# EXTENDING WIRELESS SENSOR NETWORK LIFETIME USING BATTERY STATE OF CHARGE ESTIMATION AND MEDIA ACCESS CONTROL LAYER IMPROVEMENTS

**OMER ALI**

**UNIVERSITI SAINS MALAYSIA**

**2022**

## TABLE OF CONTENTS

**ACKNOWLEDGEMENT** ..... **ii**

**DEDICATION** ..... **iii**

**TABLE OF CONTENTS** ..... **iv**

**LIST OF TABLES** ..... **x**

**LIST OF FIGURES** ..... **xii**

**LIST OF EQUATIONS** ..... **xvii**

**LIST OF SYMBOLS** ..... **xix**

**LIST OF ABBREVIATIONS** ..... **xx**

**LIST OF APPENDICES** ..... **xxii**

**ABSTRAK** ..... **xxiii**

**ABSTRACT** ..... **xxvi**

**CHAPTER 1 INTRODUCTION** ..... **1**

1.1 Background ..... 1

1.2 Scope and Motivation ..... 5

1.3 Problem Statement ..... 6

1.3.1 Unreliable communication ..... 7

1.3.2 Stringent energy constraints ..... 7

1.3.3 Improving Fault Tolerance ..... 8

1.3.4 Design challenges of WSN investigation in this research ..... 8

1.3.1(a) Design consideration in PHY Layer ..... 8

1.3.1(b) Design consideration in Data link Layer ..... 11

1.3.1(c) Energy Estimation Challenges in WSNs ..... 13

1.4 Research Objectives ..... 15

1.5 Research Scope ..... 15

1.6 Research Contributions ..... 17

1.7 Thesis Outline ..... 20

**CHAPTER 2 LITERATURE REVIEW ..... 22**

2.1 Wireless Sensor Network (WSN) Architecture ..... 22

2.2 Design constraints effecting WSN lifetime ..... 25

2.2.1 Design Challenges at the Physical Layer ..... 26

2.2.2 Design challenges at Data-link Layer ..... 27

2.2.3 Design challenges at upper layers ..... 28

2.3 Cross-layer energy conservation techniques for WSN ..... 29

2.3.1 Transmission Power Control (TPC) ..... 29

2.3.2 Energy-efficient techniques for WSN ..... 36

2.3.3 Duty cycled schemes ..... 38

2.3.4 Data driven techniques ..... 52

2.4 Battery Types and the need for characterization ..... 53

2.4.1 Classification of Batteries ..... 54

2.4.2 Types of battery chemistries ..... 55

2.4.2(a) Lithium-Ion batteries ..... 56

2.4.2(b) Lithium-Polymer (LiPo) ..... 56

2.4.2(c) Nickel-metal Hydride (Ni-MH) ..... 57

2.4.2(d) Zinc-Manganese Dioxide (Alkaline) ..... 58

2.4.3 Battery characterization for small-load applications ..... 58

2.5 Battery performance and capacity estimation ..... 60

2.6 Energy estimation and WSN architecture design ..... 64

2.6.1 Initial battery voltage estimation errors ..... 65

2.6.2 Linear analytical battery modelling ..... 65

2.6.3 Effect of discharge rate, and self-discharge on battery capacity ..... 66

2.6.4 Inability to record the impact of environmental factors on batteries ..... 67

2.7 WSN enhancements at PHY and MAC layers ..... 67

2.8 Summary ..... 71

**CHAPTER 3 BATTERY CHARACTERIZATION FOR WSN ..... 73**

3.1 Characterization perspective and Safe Operating Area (SOA) ..... 73

3.2 Battery Characterization System Overview ..... 76

3.2.1 Device Discharge Profiles ..... 77

3.2.2 Standards and protocols for Low-power and Low-rate applications ..... 78

3.2.3 Environmental Chamber ..... 80

3.2.4 Battery discharge system ..... 81

3.2.5 Battery Gauging – Test Procedure, Cycling and SoC Measurement ..... 83

3.3 Experiment Analysis ..... 85

3.3.1 Battery Characterization procedure ..... 85

3.3.2 Pre-conditioning Test ..... 86

3.3.3 Relaxation Test ..... 87

3.3.4 Battery Capacity Test ..... 89

3.3.5 Open Circuit Voltage Versus State-of-Charge ..... 92

3.3.6 Battery Surface Temperatures under Load ..... 94

3.4 Summary ..... 95

**CHAPTER 4 STATE OF CHARGE (SOC) ESTIMATION ..... 97**

4.1 SoC Estimation using Regression based techniques ..... 97

4.1.1 Data Filtering and Normalization ..... 99

4.1.2 Data Partitioning ..... 101

4.1.3 Performance Metrics (PM) ..... 102

4.1.4 Linear Regression (LR) ..... 102

4.1.5 LR Model Results ..... 104

4.2 Machine Learning (ML) Models ..... 108

4.2.1 MATLAB FFNN Model ..... 110
4.2.2 ML Algorithm Results ..... 110
4.2.3 FFNN Model performance for Li-Ion battery datasets ..... 112
4.2.4 FFNN Model performance for Li-Po battery datasets ..... 119
4.2.5 FFNN Model performance for Ni-MH battery datasets ..... 122
4.2.6 Summary ..... 125
4.3 Support Vector Machines (SVM) ..... 126
4.3.1 SVM Model performance of Li-Ion battery dataset ..... 130
4.3.2 SVM Model performance of Li-Po battery dataset ..... 135
4.3.3 SVM Model performance of Ni-MH battery dataset ..... 139
4.3.4 Short Summary ..... 143
4.4 Gaussian Process Regression (GPR) ..... 145
4.4.1 GPR Results ..... 148
4.4.2 GPR Model optimization ..... 153
4.4.3 GPR SoC estimation for Li-Ion batteries ..... 155
4.4.4 GPR SoC estimation for Li-Po batteries ..... 158
4.4.5 GPR SoC estimation for Ni-MH batteries ..... 160
4.4.6 GPR Model Performance Summary ..... 162
4.5 Summary ..... 164
**CHAPTER 5 MODEL DEPLOYMENT AND ON-LINE SOC CALCULATIONS ..... 165**
5.1 The Contiki Operating System ..... 165
5.2 Lightweight SOC estimation models ..... 167
5.2.1 On-device SOC estimation ..... 168
5.2.2 Contiki OS energy calculation ..... 172
5.2.3 KiBAM energy calculations ..... 173
5.2.4 Temperature-dependent Kinetic Battery Voltage (T-KiBAM) Model ..... 175

5.2.5 Limitations of Analytical battery models ..... 177
5.2.6 GPR based energy calculation ..... 178
5.3 Simulation Environment for Node's energy estimation ..... 179
5.3.1 Node requirements ..... 180
5.3.2 Network Requirements ..... 182
5.3.2(a) Estimating effect of CCR on PDR values ..... 184
5.3.2(b) Energy estimation in Cooja Simulator ..... 188
5.4 Results – Energy Estimation ..... 190
5.4.1 Performance based on average power consumption ..... 191
5.4.2 Average energy consumption of Source Nodes ..... 191
5.4.3 Average energy consumption of Sink Node ..... 195
5.4.4 Performance based on Node Duty Cycle ..... 198
5.4.4(a) Effect on Duty Cycles for Source nodes ..... 199
5.4.4(b) Effect on Duty Cycles for Sink node ..... 202
5.4.5 Node Lifetime estimation comparison ..... 205
5.5 Summary ..... 209
**CHAPTER 6 ADAPTIVE CLEAR CHANNEL ASSESSMENT (A-CCA) . 210**
6.1 Energy detection and Clear Channel Assessment ..... 210
6.1.1 MAC protocols for ED and collision avoidance in IEEE 802.15.4 ..... 212
6.1.2 Investigating effect of CCA adjustment ..... 214
6.2 Adaptive Clear Channel Assessment – System Design ..... 218
6.2.1 Network Environment and Configurations ..... 222
6.2.1(a) Network requirement ..... 222
6.2.1(b) Contiki OS Implementation ..... 224
6.3 Results and analysis ..... 227
6.3.1 Average Duty Cycle ..... 227
6.3.2 Average Network Power Consumption ..... 229

6.3.3 Node lifetime estimation ..... 230
6.4 Real-time on-device Investigation..... 232
6.4.1 A-CCA and its impact on device lifetime ..... 234
6.4.2 Transmission range evaluation..... 236
6.4.3 The impact of A-CCA on packet reception and radio state ..... 239
6.5 Summary ..... 245
**CHAPTER 7 CONCLUSION AND FUTURE RECOMMENDATIONS.... 246**
7.1 Conclusion..... 246
7.2 Recommendations for Future Work..... 250
7.2.1 Estimating viability of advance battery chemistries ..... 250
7.2.2 Calculating the effect of sub-zero temperature on performance ..... 250
7.2.3 Modelling charging profiles for energy-harvesting applications ..... 251
7.2.4 Investigation of A-CCA's impact on Routing layer..... 251
**REFERENCES..... 252**
APPENDICES
LIST OF PUBLICATIONS

## LIST OF TABLES

**Page**

Table 2.1: Battery types, pros, and cons for consumer electronics.....55

Table 2.2: Recent literature on battery discharge characterization. .....59

Table 2.3: A review of SoC estimation techniques.....63

Table 3.1: Application specific cell characterization perspective.....75

Table 3.2: Battery specifications used for characterization in the study.....76

Table 3.3: CC1352R2 platform power consumption specifications. .....80

Table 3.4: Tested drive cycles and corresponding discharge profiles .....83

Table 3.5: Recorded voltage measurements at SOC (10%, 50%, and 90%) after relaxation period.....89

Table 4.1: LR model statistical output of various battery chemistries.....105

Table 4.2: Model performance analysis for various battery datasets.....111

Table 4.3: Kernel functions used to develop Gram Matrix for predictor variables .129

Table 4.4: Performance metrics of SVM Kernels for Li-Ion battery dataset.....131

Table 4.5: Performance metrics of SVM Kernels for Li-Po battery dataset.....136

Table 4.6: Performance metrics of SVM Kernels for Ni-MH battery dataset .....140

Table 4.7: Model performance at various operating conditions .....151

Table 5.1: Contiki OS salient architectural features .....166

Table 5.2: CC1352R2 platform-based energy profiles (Lithium-based batteries) for each model.....181

Table 5.3: Network configurations for energy estimation simulations using Cooja simulator.....184

Table 5.4: Network protocol configurations across multiple layers. .....189

Table 5.5: Average change in energy consumption for the schemes under investigation.....198

Table 5.6: Average change in duty cycle for the schemes under investigation .....205

Table 5.7: Simulated device lifetime estimation using TKiBAM and proposed scheme. .....206

Table 5.8: Experimental evaluation of proposed schemes for device lifetime estimation.....208

Table 6.1: ContikiMAC based CCA check time intervals .....225

Table 6.2: Idle-listening and false wakeup times of nodes running ContikiMAC ..226

Table 6.3: Average time consumption of a in various transceiver states .....228

Table 6.4:Average Radio Duty Cycle for Rx and Tx modes .....228

Table 6.5: Average network Power consumption .....229

Table 6.6: Node lifetime estimation (simulations) comparison of ContikiMAC and proposed scheme. .....231

Table 6.7:Node lifetime estimation of proposed A-CCA scheme. .....235

## LIST OF FIGURES

**Page**

Figure 1.1: Energy Conservation architecture for WSNs in literature. ..... 3

Figure 2.1: Communication in a typical Wireless Sensor Network (WSN) ..... 23

Figure 2.2: A typical architecture of modern WSN node. ..... 24

Figure 2.3: Taxonomy of energy-efficient techniques discussed in the literature. .... 36

