90–105, 2019.

[96] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Data collection and wireless communication in internet of things (iot) using economic analysis and pricing models: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2546–2590, 2016.

[97] J. Plata-Chaves, A. Bertrand, M. Moonen, S. Theodoridis, and A. M. Zoubir, "Heterogeneous and multitask wireless sensor networks—algorithms, applications, and challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 450–465, 2017.

[98] S. Mini, S. K. Udgata, and S. L. Sabat, "M-connected coverage problem in wireless sensor networks," *ISRN Sensor Networks*, vol. 2012, 2012.

[99] T. M. Cheng and A. V. Savkin, "A distributed self-deployment algorithm for the coverage of mobile wireless sensor networks," *IEEE Communications Letters*, vol. 13, no. 11, pp. 877–879, Nov. 2009.

[100] M. Hammoudeh, F. Al-Fayez, H. Lloyd, R. Newman, B. Adebisi, A. Bounceur, and A. Abuarqoub, "A wireless sensor network border monitoring system: Deployment issues and routing protocols," *IEEE Sensors Journal*, vol. 17, no. 8, pp. 2572–2582, Apr. 2017.

[101] S. Zhou, M. .-. Wu, and W. Shu, "Finding optimal placements for mobile sensors: Wireless sensor network topology adjustment," in *Proceedings of the IEEE 6th Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication (IEEE Cat. No.04EX710)*, vol. 2, 2004, pp. 529–532.

[102] K. Chakrabarty, S. S. Iyengar, Hairong Qi, and Eungchun Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Transactions on Computers*, vol. 51, no. 12, pp. 1448–1453, Dec. 2002.

[103] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, vol. 3, Mar. 2003, pp. 1609–1614.

[104] V. Nazarzehi, A. V. Savkin, and A. Baranzadeh, "Distributed 3d dynamic search coverage for mobile wireless sensor networks," *IEEE Communications Letters*, vol. 19, no. 4, pp. 633–636, Apr. 2015.

[105] F. Al-Turjman, "Optimized hexagon-based deployment for large-scale ubiquitous sensor networks," *Journal of Network and Systems Management*, vol. 26, no. 2, pp. 255–283, 2018.

[106] D. M. West and J. R. Allen, "How artificial intelligence is transforming the world," *Report. April*, vol. 24, 2018.

[107] K. Rabah, "Convergence of ai, iot, big data and blockchain: A review," *The Lake Institute Journal*, vol. 1, no. 1, pp. 1–18, 2018.

[108] A. Kankanhalli, Y. Charalabidis, and S. Mellouli, "Iot and ai for smart government: A research agenda," *Government Information Quarterly*, vol. 36, no. 2, pp. 304–309, 2019.

[109] D. P. Kumar, T. Amgoth, and C. S. R. Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey," *Information Fusion*, vol. 49, pp. 1–25, 2019.

[110] S. B. Calo, M. Touna, D. C. Verma, and A. Cullen, "Edge computing architecture for applying ai to iot," in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 3012–3016.

[111] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[112] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.

[113] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural networks: Tricks of the trade*, Springer, 2012, pp. 437–478.

[114] D. Murray, L. Stankovic, and V. Stankovic, "Explainable nilm networks," in *5th International Workshop on Non Intrusive Load Monitoring*, 2020.

[115] A. Koutsoukas, K. J. Monaghan, X. Li, and J. Huan, "Deep-learning: Investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data," *Journal of cheminformatics*, vol. 9, no. 1, pp. 1–13, 2017.

[116] C. Imrie, S. Durucan, and A. Korre, "River flow prediction using artificial neural networks: Generalisation beyond the calibration range," *Journal of Hydrology*, vol. 233, no. 1, pp. 138–153, 2000, ISSN: 0022-1694. DOI: https://doi.org/10.1016/S0022-1694(00)00228-6.

[117] S. Mishra, S. Mishra, A. Kayal, and S. R. Chudi, "Simulation in wireless sensor networks," *International Journal of Electronics Communication and Computer Technology (IJECCT)*, vol. 2, no. 4, p. 176, 2012.

[118] D. Zorbas, P. Raveneau, and Y. Ghamri-Doudane, "Assessing the cost of deploying and maintaining indoor wireless sensor networks with rf-power harvesting properties," *Pervasive and Mobile Computing*, vol. 43, pp. 64–77, 2018.

[119] The Pi Hut. (2020). "Raspberry pi zero w." Accessed 2019-10-10, [Online]. Available: `https : / / thepihut . com / collections / raspberry - pi/products/raspberry-pi-zero-w`.