Figure 2.4: An illustration of radio duty cycling mechanism ..... 37

Figure 2.5: Classification of MAC protocols for WSNs. ..... 39

Figure 3.1: A typical operation of a Li-Ion cell under its SOA region ..... 74

Figure 3.2: System design and experiment flow chart for battery characterization... 77

Figure 3.3: Battery pre-conditioning tests in a thermal chamber ..... 81

Figure 3.4: Battery discharge cycling and data logging schematic ..... 82

Figure 3.5: Five-step test sequence for effective battery characterization. ..... 86

Figure 3.6: Discharging Capacity of batteries during preconditioning tests. ..... 87

Figure 3.7: Discharge characteristics of the batteries under IEEE 802.15.4 discharge profiles. ..... 91

Figure 3.8: OCV measurements against battery SOC while discharging. ..... 93

Figure 3.9: Effect of load current on battery surface temperatures. Nickel Metal Hydride (top left), Lithium Polymer (top right), lithium-Ion (bottom left), and Alkaline (bottom right). ..... 95

Figure 4.1: Experimental Flow diagram for ML modelling techniques ..... 99

Figure 4.2: A comparison of actual versus predicted SOC values in LR model ..... 106

Figure 4.3: Residual histogram for LR model ..... 107

Figure 4.4: Residual histogram distribution for LR model ..... 108

Figure 4.5: FFNN architecture in MATLB for SOC estimation. ..... 110

Figure 4.6: Number of epochs and associated network accuracy for Li-Ion batteries. ..... 113

Figure 4.7:Error histogram of models prior, and post optimization ..... 115

Figure 4.8: The function fitting mapping for predictor variable (voltage) for FFNN – Li-Ion batteries. Default configuration (top), optimized network (bottom). ..... 116

Figure 4.9: RMSE improvements by observing iterative increase in neurons in the hidden layers ..... 118

Figure 4.10: Model predicted voltage values mapping for Li-Ion battery datasets . 119

Figure 4.11: Error histogram for optimized 2-layer model for Li-Po batteries ..... 120

Figure 4.12: The function fit plot for predictor variable (Voltage) for 2-layer FFNN – LiPo batteries. ..... 121

Figure 4.13: Model predicted voltage values mapping for Li-Po battery datasets .. 122

Figure 4.14: Error histogram for 2-layer FFNN model of Ni-MH batteries ..... 123

Figure 4.15: The function fit plot for predictor variable (Voltage) for 2-layer FFNN – Ni-MH batteries. ..... 124

Figure 4.16: Model predicted voltage mapping for Ni-MH battery datasets ..... 124

Figure 4.17: An illustration SVR hyperplane and error margin boundary. ..... 129

Figure 4.18: Predicted SOC values for Li-Ion batteries using three SVM Kernel models ..... 134

Figure 4.19: Maximum deviations resulting in residuals plotted at error indices for Li-Ion battery dataset ..... 134

Figure 4.20: Gaussian Kernel based SVM optimization on Li-Ion battery dataset . 135

Figure 4.21: Predicted SOC values for Li-Po batteries using three SVM Kernel models ..... 138

Figure 4.22: Maximum deviations resulting in residuals plotted at error indices for Li-Po battery dataset ..... 139

Figure 4.23: Polynomial Kernel based SVM optimization on Ni-MH battery dataset ..... 141

Figure 4.24: Predicted SOC values for Ni-MH batteries using three SVM Kernel models ..... 143

Figure 4.25: The residual plots for the batteries under various optimized SVM kernels. ..... 143

Figure 4.26: Gaussian Process Regression kernel parameters mapping ..... 145

Figure 4.27: Discharge characteristics of various batteries under drive cycles and temperatures. ..... 149

Figure 4.28: GPR model Hyper-Tuning for optimized Kernel Parameters estimations ..... 155

Figure 4.29: Hyper tuned GPR Kernel response for Li-Ion battery dataset..... 156

Figure 4.30: GPR model hyper-tuning kernel comparison for Li-Ion batteries..... 157

Figure 4.31: Residual plots comparing Kernel optimization for Li-Ion battery dataset ..... 158

Figure 4.32: Hyper tuned GPR Kernel response for Li-Po battery dataset..... 159

Figure 4.33: GPR model hyper-tuning kernel comparison for Li-Po batteries..... 160

Figure 4.34:Residual plots comparing Kernel optimization for Li-Po battery dataset..... 160

Figure 4.35: Hyper tuned GPR Kernel response for Ni-MH battery dataset ..... 161

Figure 4.36: GPR model hyper-tuning kernel comparison for Ni-MH batteries ..... 162

Figure 4.37: Residual plots comparing Kernel optimization for Ni-MH battery dataset..... 162

Figure 5.1: GPR Model generation flowchart for embedded platform. ..... 168

Figure 5.2: A flowchart illustrating the SOC estimation and comparison of various strategies..... 171

Figure 5.3: A two-charge-well model of the Kinetic Battery Model (KiBAM) ..... 174

Figure 5.4: Proposed GPR-based scheme for device lifetime calculation. ..... 178

Figure 5.5: Contiki OS Energest Module algorithm for energy estimation. ..... 182

Figure 5.6: Network topology for CC1352R nodes in Cooja Simulator. ..... 183

Figure 5.7: Average PDR values versus Packet rates. .....185
Figure 5.8: Network Performance evaluation: PDR rates with CCR patterns .....186
Figure 5.9: Average duty cycle of the nodes in RX and TX states. .....187
Figure 5.10: Average energy consumption for source nodes at IDLE state .....192
Figure 5.11: Average energy consumption for source nodes at 1 pkt/s .....193
Figure 5.12: Average energy consumption for source nodes at 16 pkt/s .....194
Figure 5.13: Average energy consumption for source nodes at 32 pkt/s .....194
Figure 5.14: Average energy consumption for sink node at IDLE state (0 pkt/s) ...196
Figure 5.15: Average energy consumption for sink node at 1 pkt/s .....197
Figure 5.16: Average energy consumption for sink node at 16 pkt/s .....198
Figure 5.17: Average energy consumption for sink node at 32 pkt/s .....198
Figure 5.18: Average Duty Cycle for source node at IDLE State (0 pkt/s) .....200
Figure 5.19: Average Duty Cycle for source node at 1 pkt/s.....201
Figure 5.20: Average Duty Cycle for source node at 16 pkt/s.....202
Figure 5.21: Average Duty Cycle for source node at 32 pkt/s.....202
Figure 5.22:Average Duty Cycle for sink node at IDLE state (0 pkt/s) .....203
Figure 5.23:Average Duty Cycle for sink node at 1 pkt/s .....204
Figure 5.24:Average Duty Cycle for sink node at 16 pkt/s .....204
Figure 5.25:Average Duty Cycle for sink node at 32 pkt/s .....204
Figure 5.26: Experimental evaluation of remaining device lifetime.....208
Figure 6.1: Impact on radio duty cycle with increased channel check rate. .....215
Figure 6.2: Impact on average PDR with increased network load.....216
Figure 6.3: A logical flow diagram of the proposed A-CCA algorithm .....221
Figure 6.4: Network topology comprising of sky nodes for RPL routing .....223
Figure 6.5: The floor plan for laboratory experiments used for node placement. ...233
Figure 6.6: CC1352R boards configuration for laboratory experiments .....234

Figure 6.7:Transmission link distance and measured packet reception rate (with default ContikiMAC values and with transmit power variations in A-CCA) .....238

Figure 6.8: Standard ContikiMAC Versus A-CCA: 1) Tx request preceded by PL delay and CCA strobes to show transmission intent. 2) Periodic CCA checks to detect any incoming packets. 3) Data completion acknowledgements, resulting in no further CCA strobes. .....240

Figure 6.9: Packet reception rate comparison under IEEE 802.11 based network interference. .....243

Figure 6.10: Radio on Time analysis under IEEE 802.11 interference. .....244

## LIST OF EQUATIONS

**Page**

Equation (1): Device frequency calculation via Gain function....84
Equation (2): LTC4150 device frequency calculation....85
Equation (3): Unit charge calculation per interrupt (INT)....85
Equation (4): Unit charge calculation in terms of Coulombs....85
Equation (5): OCV errors over time interval....88
Equation (6): Battery discharge time calculation....90
Equation (7): Battery nominal discharge capacity with Peukert Coefficient....90
Equation (8): Normalized scaling transforms....100
Equation (9): Power transformation function....101
Equation (10): Polynomial regression model....103
Equation (11): Model equation for SOC estimation using regression model....103
Equation (12): Support Vector Regression Model (SVR)....127
Equation (13): Hyperplane estimation for SVR....128
Equation (14): Error constraints threshold for SVR....128
Equation (15): Bayes' rule for posterior calculation....146
Equation (16): Posterior distribution function....146
Equation (17): Gaussian probability density function (PDF)....146
Equation (18): Gaussian distribution estimation....147
Equation (19): Multi-variate Gaussian distribution....147
Equation (20): M-dimensional Gaussian process estimation....147
Equation (21): Co-variance matrix calculation for Radial Bias Function (RBF)....148
Equation (22): Multi-variate Gaussian distribution with co-variance....148

Equation (23): Sample distribution function for co-variance matrix ..... 148
Equation (24): Sample vectors distribution of Gaussian space ..... 148
Equation (25): Joint Gaussian distribution function ..... 148
Equation (26): Multi-variate conditional distribution ..... 148
Equation (27): Marginal predicted variance function ..... 148
Equation (28): $K^{th}$ order polynomial equation ..... 153
Equation (29): Polynomial equation and coefficients ..... 153
Equation (30): Simplified polynomial equation of $n^{th}$ order coefficients ..... 153
Equation (31): Quadratic polynomial mapping function ..... 154
Equation (32): Polynomial regression coefficients ..... 154
Equation (33): Total power consumption of a node in various states ..... 173
Equation (34): Manwell differential equations for charge wells ..... 175
Equation (35): $n^{th}$ order discrete battery discharge equations ..... 175
Equation (36): Differential equation to estimate battery internal voltage ..... 176
Equation (37): Time consumption for sensor node in each state ..... 189
Equation (38): Energy consumption of sensor node in CPU state ..... 190
Equation (39): Energy consumption of sensor node in LPM state ..... 190
Equation (40): Energy consumption of sensor node in TX state ..... 190
Equation (41): Energy consumption of sensor node in RX state ..... 190
Equation (42): Energy consumption of a typical quadrature receiver ..... 211
Equation (43): Node activity in each state ..... 226
Equation (44): Total power consumption of a sensor node ..... 227

## LIST OF SYMBOLS

| | |
|---|---|
| $dB$ | Decibels |
| $kWH$ | Kilo-Watt per Hour |
| $mA$ | Milli-Amperes |
| $mAh$ | Milli-Amperes per Hour |
| $C$ | Charge rate or C-rate |
| $Wh$ | Watt-Hour |
| $^{\circ}C$ | Degree Celsius |
| $\mu A$ | Micro-Ampere |
| $\mu s$ | Micro-seconds |
| $Kbps$ | Kilobits per second |
| $f$ | Device frequency |
| $V_{\delta t}$ | Voltage over time intervals |
| $\delta_{OCV}$ | OCV errors over time |
| $t$ | Time in hours |
| $k$ | Peukert Exponent |
| $C_p$ | Nominal discharge capacity |
| $\mu$ | Mean values |
| $Z$ | Normalized feature function |
| $\sigma$ | Standard Deviation |
| $\psi$ | Power Transformation |
| $\lambda$ | Scaling Function |
| $N_h$ | Number of hidden layers |
| $Y_i$ | Support vector regressor |
| $\varepsilon$ | Support vector constraints |
| $P(X)$ | Gaussian distribution |
| $\vec{y}^*$ | Conditional distribution |
| $\Delta^2$ | Predicted variance |
| $P_{TOTAL}$ | Total node power consumption |
| $V_b$ | Battery voltage |
| $E$ | Energy of quadrature receiver |