[120] A. Abuarqoub, M. Hammoudeh, F. Alfayez, and O. Aldabbas, "A survey on wireless sensor networks simulation tools and testbeds," *Sensors, transducers, signal condi-tioning and wireless sensors networks advances in sensors: reviews*, vol. 3, no. 14, pp. 283–302, 2016.

[121] H. Sundani, H. Li, V. Devabhaktuni, M. Alam, and P. Bhattacharya, "Wireless sensor network simulators a survey and comparisons," *International Journal of Computer Networks*, vol. 2, no. 5, pp. 249–265, 2011.

[122] UK Government. (2020). "New national restrictions from 5 november," [Online]. Available: `https://www.gov.uk/guidance/new-national-restrictions-from-5-november#going-to-work-and-being-covid-19-secu`.

[123] A. Sobeih, W.-P. Chen, J. C. Hou, L.-C. Kung, N. Li, H. Lim, H.-Y. Tyan, and H. Zhang, "J-sim: A simulation environment for wireless sensor networks," in *38th Annual Simulation Symposium*, IEEE, 2005, pp. 175–187.

[124] S. Sundresh, W. Kim, and G. Agha, "Sens: A sensor, environment and network simulator," May 2004, pp. 221–228.

[125] F. Österlind, "A sensor network simulator for the contiki os," *SICS Research Report*, 2006.

[126] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: Accurate and scalable simulation of entire tinyos applications," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 126–137.

[127] G. Simon, "Prowler: Probabilistic wireless network simulator," *Institute for Software Integrated Systems, Nashville*, 2003.

[128] C. McDonald. (2009). "Cnet network simulator," [Online]. Available: `https://www.csse.uwa.edu.au/cnet`.

[129] S. Park, A. Savvides, and M. B. Srivastava, "Sensorsim: A simulation framework for sensor networks," in *Proceedings of the 3rd ACM*

international workshop on Modeling, analysis and simulation of wireless and mobile systems, 2000, pp. 104–111.

[130] M. Arnesano, G. Revel, and F. Seri, "A tool for the optimal sensor placement to optimize temperature monitoring in large sports spaces," *Automation in Construction*, vol. 68, pp. 223–234, 2016.

[131] Z. Y. Wu, K. Zhou, H. W. Shenton, and M. J. Chajes, "Development of sensor placement optimization tool and application to large-span cable-stayed bridge," *Journal of Civil Structural Health Monitoring*, vol. 9, no. 1, pp. 77–90, 2019.

[132] Persistent Sentinel. (2020). "Spot – sensor placement optimization tool," [Online]. Available: `https://www.persistentsentinel.com/products/spot`.

[133] Y.-S. Ma, J. Song, J.-Y. Kwak, M. Yu, D.-K. Woo, and P. Mah, "Realssim: A simulator for indoor sensor network systems," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, 2010, pp. 369–370.

[134] T. W. Garaas, "Sensor placement tool for rapid development of video sensor layouts.," in *SpringSim (SimAUD)*, Citeseer, 2011, pp. 134–137.

[135] IPICA Software LLC. (2020). "Jvsg: Cctv design software," [Online]. Available: `https://www.jvsg.com`.

[136] CCTVCAD Software. (2020). "Cctvcad," [Online]. Available: `http://www.cctvcad.com/cctv_design_software.html`.

[137] D-Link. (2020). "D-link surveilance floor planner," [Online]. Available: `https://tools.dlink.com/intro/sfp`.

[138] S. Mini, S. K. Udgata, and S. L. Sabat, "Sensor deployment and scheduling for target coverage problem in wireless sensor networks," *IEEE Sensors Journal*, vol. 14, no. 3, pp. 636–644, Mar. 2014.

[139] A. N. Njoya, A. A. A. Ari, M. N. Awa, C. Titouna, N. Labraoui, J. Y. Effa, W. Abdou, and A. Gueroui, "Hybrid wireless sensors deployment scheme with connectivity and coverage maintaining in wireless sensor networks," *Wireless Personal Communications*, pp. 1–25,

[140] R. Williams, *The geometrical foundation of natural structure: A source book of design*. Dover New York, 1979.

[141] author unknown. (1905). "Circles within rectangles," [Online]. Available: `https://www.engineeringtoolbox.com/circles-within-rectangled_1905.html`.

[142] ——, (year unknown). "Maximum number of circle packing into a rectangle," [Online]. Available: https://math.stackexchange.com/questions/2548513/maximum-number-of-circle-packing-into-a-rectangle/3181467#3181467.

[143] You-Chiun Wang, Chun-Chi Hu, and Yu-Chee Tseng, "Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks," in *First International Conference on Wireless Internet (WICON'05)*, Jul. 2005, pp. 114–121.

[144] Rapid Electronics Online Limited. (2020). "Hc-sr501 pir motion detector data sheet," [Online]. Available: https://static.rapidonline.com/pdf/74-1108_v2.pdf.