## LIST OF ABBREVIATIONS

| | |
|---|---|
| ACK | Acknowledge |
| ADC | Analog-to-Digital Converter |
| AH | Ampere-Hour |
| AI | Artificial Intelligence |
| ANN | Artificial Neural Networks |
| AODTPC | Adaptive On-demand Transmission Power Control |
| AP | Access Point |
| ARM | Advanced RISC Machines |
| ARQ | Automatic Repeat Request |
| BAN | Body Area Network |
| BASN | Body Area Sensor Network |
| BER | Bit Error Rate |
| BMAC | Berkeley MAC |
| BMS | Battery Management System |
| CA | Collision Avoidance |
| CCA | Clear Channel Assessment |
| CCR | Channel Check Rate |
| CH | Cluster Head |
| COTS | Commercially Off-the-shelf |
| CPU | Central Processing Unit |
| CRC | Cyclic Redundancy Check |
| CS | Carrier Sense |
| CSMA | Carrier Sense Multiple Access |
| CTS | Clear to Send |
| DC | Direct Current |
| DNN | Deep Neural Networks |
| DOD | Depth of Discharge |
| DSSS | Direct Sequence Spread Spectrum |
| ECC | Error Correction Codes |
| ECM | Electrical Circuit Models |
| EIS | Electrochemical Impedance Spectroscopy |
| EKF | Extended Kalman Filters |
| ELM | Extreme Learning Machine |
| FEC | Forward Error Correction |
| FFNN | Feedforward Neural Networks |
| FSK | Frequency Shift Keying |
| GA | Genetic Algorithm |
| GP | Gaussian Process |
| GPR | Gaussian Process Regression |
| GPU | Graphical Processing Unit |
| ICMP | Internet Control Messaging Protocol |
| IEEE | Institute of Electrical and Electronic Engineering |
| IP | Internet Protocol |
| LLC | Logical Link Control |
| LPL | Low Power Listening |
| LQI | Link Quality Index |

| | |
|---|---|
| LSTM | Long Short-term Memory |
| MAC | Medium Access Control |
| MAE | Mean Absolute Error |
| MCU | Microcontroller Unit |
| MIPS | Millions Instructions Per Second |
| ML | Machine Learning |
| MLP | Multilayer Perceptron |
| MSE | Mean Squared Error |
| NN | Neural Network |
| OCV | Open Circuit Voltage |
| ODTPC | On-demand Transmission Power Control |
| OS | Operating System |
| OSI | Open Systems Interconnection |
| PCA | Principle Component Analysis |
| PDF | Probability Density Function |
| PDR | Packet Delivery Rate |
| PHY | Physical Layer |
| PL | Phase Lock |
| PRR | Packet Reception Rate |
| QPSK | Quadrature Phase shift Keying |
| RBF | Radial Bias Filter |
| RDC | Radio Duty Cycling |
| RF | Radio Frequency |
| RFID | Radio Frequency Identification |
| RMS | Root Mean Square |
| RMSE | Root Mean Squared Error |
| RNN | Recurrent Neural Networks |
| ROT | Radio ON Time |
| RPL | Routing Protocol for Low-power and Lossy Networks |
| RQ | Research Questions |
| RSSI | Received Signal Strength Indicator |
| SINR | Signal to Interference and Noise Ratio |
| SJF | Shortest Job First |
| SMO | Sliding Mode Observer |
| SNR | Signal to Noise Ratio |
| SOA | Safe Operating Area |
| SOC | State of Charge |
| SOH | State of Health |
| SSE | Sum of Squared Errors |
| SVM | Support Vector Machines |
| SVR | Support Vector Regression |
| TCP | Transmission Control Protocol |
| TDMA | Time Division Multiple Access |
| TP | Transmission Power |
| TPC | Transmission Power Control |
| UDP | User Datagram Protocol |
| UWSN | Underwater Sensor Networks |
| WBAN | Wireless Body Area Network |
| WPAN | Wireless Personal Area Network |
| WSN | Wireless Sensor Network |

## LIST OF APPENDICES


| | |
|---|---|
| Appendix A | Battery parameters and specifications for system design |
| Appendix B | Feedforward Neural Network architecture requirements |
| Appendix C | Performance evaluation metrics for ML models |
| Appendix D | Energy estimation model calculations |
| Appendix E | Effect of CCA adjustments on Radio Duty Cycle of nodes |
| Appendix F | Effect of CCA adjustments on node energy consumption |

# MEMPERLUASKAN JANGKA HAYAT RANGKAIAN SENSOR TANPA WAYAR MENGGUNAKAN ANGGARAN KEADAAN CAS BATERI DAN PENINGKATAN LAPISAN KAWALAN AKSES MEDIA

## ABSTRAK

Rangkaian Penderia Tanpa Wayar terdiri daripada banyak nod penderia berkuasa rendah dan terhad yang biasanya digunakan dalam jarak yang dekat untuk melaksanakan pelbagai tugas penderiaan dan perolehan data. Peranti-peranti ini dikuasakan oleh bateri dalam kebanyakan kes, dengan tenaga atas papan yang terhad (kapasiti). Oleh itu, jangka hayat nod penderia biasanya ditentukan oleh penghujung baterinya. Tambahan pula, keadaan operasi (seperti beban) dan keadaan persekitaran (seperti suhu dan kelembapan) boleh memberi kesan langsung ke atas kapasiti bateri yang tersedia. Pada tahun kebelakangan ini, banyak usaha telah dilakukan untuk membangunkan platform WSN berkuasa rendah, cekap tenaga, teknik-teknik pengekodan yang mudah untuk lapisan-lapisan PHY, protokol MAC sedar tenaga untuk pautan data dan protokol penghalaan penyesuaian cepat menumpu untuk lapisan rangkaian.

Walaupun teknik-teknik ini berfungsi dengan baik dengan sendirinya, menggabungkan teknik-teknik ini ke dalam reka bentuk tujuan umum rentas lapisan menambah kerumitan pelaksanaan dan mengehadkan penggunaannya kepada aplikasi tertentu. Sebagai permulaan, pembangunan WSN biasanya dilakukan dalam penyelaku-penyelaku yang hanya boleh menganggarkan sebilangan kecil parameter operasi, yang mengakibatkan anggaran dunia sebenar yang tidak tepat daripada permulaan lagi. Ketidakupayaan untuk memantau kesan luaran pada bateri nod WSN adalah jarang dilaporkan yang mengakibatkan ketidaktepatan anggaran tenaga utama, yang seterusnya mencerminkan operasi setiap lapisan. Penambahbaikan berasaskan

Penilaian Saluran Jelas (CCA) telah dicadangkan oleh protokol MAC sebagai cekap dari segi pengiraan, dan lebih mudah untuk dilaksanakan; namun, disebabkan jadual yang tetap, ia mengehadkan kebolehskalaan apabila rangkaian berkembang. Untuk mengelakkan kesesakkan dan perlanggaran paket, banyak teknik telah menilai pelarasan tahap kuasa penghantaran. Kelemahan yang jelas mempunyai kesan langsung pada hayat nod dan kebolehsuaian apabila saiz rangkaian dan topologi-topologi berubah.

Untuk menangani isu-isu ini, penyelidikan ini memfokuskan pada pendekatan baru yang menyesuaikan set baharu teknik rentas lapisan yang lebih mudah dilaksanakan, lebih cekap dari segi pengiraan, bebas platform, meningkatkan ketepatan dan boleh digunakan pada teknik sedia ada. Sebagai permulaan, pencirian bateri yang menyeluruh dijalankan untuk menilai pelbagai kimia bateri serta pelbagai keadaan operasi dan persekitaran biasa WSN. Dalam hal ini, metodologi lima langkah baru telah dicadangkan yang mencadangkan keadaan-keadaan operasi bateri yang optimum untuk aplikasi-aplikasi WSN.

Kedua, parameter-parameter bateri telah dipetakan untuk mencipta algoritma-algoritma pembelajaran mesin untuk menganggar Keadaan Caj (SoC) bateri. Penilaian komprehensif beberapa teknik dibentangkan dalam hal ini. Berikutan itu, teknik Penapis Pincang Berjejarian berasaskan Regresi Proses Gaussian (GPR-RBF) yang ringan telah dicadangkan yang boleh mencapai ketepatan anggaran SoC melebihi 98 peratus untuk semua jenis bateri dan keadaan-keadaan operasi. Tambahan pula, sepanjang pengetahuan kami, pelaksanaan GPR-RBF untuk anggaran SoC pada peranti masa nyata adalah yang pertama seumpamanya dalam literatur. Ketiga, ujian meluas dalam penyelaku OS Contiki telah dijalankan untuk menentukan ketepatan algoritma ringan yang dicadangkan. Akhir sekali, untuk menyiasat kesan CCA dan

pelarasan kuasa menghantar, cadangan skema rentas lapisan Penilaian Saluran Jelas Suai (A-CCA) telah disahkan untuk peningkatan prestasi. Dapat disimpulkan bahawa dengan A-CCA, julat penghantaran, kadar penghantaran paket (PDR) dan kadar penerimaan paket (PRR) telah dinaikkan tanpa menanggung kos kitaran tugas tambahan. Jangka hayat nod penderia dilaporkan telah meningkat sebanyak 12% disebabkan gabungan pengoptimuman-pengoptimuman rentas lapisan.

# EXTENDING WIRELESS SENSOR NETWORK LIFETIME USING BATTERY STATE OF CHARGE ESTIMATION AND MEDIA ACCESS CONTROL LAYER IMPROVEMENTS

## ABSTRACT

Wireless Sensor Networks (WSNs) are comprised of many low-power, constrained sensor nodes that are typically deployed in close proximity to perform various sensing and data acquisition tasks. These devices are powered by batteries in most cases, with limited on-board energy (capacity). Therefore, a sensor node's lifetime is typically defined by the end of its batteries. Furthermore, operational conditions (such as load) and environmental (such as temperature and humidity) conditions can have a direct impact on the available battery capacity. In recent years, a lot of work has gone into developing low-power, energy-efficient WSN platforms, simple encoding techniques for PHY layers, energy-aware MAC protocols for data links, and fast-converging adaptive routing protocols for network layers.

Although these techniques perform well on their own, combining them into a cross-layer general purpose design adds to the implementation complexity and limits their use to specific applications. To begin with, WSN development is usually done in simulators that can only estimate a small number of operational parameters, resulting in inaccurate real-world estimates right from the start. The inability to monitor the external effect on WSN node batteries is rarely reported resulting in major energy estimation inaccuracies, that further reflects on every layer’s operation. Clear Channel Assessment (CCA) based enhancements were suggested by MAC protocols as being computationally efficient, and simpler to implement; however, due to the fixed schedule, they limit scalability as the network grows. To avoid congestion and packet

collision, many techniques evaluated transmission power level adjustment. The obvious drawbacks had a direct impact on node lifetime and adaptability as network size and topologies changed.

To address these issues, this research focuses on a novel approach that adapts a new set of cross-layer techniques that are easier to implement, more computationally efficient, platform independent, increase accuracy, and can be applied to existing techniques. To begin with, a thorough battery characterization is carried out to assess various battery chemistries as well as a variety of typical WSN operational and environmental conditions. In this regard, a novel five-step methodology was proposed suggesting the optimal battery operating conditions for WSN applications.

Second, the battery parameters were mapped to create machine learning algorithms for estimating the battery's State of Charge (SoC). A comprehensive evaluation of several techniques is presented in this regard. Following that, a lightweight Gaussian Process Regression based Radial Bias Filter (GPR-RBF) technique was proposed that can achieve SoC estimation accuracy of over 98 percent for all battery types and operational conditions. Furthermore, to the best of our knowledge the implementation of GPR-RBF for real-time on-device SoC estimation is the first of its kind in the literature. Third, extensive tests in the Contiki OS simulator were conducted to determine the accuracy of the proposed lightweight algorithm. Finally, to investigate the effect of CCA and transmit power adjustments, the proposed Adaptive Clear Channel Assessment (A-CCA) cross-layer scheme was validated for performance improvements. It was concluded that with A-CCA, the transmission range, packet delivery rate (PDR), and packet reception rate (PRR) were increased without incurring additional duty-cycle cost. Sensor node lifetime was reported to have increased by 12% because of the combined cross-layer optimizations.

# CHAPTER 1

# INTRODUCTION

This chapter begins by introducing the architecture of Wireless Sensor Networks (WSNs) by providing background information on some of the most important layers, such as physical, datalink, and network. Following that, major research challenges and research objectives are defined that helped to outline this dissertation and list the major contributions of the proposed work in relation to energy conservation of WSNs.

## 1.1 Background

The majority of WSN deployments are in remote areas where these devices are left unattended for the duration of their lives. Because of this, the research focus on WSN has always been on energy conservation. Typically, energy conservation can be distinguished by examining the various roles of a sensor node with a specific energy consumption footprint. Sensing, data processing, and communication are all included in this broad category (Basumatary & Debbarma, 2021). The sensing operation typically necessitates the acquisition of specific data from an on-board sensor, which is then converted into a digital signal for processing via the on-board Analog-to-Digital converter (ADC). The operation of a WSN sensor consumes very little power, and because it is extremely rare for a sensor node to continuously monitor the environment, the duty-cycled sensing operation consumes very little power. As previously stated, the device consumes a negligible amount of energy while processing sensed data due to the availability of ultra-low power hardware for WSN applications. Furthermore, because the data is not continuous, the computation hardware spends very little time processing its request while keeping a large schedule in deep sleep. Communication, on the other hand, entails data aggregation, transmission, and reception from one node to another.

This necessitates the radio being in its operational state, which consumes the most power. It is estimated that the radio transceivers consume almost 80% of the overall node energy (Anastasi, Conti, Di Francesco, & Passarella, 2009). Therefore, radio energy conservation of a WSN at various architectural layers is the principle focus of the research. (Odeyinka et al., 2020) presented an energy conservation architecture, as illustrated in Figure 1.1, which is later discussed at every operational layer.

In this thesis, battery powered wireless nodes with inability to recharge are considered. In such stringent energy budget applications, the wireless nodes can only operate before the batteries reach its cut-off voltage, therefore, a cross-layer optimization is most effective for energy conservation. In this research the adapted energy conservation architecture mostly focuses on MAC layer optimizations, in addition to the PHY layer adjustments that directly control radio transceiver's transmit power. These techniques ensure a reduction in idle-listening, collision and re-transmission, and partial compensation for over-hearing problems, as highlighted in Figure 1.1.

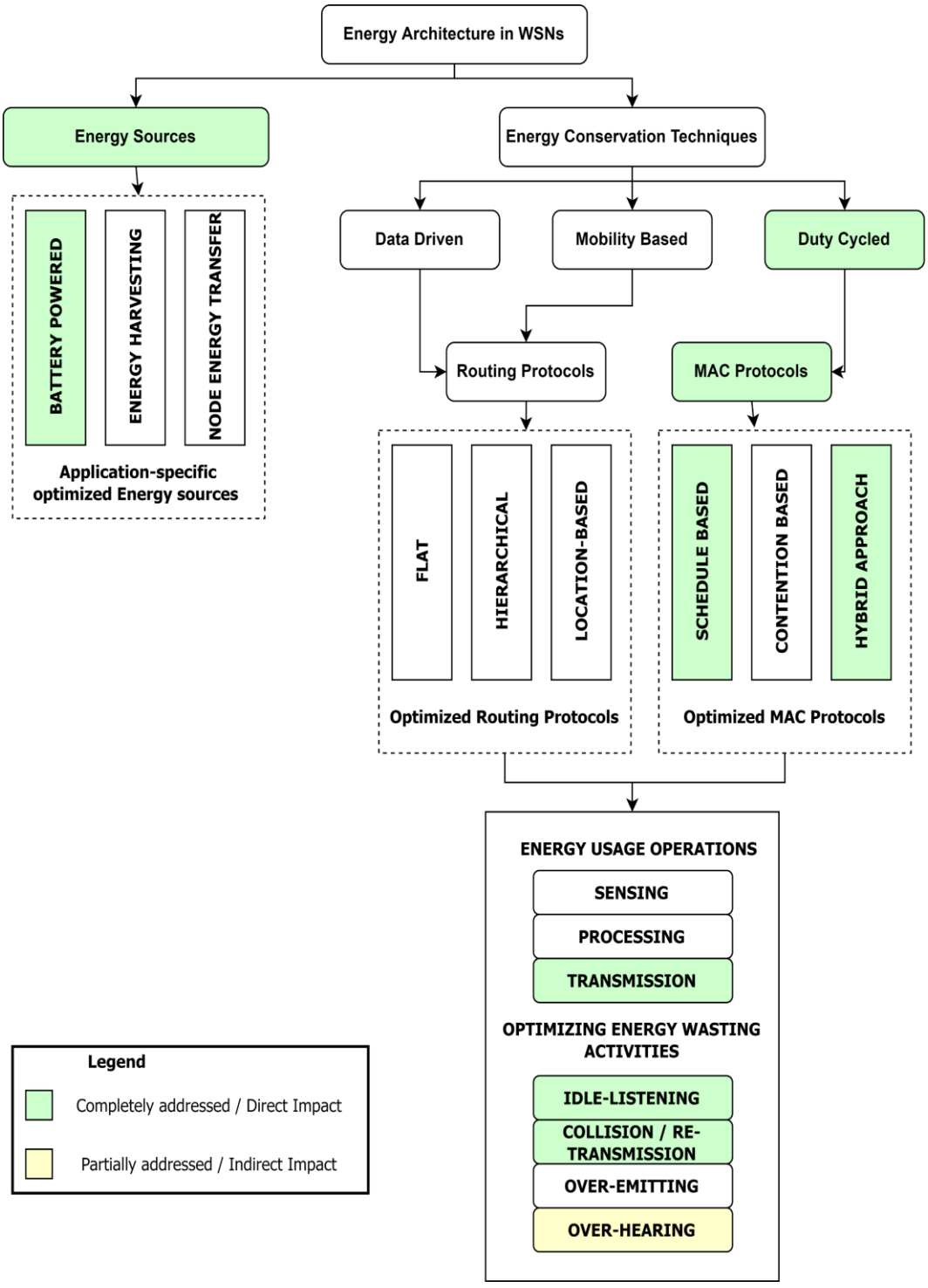



Figure 1.1: Energy Conservation architecture for WSNs in literature.

It is therefore necessary to present the functions of each layer for a WSN architecture, followed by some of the constraints and design challenges of the associated layers considered for this research. In this case, the functions that are supported by each layer are as follows:

- **Application Layer**: This is primarily an application-specific layer in which data is presented and integrated. Although WSNs have traditionally only transmitted data, current integration with IoT devices, gateways, and enterprise applications has expanded this layer in a variety of ways. However, the application layer has no direct impact on energy conservation in WSNs.
- **Transport Layer**: This layer oversees end-to-end data delivery, network integrity and reliability, and congestion control. Unlike traditional seven-layer models, WSNs mostly use IEEE 802.15.4 and a scaled-down IP stack with standard transport layer implementations. The transport layer may not have a direct impact on energy conservation in its standard WSN implementation.
- **Network Layer**: The network layer manages network topology, convergence, and routing, which controls packet and information distribution from source to sink nodes. The network layer has received a lot of attention, with energy-efficient routing protocols designed for faster network convergence, sparse network avoidance, and energy-efficient routing to increase network lifetime.
- **Data link Layer**: This layer's primary responsibility is to control access to the communication medium (the channel), neighbours, and associated link quality. The data link layer ensures channel availability, bandwidth management, and network scheduling, which aids in the provision of communication resources to all network devices. MAC protocols are typically designed to ensure high channel availability, reliable transmission, and low collision rates, allowing for significant energy

conservation in WSNs. The primary focus in this research is on optimising an established MAC protocol to address the challenges.

- **Physical Layer**: The physical layer is primarily in charge of the physical properties of communication technology, such as power levels, error correction techniques, modulation, and synchronization, to name a few. Because the physical layer parameters are directly related to the hardware, a great deal of energy conservation research has been conducted on this layer. In this study, in addition to making changes to the MAC layer, the PHY layer is used to make changes to radio power levels so that communication is more reliable and energy costs are less.

## 1.2 Scope and Motivation

WSNs have been deployed in remote areas to perform a variety of applications such as environmental monitoring, critical alert triggering, incident and event monitoring, and natural calamity and disaster detection, that can save lives. With its integration with the internet in recent years, the WSN has become the architectural foundation for Internet of Things (IoT) devices, which extend its functionality by providing two-way data and control information dissemination over the internet. Although there has been significant progress in both the hardware and software layers of the WSN, the fundamental purpose and operational principle remain the same. WSNs are typically deployed in clusters, with information broadcasted and relayed through the network's nodes. This allows for a variety of network topologies that change dynamically during operation. One of the primary causes of network topology change is the amount of available energy resources on the nodes, which can result in sparse networks if a node fails (I. F. Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002).

Because WSN nodes are primarily powered by battery sources, which regulate device lifetime based on available energy, significant research has been conducted to optimise battery usage in order to extend node and network lifetime (BenSaleh, Saida, Kacem, & Abid, 2020).

Another intriguing aspect is the cost and volume of nodes deployed in remote areas. WSNs are typically composed of hundreds, if not thousands, of nodes with low-cost hardware and limited battery capacities that are difficult or impossible to recharge (Kori, Angadi, Hiremath, & Iddalagi, 2009). This piqued the interest of scientists and researchers, who decided to focus on power-saving techniques from the physical to network layers to increase the lifetime of the WSN. Currently, cost-effective WSN platforms that consume very little power and allow for low-power and deep sleep states for nodes are widely available. Some chip manufacturers and system designers even focus on ultra-low power transceiver design for efficient power management, (Dutta, 2016; Molnar, Lu, Lanzisera, Cook, & Pister, 2004; Otis et al., 2018; Peiris et al., 2007). Furthermore, a lot of research has been conducted on MAC and network layers for energy conservation of WSN nodes, that is described below (Alfayez, Hammoudeh, & Abuarqoub, 2015; Kumar, Zhao, Wong, Guan, & Chong, 2018; Quintero, Estevez, Orchard, & Pérez, 2019; Samant, Kumar, & Swayamsiddha, 2020).

## 1.3 Problem Statement

Although WSNs are comprised of very simplified architectures to perform a variety of operations, the fundamental operations can be divided into sensing, processing, and communication, as described in previous sections. There is a slew of separate challenges for both its architecture and operation. However, when all of the aforementioned operations are investigated concurrently, a few constraints have a direct

impact on the design considerations (L. Wang, 2006), as discussed in the following sub-sections.