[145] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[146] Melexis NV. (2019). "Download datasheet for mlx90640," [Online]. Available: https://www.melexis.com/en/documents/documentation/datasheets/datasheet-mlx90640.

[147] R. P. Foundation. (2020). "Hardware documentation: Camera module." Accessed 2019-10-10, [Online]. Available: https://www.raspberrypi.org/documentation/hardware/camera.

[148] CO2Meter Inc. (2020). "Se0118 / k30 datashet," [Online]. Available: http://co2meters.com/Documentation/Datasheets/DS_SE_0118_CM_0024_Revised9%5C%20(1).pdf.

[149] R. P. Foundation. (2020). "Product brief raspberry pi 3b+." Accessed 2019-10-10, [Online]. Available: https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf.

[150] C.-W. Tsai, C.-H. Hsia, S.-J. Yang, S.-J. Liu, and Z.-Y. Fang, "Optimizing hyperparameters of deep learning in predicting bus passengers based on simulated annealing," *Applied Soft Computing*, vol. 88, p. 106 068, 2020.

[151] Y. Yoo, "Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches," *Knowledge-Based Systems*, vol. 178, pp. 74–83, 2019.

[152] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, "Hyperparameter optimization for machine learning models based on bayesian optimization," *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, 2019.

[153] C. Halimu, A. Kasem, and S. H. S. Newaz, "Empirical comparison of area under roc curve (auc) and mathew correlation coefficient (mcc) for evaluating machine learning algorithms on imbalanced datasets for binary classification," in *Proceedings of the 3rd International Conference on Machine Learning and Soft Computing*, New York, NY, USA, 2019, pp. 1–6.

[154] CO2Meter Inc. (2019). "Co2 safety monitor installation tips," [Online]. Available: `https://www.co2meter.com/blogs/news/23850561-co2-safety-monitor-installation-tips`.

[155] E. W. Weisstein. (2020). "Tessellation," [Online]. Available: `https://mathworld.wolfram.com/Tessellation.html`.

[156] P. Steadman, "Why are most buildings rectangular?" *Architectural Research Quarterly*, vol. 10, no. 2, pp. 119–130, 2006.

[157] Z. Chen, K. Gryllias, and W. Li, "Intelligent fault diagnosis for rotary machinery using transferable convolutional neural network," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 339–349, 2019.

[158] R. K. Samala, H.-P. Chan, L. Hadjiiski, M. A. Helvie, C. D. Richter, and K. H. Cha, "Breast cancer diagnosis in digital breast tomosynthesis: Effects of training sample size on multi-stage transfer learning using deep neural nets," *IEEE transactions on medical imaging*, vol. 38, no. 3, pp. 686–696, 2018.

[159] J. Prusa, T. M. Khoshgoftaar, and N. Seliya, "The effect of dataset size on training tweet sentiment classifiers," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2015, pp. 96–102.

[160] (2020). "Openems," [Online]. Available: `https://openems.io`.

[161] (2020). "Eclipse public license version 2.0," [Online]. Available: `https://www.eclipse.org/legal/epl-2.0`.

[162] A. C. Menezes, A. Cripps, D. Bouchlaghem, and R. Buswell, "Predicted vs. actual energy performance of non-domestic buildings: Using post-occupancy evaluation data to reduce the performance gap," *Elsevier, Applied energy*, vol. volume 97, pp. 355–364, 2012.

[163] (2020). "Energyplus," [Online]. Available: `https://energyplus.net`.

[164] S. Labs. (2020). "Datasheet si1132." Accessed 2019-10-10, [Online]. Available: `https://www.silabs.com/documents/public/data-sheets/Si1132.pdf`.

[165] Mouser Electronics Limited. (2020). "Products," [Online]. Available: `https://www.mouser.co.uk`.

[166] The Pi Hut. (2020). "The pi hut," [Online]. Available: https://thepihut.com.

[167] CO2Meter Inc. (2020). "The co2 specialists," [Online]. Available: https://www.co2meter.com.

[168] Piqqit. (2020). "Piqqit store," [Online]. Available: https://piqqit.store.

[169] University of Illinois. (2020). "Application guide for ems - energy management system user guide." Accessed 2019-10-10, [Online]. Available: https://www.energyplus.net/sites/default/files/docs/site_v8.3.0/EMS_Application_Guide/EMS_Application_Guide/index.html.

[170] C. K. Metallidou, K. E. Psannis, and E. A. Egyptiadou, "Energy efficiency in smart buildings: Iot approaches," *IEEE Access*, vol. 8, pp. 63 679–63 699, 2020.

[171] L. Stankovic and V. Stankovic, "The risks and benefits of ai smart meters," 2020.