### 1.3.1 Unreliable communication

As WSN nodes are deployed mostly in harsh environments, environmental factors may have a great impact on the communication. Some of these challenges are purely natural, such as temperature, humidity, interference, non-line of sight, resulting in multi-path fading. Whereas the presence of noise, jamming signals, metals, and protocol-based interference can result in an elevated channel error ratio, resulting in unreliable communication and additional costs for re-transmission. This also results in network segmentation and early node failures. Although energy conservation is the prime design consideration, it is very common to observe a trade-off between reliable communication and an early end of life. After all, a WSN without reliable communication may not serve its intended purpose eventually.

### 1.3.2 Stringent energy constraints

WSN nodes are typically equipped with batteries with limited capacity, sometimes as low as 500 mAh. Furthermore, most Wireless Sensor Networks are set up outside, and without energy harvesting, it is impossible to recharge the power source. Sensor node lifetime is strongly dependent on the battery lifetime. A few dead nodes may result in network segmentation and holes, which further leads to unreliable communication and, in some cases, no communication at all. Because most WSNs are expected to last for years or decades, energy conservation and power management are becoming increasingly important in the design of Wireless Sensor Networks.

It is also crucial to remember that sensing and processing operations use a fraction of the energy that communication does. These operations are mostly performed at low frequencies, consuming less than 1 mW of power in some cases, on modern ultra-

low power platforms designed for WSN applications. In this regard, it has been demonstrated that using standard communication channels and operational parameters, a node transmitting 1 KB of data consumes almost the same energy as that of executing 3 million instructions by a processor capable of performing 100 MIPS (millions of instructions per second). This signified the fact that the node must spend most of its time in sleep or processing states and the communication must be regulated by duty-cycle to maintain the required quality and reception rate.

### 1.3.3 Improving Fault Tolerance

Energy constraints, size, cost, memory, bandwidth, and computational resources all constrain WSN designs. To avoid communication failure, a network should have redundancy, which is common in WSNs due to their use in hostile environments. The network must also be fault tolerant, with the ability to self-adapt and maintain reliable communications. However, resource optimization may have a direct impact on the node's energy consumption footprint and thus the network's lifetime.

### 1.3.4 Design challenges of WSN investigation in this research

For WSN, energy-efficient techniques are mostly designed for PHY to Network layers. In this regard, a great deal of research has been conducted on these layers independently, as well as some cross-layer optimization techniques have also been proposed.

### 1.3.1(a) Design consideration in PHY Layer

In the design of the physical layer for sensor networks, energy conservation is crucial, even though scattering, shadowing, reflection, multi-pathing, and fading effects are the major design considerations of PHY layer. If the number of sensors in a network is high enough, multi-hop communication can work well to avoid shadowing and path

loss effects. In the same way that propagation losses and channel capacity make data less reliable, but the same time it can be used to reuse spatial frequencies. Error correction codes, may make communication more reliable, but they add a lot of extra work that shortens the life of a node (I. F. a. V. Akyildiz, M.C., 2010). The goal of source coding is to reduce the amount of data that needs to be sent by taking advantage of the statistical features of the information. Use the statistical properties of the channel to make the information sent look more like it's not as important as it should be. Thus, even if some parts of the information are distorted, it can still be decoded successfully. It is also called error control coding (ECC) or forward error correction (FEC). There are a lot of clever channel codes that have been used in communication systems. To keep things from going wrong, these codes are the foundation for things like automatic repeat request (ARQ), forward error correction (FEC), and hybrid ARQ schemes.

Because of their simplicity and low computational overhead, block codes are preferred in WSN for error detection and correction. Error detection codes are typically simpler, but as error correction is implemented, more processing overhead is required. For error detection and correction, most current WSN implementations prefer block codes such as BCH, CRC, and RS codes. Although error correction increases the computational cost, successful recovery of a corrupted packet without the need for re-transmission saves a significant amount of communication overhead, extending the network lifetime. In theory, joint source–channel coding performs better than traditional separation techniques (source–channel separation). Existing communication platforms, still adhere to the source–channel separation principle because they are mostly used in the digital realm. Joint source–channel coding necessitates analogue communication techniques, that are typically nonlinear and more difficult to implement. Therefore, there have been very few practical implementations of joint source–channel coding

techniques. For instance, in delay sensitive applications, (Karlsson & Skoglund, 2011) proposed a nth dimensional joint-source coding technique that used node's instantaneous channel quality to report a 1dB gain.

Similarly, the PHY layer makes use of various modulation schemes for optimization that have a direct impact on energy consumption. The sensed data is processed and presented in digital form. However, the wireless nature of the WSN demands the antenna to propagate this signal in the analogue domain. This very fact requires modulation schemes to convert the digital data to analog. WSNs mostly use Frequency Shift Keying (FSK) and Quadrature Pulse Shift Keying (QPSK) based techniques at the PHY layer. FSK schemes are easier to implement with a very simple receiver side design. However, they're prone to interference and channel noise, resulting in performance degradation. QPSK schemes are mostly used in conjunction with Direct-Sequence Spread Spectrum (DSSS), as is the case with the IEEE 802.15.4 standard for low-power wireless networks. QPSK modulation provides better bandwidth utilisation and noise immunity than Together with DSSS, both time and frequency domains are used for modulation and transmission of bits, which enables us to reduce interference and fading effects. These techniques help to a great extent in indoor environments as well as in dense outdoor deployments with higher network traffic.

Standard implementations, such as IEEE 802.15.4, leave little room for PHY layer optimization. Even though researchers are focusing on PHY layer optimizations, most of the proposed schemes have yet to be implemented. In addition, industry standards and the availability of commercially off-the-shelf (COTS) platforms for rapid deployment strictly follow the IEEE 802.15.4 standard, leaving the above-mentioned schemes primarily for academic research. To improve network performance, the only

option left is to investigate transceiver programmable power levels that can help extend range, communicate over noisy channels, improve Signal-to-Interference-plus-Noise Ratio (SINR), and thus reduce Bit Error Rate (BER).

### 1.3.1(b) Design consideration in Data link Layer

The data link layer serves as a connection point between the network and the physical layer. Furthermore, it is divided into two components: Medium Access Control (MAC) and Logical Link Control (LLC). By efficiently allocating medium access to contending nodes, the MAC module contributes significantly to network life conservation. The LLC layer sits above the MAC layer and is responsible for performing cyclic redundancy checks (CRCs), sequencing data, and adding suitable source and destination details. Additionally, the data link layer is responsible for data stream multiplexing and data frame detection. A typical network design consideration first establishes a network infrastructure, which includes establishing communication links between potentially thousands of nodes and enabling the network to self-organize. Second, the data link layer is made up of different protocols that can share communication resources between all nodes fairly and efficiently.

Because sensor nodes must be active most of the time to perform functions such as sensing and processing tasks, the data link layer in WSNs must be well defined. The MAC algorithms at this layer always ensure adequate energy conservation strategies to extend sensor node lifetime. Furthermore, network throughput is a critical design consideration that governs the device schedule and affects its lifetime. Furthermore, the MAC protocols must be scalable and adaptive in order to account for the dynamic network topologies that occur as a result of nodes joining or dying, which is common in WSN deployments (Verma, Singh, Singh, & Kumar, 2015). One of the most frequently reported and earliest issues in wireless networks was the hidden node

problem, which occurs when two nodes on opposite sides of a central node can hear it but not from each other. This occurred because of the distance between them (range) or an obstruction (as typically is the case of WSN deployments). IEEE 802.11 addressed this issue at the MAC layer by introducing Request to Send (RTS) and Clear to Send (CTS) frames. When this feature is enabled, the sending station sends an RTS and then waits for the AP to respond with a CTS. Since all channels in the network can listen to the Access Point (AP), the CTS induces them to delay any planned transmissions, enabling the transmitting terminal to transmit and receive a packet affirmation without the possibility of colliding. Because RTS/CTS adds overhead to the system by momentarily retaining the medium, it is generally used only for the largest packets whose transmission would be prohibitively expensive in terms of bandwidth. This mechanism reduces the likelihood of a collision in the receiver area caused by a station hidden from the transmitter for the duration of the RTS transmission, since all stations hear the CTS and keep the medium busy until the transaction is completed. While it alleviates the problem of hidden nodes by avoiding collisions and adding wait time, it also keeps the medium busy, resulting in increased energy conservation owing to the transceiver's active states and processing overheads (Garg, 2007).

Since transmission is not continuous, most WSN deployments use duty-cycled MAC protocols. The nodes are only required to sense environmental parameters after a certain amount of time has passed or when a trigger occurs. Furthermore, most of this data is processed locally, aggregated, and then sent out after a pre-determined interval governed by network throughput and QoS requirements. Therefore, duty-cycled MAC protocols are an excellent choice for energy-saving strategies. To save energy, a duty-cycled algorithm will typically toggle the transceiver between transmission and sleep states. The MAC protocol, as previously stated, makes sure that there is an efficient

duty cycle interval to make sure that transmissions are reliable and that the intended throughput and QoS are kept, while also cutting down on the time that the transceiver is in use to save energy.

In this thesis most of the duty-cycled MAC protocols such as S-MAC, T-MAC, DS-MAC, WiseMAC, X-MAC and ContikiMAC are discussed in the literature. These protocols implement those fundamental requirements set by IEEE 802.11 protocol for medium access management, as well as IEEE 802.15.4 protocol for Zigbee/WSN for protocol as well as transceiver standards (Alfayez et al., 2015). These MAC protocols mostly propose collision avoidance, active-duty cycle, frame synchronization, periodic wakeup schedules, and preamble sampling between neighbouring nodes for energy conservation. However, the shortcomings were mostly observed in network throughput, latency, and adaptability with respect to traffic patterns. Furthermore, the carrier sense implementation of the IEEE 802.15.4 protocol was observed to witness several implementations with varying numbers of clear channel assessment (CCA) strategies for collision avoidance (Zhu, Jia, & Zhang, 2022). This piqued our interest to investigate effective CCA implementations that could reduce collisions, network access time, and false wakeups. In this regard, the research also investigated some of the recent studies on CCA-based strategies, including LW-CCA, PR-CCA, and P-DCCA. These proposed techniques involved CCA reduction, random CCA, and power differential based CCA to avoid collisions, false wakeups, and reduced re-transmissions, thus setting forth the motivation for the proposed novel A-CCA algorithm.

#### 1.3.1(c) Energy Estimation Challenges in WSNs

As already established, WSNs comprise hundreds, if not thousands, of nodes. This very fact regulates the design process, where system designers mostly utilise simulators for the design and evaluation of WSNs before deployment. Furthermore,

most of the real-world WSN applications are deployed and later optimised in future iterations. On the one hand, simulators and emulators enable us to design and observe the operation of hardware for intended applications. However, it also limits key operational analytics that are crucial for optimization, energy estimation, and thus designing energy-aware algorithms. For instance, most of the WSN simulators use linear energy estimation algorithms that calculate a linear battery energy drop based on the hardware state during each iteration. The aggregated sum of these energy calculations helps to estimate the energy consumed and thus the remaining available energy that may power that node. Although these linear energy estimation models are computationally efficient and simple to implement, estimation accuracy is one of the most significant issues. Because batteries are highly non-linear devices, a linear energy decay adds up to estimation inaccuracy over the course of a full discharge cycle, yielding highly inaccurate results (in terms of available energy, and thus the device lifetime). Furthermore, battery chemistry, operational states, charging patterns, rest period, and environmental conditions (such as temperature and humidity) all have an impact on battery capacity and device lifetime. Although there are a variety of empirical and filter-based techniques for estimating battery charge, they are mostly used in high-power applications like microgrids and electric vehicles (EVs), to name a few.

Because batteries change their discharge behaviour in response to the applied load, it's only natural to assume that empirical techniques designed for high-power applications may not adapt well to low-power applications like WSNs. Furthermore, there is a lot of computational and storage power in high-power battery-based systems. A typical WSN node is powered by a low-power microcontroller unit (MCU) that is incapable of handling such sophisticated algorithms. This piqued our interest in developing a simple algorithm that accurately predicts various battery chemistries while

considering the effects of operating and environmental factors. Furthermore, a lightweight algorithm was proposed that is both accurate and can be run from an MCU without incurring additional unforgiving computational costs that shorten the lifespan of the WSN.

## 1.4 Research Objectives

The main research objectives of this thesis include:

1) To design a novel battery characterization methodology for WSNs that accounts for various battery chemistries, device load profiles, and effect of environmental stimuli on batteries.
2) To develop lightweight and adaptive SOC estimation ML models for embedded platforms.
3) To validate the performance of battery SOC estimation algorithm with existing techniques in an experimental setup
4) To design and validate the performance of (A-CCA) MAC protocol and study its impact on WSN node lifetime.

## 1.5 Research Scope

This study focused on enhancing the lifetime of WSN by cross-layer design optimization techniques. WSN are mostly designed and tested in simulated environments that do not account for actual battery energy available, thus leading to estimation errors. This research focused on extensively characterizing various battery chemistries including alkaline, lithium-ion, lithium-polymer, and nickel-metal hydride. However, advanced battery chemistries were not used as the operational complexity and cost are not suitable for WSN applications. Furthermore, the operational temperature ranges (0˚C, 5˚C, 15˚C, 25˚C, 35˚C, and 45˚C) were used to observe the discharge

characteristics of these batteries. The battery discharge behaviors at sub-zero temperatures were avoided in this research. In addition, a sample resolution time of 15 minutes was used as standard for all experiments. As the MAC protocol used in this research is duty-cycled, the sample resolution was selected to account for most of the event occurring. In this regard, Coulomb counting method was adapted to record battery SOC, voltage, and remaining capacity during all experiments. Moreover, only the effect of surface temperature on battery due to attached load and ambient conditions were investigated.

The battery discharge datasets were then used to design scalable, lightweight, and fast converging ML models that are suitable for embedded platform deployments. Although advanced ML techniques including convolutional neural networks (CNN), deep neural networks (DNN), recurrent neural networks (RNN), and long short-term memory (LSTM) performed well on initial investigation, however, these techniques are not suitable for embedded resource constrained platforms, thus avoided. In this regard, extensive MATLAB simulations were carried out to evaluate the SOC estimation accuracy on various ML techniques (including linear regression, feedforward neural networks, support vector machines, and Gaussian process regression). The best performing models were further evaluated, optimized, and tested for embedded platform. In this regard, novel code libraries were created to deploy the Gaussian process regression-based model on ARM Cortex M4 embedded platform.

Finally, Contiki OS was used for this study due to its ability to support a wide range of IEEE 802.15.4 supported hardware devices, as well as the ability to emulate the actual firmware in its simulator. However, cross-comparison of our proposed work on other WSN simulators is proposed as the future extension to this work. The energy estimation calculations were compared with Contiki OS's default mechanism, empirical

based models, as well as the proposed scheme. Filter and Kernel based schemes that are typically preferred for battery management systems (BMS) were neglected due to their higher computational cost and inability to adapt to embedded platform.

## 1.6 Research Contributions

The challenges presented in the previous section enabled us to formulate the research design questions (RQ) that were later addressed and contributed to this research.

- **RQ1:** What is the most suitable battery type for the WSN application?
- **RQ2:** What are the optimum conditions for charging and discharging a specific battery type?
- **RQ3:** Is there a standard for pre-conditioning and operating batteries for use in WSN applications?
- **RQ4:** What effect do environmental (such as temperature) and operational (such as load current) factors have on battery performance?
- **RQ5:** What are the possibilities for designing a lightweight algorithm for estimating battery capacity (such as SoC)?
- **RQ6:** How do machine learning algorithms for WSN applications compare to empirical and linear models?
- **RQ7:** What cross-layer design changes, based on accurate battery SoC information can improve device lifetime?

The following is a synopsis of the contributions presented in this thesis based on the above-mentioned research questions.

1. The design of battery characterization experiments to find out how different battery chemistries react to different environmental and

operational conditions. In this regard, four commercially available battery chemistries, including alkaline, lithium-ion, lithium-ion polymer, and nickel metal hydride-based, were chosen for investigation. A five-step novel battery characterization technique was proposed to mitigate hysteresis and pre-conditioning losses. Additionally, the optimum rest period was investigated where the battery recovers to its maximum capacity. Next, a programmable battery discharge rig was designed that discharged these batteries under controlled environmental conditions following IEEE 802.15.4 protocol based current profiles. This novel characterization enabled us to observe comprehensive battery discharge patterns for WSN-specific applications that, to the best of our knowledge, were not previously available. The proposed battery characterization revealed the suitable battery types for WSN applications by reporting complete SoC profiles during the discharge cycle. Furthermore, these experiments also enabled us to estimate the degree of variance between promised battery capacity and actual nominal capacity, which leads to energy estimation inaccuracies. Finally, the recorded datasets were utilised to design machine learning (ML) algorithms for SoC estimation.

2. Extensive experiments were conducted to contrast various ML techniques for SoC estimation. In this regard, polynomial regression, feedforward neural networks (FFNN), support vector machines (SVM), and Gaussian Regression (GR) techniques were designed for accurate SoC estimation. A novel Gaussian Process Regression (GPR) based Radial Bias Filter optimised (GPR-RBF) approach was proposed that

achieved over 98% SoC estimation accuracy. Furthermore, unlike other ML models where several models are required, the proposed technique was trained and validated on all battery datasets and performed equally for all battery types.

3. Design of MATLAB and native C code libraries to deploy the proposed scheme on ARM Cortex-M based MCU platforms. To the best of our knowledge, no previous GR based code deployment for real-time on-device estimation has been reported. In addition to the code deployment on the actual hardware, the libraries were extended to emulate the hardware in the Contiki OS environment using its Cooja simulator.

4. The hardware with the proposed GPR-RBF SoC estimation schemes was thoroughly investigated and contrasted against linear energy estimation models and empirical models. It was observed that the proposed model accurately identified the remaining battery capacity without incurring unforgiving additional computational cost. Next, the effect of temperature was compared to empirical models on the actual device, and the proposed scheme was found to be both accurate and power efficient.

5. The accurate SoC estimation enabled the design of the proposed novel energy-aware MAC algorithm (A-CCA) that reduced collisions, re-transmissions, and false wakeups. The proposed technique enables the transmitter to adaptively scan the channel for link quality, and signal interference to adjust transmit power. The transmit power and the phase-locked (PL) duration are transmitted to neighbouring receiver nodes to adjust their wakeup times. The receiver, after acknowledging the PL

duration, reduces its CCA intervals, saving both the channel activity estimation period and the busy state, thus further reducing energy consumption. Finally, with the ability to detect the transmitted signals above the noise floor, fewer re-transmissions are needed, pushing the transceiver into a fast-sleep state, conserving energy.

## 1.7 Thesis Outline

In the next chapter (Chapter 2), a detailed analysis of the current literature is presented. As this research is focused on cross-layer optimizations on PHY and MAC layers, as well as energy estimation techniques, the literature review investigates the current state-of-the art in these categories.

Chapter 3 explains various types of battery chemistries and the effects of environmental and operational parameters on their performance. Later, a comprehensive five-step methodology is presented that is used to pre-condition the batteries and discharge them to observe their response to external stimuli. In this chapter, detailed information on experimental setup, battery discharge setup, parameter extraction, and analysis for battery capacity and SoC estimation are presented.

In Chapter 4, a comprehensive design and analysis methodology for SoC estimation is presented. First, parameter extraction, and optimum data processing techniques are discussed to extract the desired feature set. Next, various regression and ML-based networks were designed for SoC estimation. This chapter details the design, evaluation, and optimization of each network.

In Chapter 5, the steps to deploying the proposed GPR-RBF scheme for SoC estimation are presented. Through detailed simulations and on-device calculations, the proposed scheme is compared with various other schemes for SoC estimation accuracy

analysis. Furthermore, a thorough examination of radio states and their effects on duty cycle, estimation accuracy, and node lifetime is provided.

Chapter 6 presents the proposed lightweight adaptive A-CCA algorithm that reduces power consumption and enhances node lifetime. A comprehensive study is presented where the proposed scheme is compared with standard ContikiMAC implementations to evaluate device range, network throughput, energy consumption, and device lifetime.

Finally, Chapter 7 concludes the research by summarising the significant contributions found in each chapter by aligning them with research objectives. This chapter also explains the shortcomings and lays out clear steps for the future in this area.

# CHAPTER 2

# LITERATURE REVIEW

In this chapter, a comprehensive literature review on wireless sensor node architecture and major design challenges are discussed. Design considerations are essential for understanding the characteristics that enable the optimization strategies for application-specific deployments. Furthermore, a comprehensive literature review on power-efficient strategies in WSNs is presented to introduce the reader to the techniques used in the following chapters to design adaptive and energy-efficient cross-layer techniques.

## 2.1 Wireless Sensor Network (WSN) Architecture

WSNs are composed of many low-power computational devices, most of which are battery-powered and have on-board sensors, that are spread across a large geographical area. WSN nodes are typically designed and developed to sense and relay a variety of environmental phenomena such as temperature, vibrations, sound intensity, humidity, and, in some specialized cases, to monitor events such as tsunami and earthquake warnings. Although almost all sensor nodes have a similar architecture, the nodes differ based on the job roles assigned to them. A typical source sensor node, for example, will detect a target phenomenon and relay the information to its cluster head (CH), who is responsible for collecting, locally processing, aggregating, and relaying this information to the sink node. In modern WSN deployments, sink nodes are either connected directly to internet gateways or they pass the information on to other internet interfaces. This allows the sensed information to be used in other places. A typical WSN network communication scenario is shown in Figure 2.1, where the target data is

transferred via the best available routing path and then relayed to the end user via the internet.

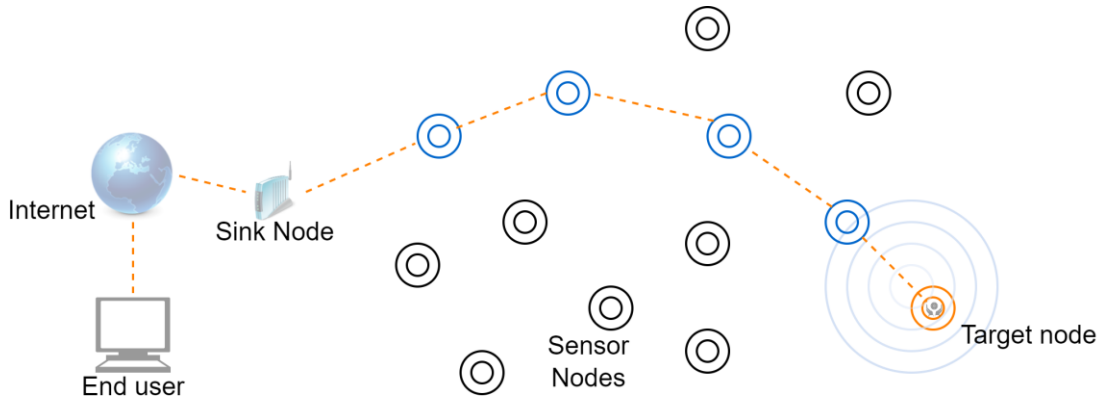



Figure 2.1: Communication in a typical Wireless Sensor Network (WSN)

WSN nodes have changed dramatically due to the availability of low power platforms with a variety of communication options. Furthermore, by connecting WSN to the internet, particularly to realise consumer Internet of Things (IoT) applications, the footprint and thus device heterogeneity have increased. Therefore, the following key components and sub-systems of a typical sensor node architecture are defined, as illustrated in Figure 2.2.

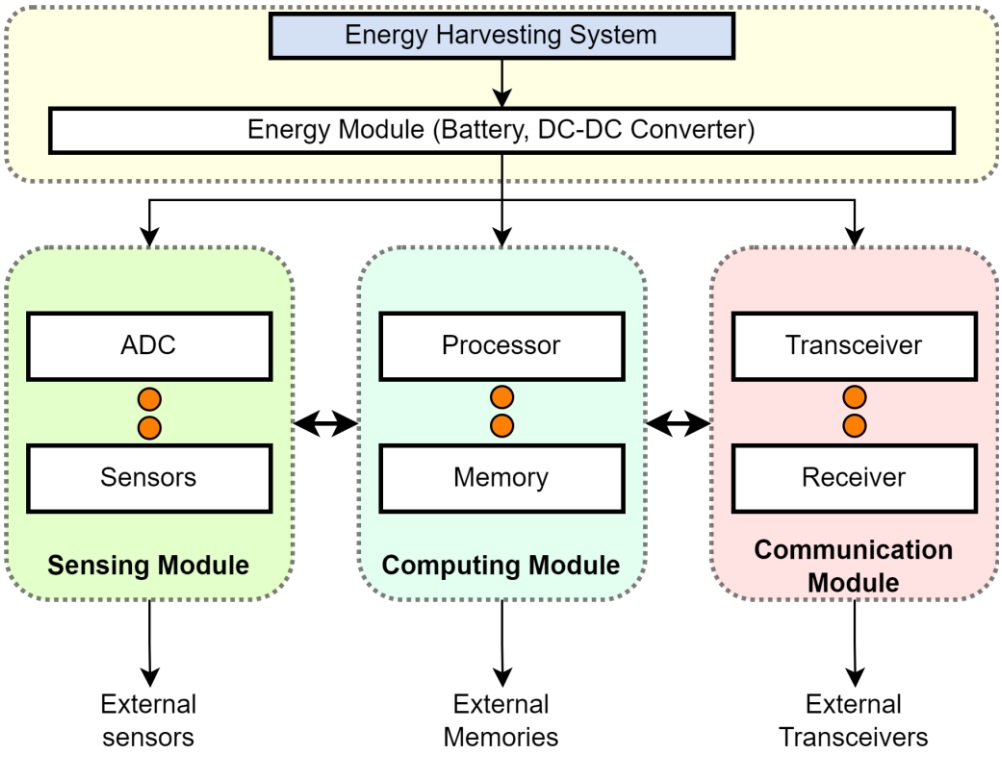


Figure 2.2: A typical architecture of modern WSN node.

1. The sensing module is normally designed as a modular and scalable subsystem with various onboard sensors to acquire environmental data. The sensor is typically programmed to collect data on a regular basis; however, in some cases, it is also triggered by external events. The sensor output is usually analog, which is later converted into digital signals using ADCs. The microcontrollers favour the digital data that originates from the ADCs, typically bound by the resolution of the device for application-specific requirements. The sensing module consumes the second highest amount of energy in comparison to the communication module. Although modern sensors are low-power, high accuracy, and low-latency devices, they're still prone to failure due to their deployment nature.
2. The processing modules usually contain the on-board microcontroller that is responsible for data acquisition, processing, and storage. In addition, this module also contains memory elements to store, aggregate, and retrieve sensor data before relaying the information to the next node. Modern micro-controller platforms designed for WSN applications are ultra-low power and can be programmed to keep the device in a deep sleep state. It is due to this ability that these modules consume a very small amount of energy.
3. The communication module that features an on-board radio transceiver is responsible for data transmission within the nodes. The radios are mostly programmable and are designed to adjust their power levels based on the range of communication, thus consuming a variable amount

of energy. The communication module consumes the most amount of energy and, on an estimate, may consume a thousand times more power as compared to the processing module. Therefore, most of the research in the WSN domain focuses on reducing the energy footprint of radio transceivers and keeping them in sleep states to extend their battery lifetime.

4. In most settings, WSN nodes are mostly battery powered with limited energy and, in most cases, are unable to recharge the batteries. Current WSN and small electronic devices are also designed to harvest energy to prolong their lifetimes. A power supply module normally consists of batteries, DC-DC converters, current and voltage limiters as per design specifications.

Additionally, these subsystems are also designed to extend their capabilities to external components to increase functionality, reliability, and integration. Next, some of the design challenges at PHY and the data-link layer are discussed. This is the main area of focus of this research.

## 2.2 Design constraints effecting WSN lifetime

The sensor nodes are mostly powered by small batteries with limited capacity to operate. This regulates the lifetime of the WSN. Although recent research highlights energy harvesting capabilities, system designers are always more inclined to make design choices based on a fixed and quantifiable supply of energy. The energy-constrained nature and the design challenges therefore require energy-efficient strategies that can conserve energy at various layers of the OSI reference architecture. In the following discussion, a brief account of these design challenges at various layers

is presented. This is further used to investigate the literature, and thus, to lay the foundation for the next chapters.

### 2.2.1 Design Challenges at the Physical Layer

The physical layer (PHY) is directly related to the device power levels, data stream conversion, and hardware-based encoding. Design optimization at the PHY layer using low-power components, reduced power levels (such as the transceiver's power) and reducing unnecessary components can significantly improve the power consumption of the sensor node. Some of the most common design factors that influence the power consumption at the PHY layer can be summarised as:

- WSNs are mostly periodic or event-driven, where the nodes gather data after a very long interval. Therefore, in WSN applications, higher order modulation schemes must be avoided to reduce bit encoding and transmission overheads. Although, modulation optimization is not the focus of this research, optimization layers at the PHY layer almost always focus on modulation schemes.
- The on-board sensors must directly relate to the location and application of the sensor node. It is very common practise to utilise a host of sensors during design or prototyping stages. However, by carefully designing the required sensors, a great deal of power can be saved. In addition, most of the current sensors are bundled into a single package that may not be directly used in a specific application. Therefore, the composite sensors must also be reduced. Finally, the choice of ADC resolution can be directly related to the accuracy of the sensor data. The design trade-off between accuracy and lifetime must always be maintained between applications.

- As such, WSN with dense deployments must be preferred as short-range communication requires less transceiver power, which could make a significant difference in device lifetime. However, with dense network deployment, multi-hop routing becomes another challenge that not only increases the routing overhead but may also add to the interference. Therefore, at the PHY layer, adaptive transceiver power adjustments must be made to actively adjust radio power levels based on network topology and conditions.

### 2.2.2 Design challenges at Data-link Layer

MAC protocols hold significance at this layer by managing the communication channels, channel sharing, and communication between the nodes. Therefore, MAC protocols must be designed with energy conservation in mind. Some of the major design issues in MAC layers can be summarised as:

- With MAC protocols that must be duty-cycled, and with adaptive switching of the transceiver, a great deal of energy can be conserved. MAC protocols must actively investigate the need for transmission, only when needed.
- Additionally, re-transmission and overhearing can waste a lot of network resources, as well as a node's energy. Packet collisions result in the re-transmission of the packets since the radio needs to be in an active state. Furthermore, due to network traffic, collisions, and proximity, the nodes can pick up packets that are not intended for them, thereby receiving, and dropping packets further consuming energy. Therefore, one of the most crucial design challenges of MAC protocols is collision and overhearing avoidance.

- MAC protocols must provide scalability and adaptability, due to dynamic deployment nature and fast-changing network topologies. A MAC protocol that cannot scale may result in segmented networks.
- Finally, MAC protocols must be adaptive in maintaining the desired system's QoS, throughput, while maintaining lower duty cycles for energy conservation.

### 2.2.3 Design challenges at upper layers

In addition to the challenges discussed above, high-level layers such as network and application are also affected by the design choices made at the lower layers. In this regard, the effect of power management on these layers to conserve energy was discussed. In a typical WSN deployment, the energy consumption of the node increases as the distance between them is increased. In addition, the nodes close to the sink or gateway nodes consume a lot more energy due to the higher level of packet transmission, thus depleting them earlier. Therefore, routing protocols must be able to address this problem. Furthermore, the application layer must carefully estimate the energy consumption of each node individually. Similarly, as there are heterogenous nodes in the network with varying energy footprints, routing protocols and application layer energy calculations must perform independent power computation calculations. Energy-aware algorithms can only be defined if the actual power consumption of the node is known. Therefore, the impact of the lower layers on the higher layers, together with the energy calculations on the application layer, may help to significantly increase the node lifetime. These challenges are now being investigated in the literature below.

## 2.3 Cross-layer energy conservation techniques for WSN

The major energy-saving techniques discussed in the literature are based on either (i) duty cycling, (ii) data-driven, or (iii) mobility-based approaches that primarily operate on the datalink and network layers (Anastasi et al., 2009; Idrees, Abou Jaoude, & Al-Qurabat, 2020; A. Rashid, Khan, Gul, Khan, & Khan, 2018). To reduce network traffic, retransmission, interference, and collision overheads, MAC and routing protocols are thoroughly investigated. These techniques can also be said to indirectly control the transceiver schedule and save a lot of energy by putting the WSN node in a sleep state most of the time. Furthermore, energy conservation techniques at the PHY layer play an important role in reducing the WSN's energy consumption footprint. However, data encoding and circuit design techniques that operate directly at PHY layers, as well as routing techniques, are not the primary focus of this research. Therefore, the proposed cross-layer design approach investigates (i) cross-layer transmission power control (TPC), (ii) energy-efficient techniques at the data-link layer, (iii) contention-based schemes at the data-link layer, and (iv) empirical modelling techniques.

### 2.3.1 Transmission Power Control (TPC)

Sensor nodes are typically used in dense environments with both external and internal environmental influences, resulting in interference and packet collisions. The network topology that forms dense and shallow networks is directly related to node power management. Because nodes in shallow networks are typically spaced apart, a higher transmit power is usually required to cover a greater range (Santi, 2005). Contention can have a significant impact on overall network performance, QoS, and energy consumption when devices transmit at their maximum transmission power. Concurrent nearby communications at significant transmission power causes packet

collisions; retransmissions and Clear Channel Assessment (CCA) struggle, with an increasing number of access attempts and backoff time (Ali, Ishak, & Bhatti, 2021; Chincoli, Syed, Exarchakos, & Liotta, 2016). Therefore, the transceivers are overused, resulting in additional battery exhaustion, transmission delays, and packet losses. In this regard, several techniques have been proposed to provide intermediate Transmission Power Control (TPC) to reduce the contention and network overhead as discussed above. Many researchers focused on TPC strategies in this regard in order to reduce their energy footprint while maintaining good overall link-quality (Chincoli & Liotta, 2018). A common consensus in the literature divides these strategies into homogenous and heterogenous transmission power configuration controls that are regulated by network topologies. Over the last decade, a great deal of research has been conducted on multi-hop communications with reduced transmission power levels. In these circumstances, the WSN nodes transmit at short, periodic low-power levels to neighbouring nodes, which then relay the information to the sink node. Although multi-hop communication can directly reduce the individual node's energy footprint, the routing overhead, packet relay, and inadvertently network traffic can further add to collisions and re-transmissions, thus reducing link quality and packet reception rates (PRR) (Fu, Sha, Hackmann, & Lu, 2012; Lin et al., 2016; Nikodem, Slabicki, Surmacz, & Wojciechowski, 2014).

Earlier investigation of TPC techniques suggested several performance improvements that could significantly reduce energy consumption. For starters, power control mechanisms improve a link's reliability. When the MAC protocol detects that link reliability is under a certain threshold, it increases transmission power, increasing the likelihood of effective data transmissions. Second, only nodes that truly need to occupy the same channel will request access to the medium, reducing the number of

network collisions. This increases network utilization, decreases latency times, and lowers the likelihood of hidden and exposed terminals. Finally, to further maximise energy savings, modulation and coding schemes with higher rates can be used at the PHY layers to adjust heavy workload (Correia, Macedo, dos Santos, Loureiro, & Nogueira, 2007; Rault, Bouabdallah, & Challal, 2014).

Son et al., (D. Son, Krishnamachari, & Heidemann, 2004) proposed one of the earliest experimental studies on TPC and its effect on PRR. The researchers discovered that network topologies in typical WSNs can change due to environmental changes or node-level incidents (such as node failure), resulting in dynamic links. Therefore, node transmission power levels can fluctuate over time, just like other dynamic variables. Furthermore, many factors, such as endpoint locations, natural landscape, and specific hardware differences, have been reported to have an impact on transmission power's impact on link quality (Zuniga et al., 2011). The proposed scheme necessitates an initial beaconing phase during which all nodes broadcast a series of messages at various transmission power levels while their neighbours record the PRR. The power levels for each link are determined by the required PRR. A packet-based signal strength control scheme is proposed to convert unreliable asymmetric and weak wireless links into credible wireless links with coherent link quality. To manage remaining unreliable links at adjusted transmission power levels, a blacklisting approach is combined, and connection and packet-based blacklisting approaches are introduced. Transmission power control with blacklisting could make new network topologies that have more reliable and consistent wireless links while saving energy and cutting down on channel interference.

Lin et al. (Lin et al., 2016) proposed an adaptive transmission power control (ATPC) technique for TPC that is lightweight. Based on the transmission power and

link quality, the nodes must create an independent model for the neighbouring node. Based on the Received Signal Strength Indicator (RSSI) parameter value, this adaptive technique employs a linear regression model to estimate the appropriate transmission power. A feedback-based controller maintains a minimum distinguishable transmission power for neighbouring nodes by correlating transmission power and link quality, which is later used to create a routing table for packet transfer. Nonetheless, in terms of implementation, the proposed technique is an efficient TPC module. However, based on variable network load and dynamic environmental conditions, actual performance metrics may result in segmented or overlapping routing entries, limiting adaptive TPC.

Along the same lines, Teng et al. (Teng et al., 2018), presented adaptive transmission power control-based reliable data forwarding technique (APRF) looks into the network topology and sink proximity to adjust transmission power levels. WSNs are known to have different energy consumption footprints in various segments of the network. For instance, the energy consumption of the near-sink segment is high as compared to that of the far-sink segment. The APRF technique utilises this information to adaptively adjust the transmission power levels of these areas. The proposed scheme aims to achieve a decrease of 40.37 percent in transmission delay, as well as an increase of 10.08% in the volume of data that is received at the sink node. However, due to complex implementation details and iterative transmission power level adjustments, the node lifetime was dropped by 13.77%, which is an alarming factor for energy-efficient techniques. Furthermore, the experimental investigation could not elaborate on the impact of TPC on re-transmissions, overhearing, and PRR.

Another interesting approach is to investigate on-demand TPC techniques that invoke TPC strategies only when needed the most. In other words, on-demand TPC may be designed to estimate the quality of links (for instance, good, intermediate, and bad)

based on the dynamically changing RSSI threshold values. In this regard, Kim et al. (J. Kim, Chang, & Kwon, 2008), presented on-demand TPC (ODTPC) scheme to address overhearing and energy conservation methods on the routing layer. In contrast to other schemes where link quality (RSSI thresholds) is measured constantly between neighbouring nodes, the proposed technique only initialises the measurement once ACK packets are exchanged between source and sink nodes. This ensures that ODTPC is only invoked when there is active communication between nodes, hence the desired time to adjust the transmission power (TP) level. In addition, no additional ACK packet is required for TP adjustment, resulting in no additional traffic overhead. This resulted in maintaining similar TP levels between communicating nodes, resulting in better overall link quality. Furthermore, due to reduced transmission overheads, the networks conserves energy, thus prolonging node lifetime. Along the same lines, Masood et al. (Masood, Ahmed, & Khan, 2012), extended the work presented in ODTPC to utilise a Kalman Filter (KF) based approach for adaptive ODTPC in WSNs. One of the main goals of this strategy is to increase ODTPC's performance when the channels are getting better or worse. Because when ODTPC reacts to changes in signal strength, AODTPC uses a Kalman filter to figure out what RSSI will be in the future based on what it has already measured. Because the Kalman filter doesn't need as much memory or computer power to process and store changes in the variable being studied, it is more computationally efficient than other prediction methods, such as Neural Network (NN) based adaptive methods.

TPC implementations are extensively studied for Wireless Body Area Networks (WBANs), a sub-set of WSNs where the onboard energy capacity is even further limited by the device form-factor and regulatory requirements. PHY layer improvements such as TPC are therefore extensively investigated in this domain. In this regard, Xiao et al.,

(Xiao, Dhamdhere, Sivaraman, & Burdett, 2009) presented a TPC strategy for body area sensor networks (BASN) for healthcare monitoring applications. The proposed multiplicative and additive increases and decreases the TPC algorithm in body area sensor networks that are complex and provide high reliability but consume more energy. In addition, the TP adjustment range is also limited in this algorithm, and energy consumption cannot be minimized. Similarly, Kim et al., (S. Kim & Eom, 2014) provided a foundation framework for adaptive TPC based on link-state-estimations of the network. In their proposed design, the researchers used RSSI threshold values to adjust short- and long-term transmission power levels to maintain reliable link quality. The experimental observations reported a 9.86% increase in node lifetime, however, no direct information about energy saving due to link-state-estimation was provided. Furthermore, important link quality parameters like packet loss, re-transmission, and so on were also not included in the report.

Moulton et al. (Moulton et al., 2010) devised adaptive feedback periodicity-based TPC for BAN, in which the time between feedback transmissions is adaptively varied to account for channel run-time variations. The proposed technique assisted in maintaining a constant transmission power level, resulting in energy savings. They did not, however, discuss energy conservation and PLR with TPC algorithms. Whereas, Vallejo et al. (Vallejo, Recas, & Ayala, 2015), presented a dynamic TPC scheme that takes active as well as reactive measures to control transmission power levels. The reactive algorithm has been found to be a good way to adapt the transmission to the changing needs of the channel and, at the same time, save energy. The reactive algorithm calculates the best transmission power level based on the packet error rate (PER) values that are recorded by a sensor that is worn on the body. This is done for every link, every posture, and every level of power in the radio. This characterization

phase must be completed by anyone who wants to use the system for the first time. For both the patient and the technical staff, this can be a very long and time-consuming process that takes a long time. The situation can get even more complicated if the patient has a long-term illness or isn't very mobile.

Together, both algorithms have PER values that are less than 6%. Even though the energy savings from the predictive strategy are a little lower than the energy savings from the reactive strategy, they still represent more than 22% of the energy savings. However, no one talks about how difficult it is to use the algorithms in the simulator and in real life. In addition, the dynamic changes in link quality caused by mobility could have a big impact on the proposed scheme.

Along the same lines, Sodhro et al.(Sodhro, Li, & Shah, 2016), investigated the transmission power management in WBAN applications, keeping packet loss ratio (PLR) in view. Their proposed technique contrasted with the previously mentioned schemes, where the central base station was polled periodically to provide transmission power feedback for neighbouring nodes. This periodic feedback was then used to design an adaptive power control (APC) algorithm for TPC. In contrast to previous schemes, dynamic TPC and RSSI values were used to effectively design the APC for better link quality estimations. The research illustrated an almost 41% increase in energy conservation compared to the previously mentioned schemes. However, one of the major drawbacks is that the APC algorithm was investigated only using simulated environments. Real-world applications are dynamically changing and cannot be estimated using simulated environments. Furthermore, the presence of other sources, especially IEEE 802.11 and IEEE 802.15.4-based sources that operate on similar bands, cannot be theoretically emulated. This results in purely limited theoretically modelling where simulated noise floor (based on RSSI values) was used to estimate transmission

power levels. Finally, the proposed technique suffered significant PLR as compared to other techniques, thus limiting its efficacy and operation ability in real-world WSN networks with dynamic environments.

### 2.3.2 Energy-efficient techniques for WSN

Several studies have been published in the literature with the goal of increasing the lifetime of WSN nodes while simultaneously reducing their power consumption. In general, such strategies can be divided into two broad categories: duty cycling techniques and data-driven techniques, respectively (Panwar & Misra, 2016). Figure 2.3 presents the taxonomy of energy-efficient techniques that are related to the proposed technique, and therefore, discussed below.

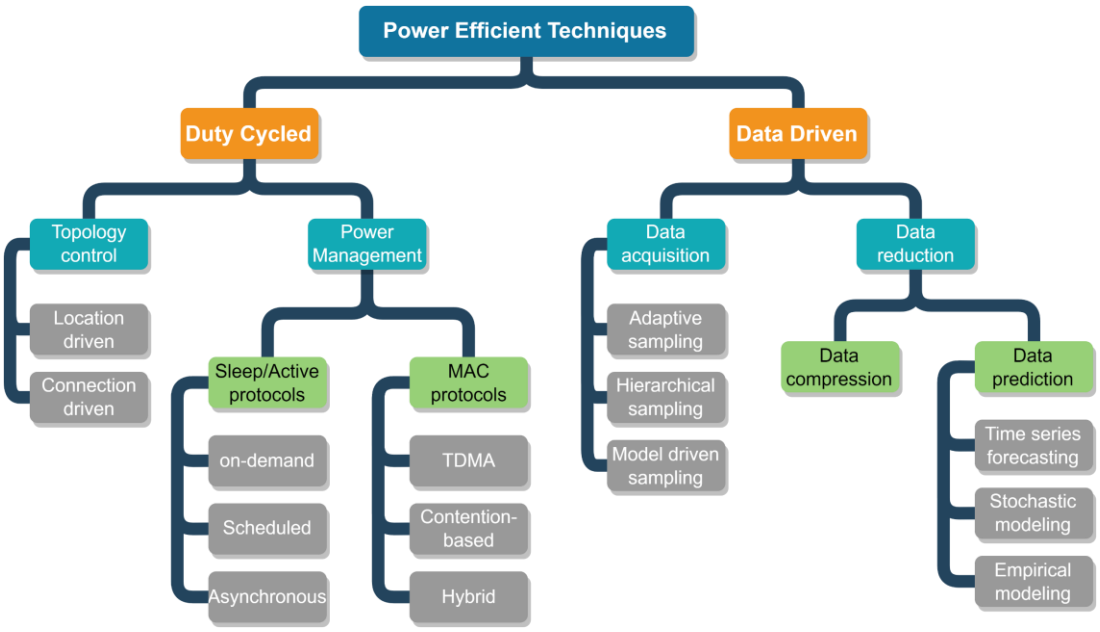


Figure 2.3: Taxonomy of energy-efficient techniques discussed in the literature.

As already mentioned in Section 2.3, the most common techniques to efficiently conserve energy in WSN is using duty cycling. It is known that radio transceivers consume the most power in a WSN node, so it is logical to only use them when needed. This emphasises the need to put the radio transceiver in a sleep state whenever communication is not needed, thus saving power. The radio transceivers are later

resumed to perform normal operation when there is a communication request (for instance, periodic event, data, or passive event). This behavior, where a radio transceiver is scheduled between its operating states, is termed "duty cycling." In other words, the duty cycle can also be defined as the time during which the radio transceiver (or the WSN node) is in its active state. Typically, all iterations of duty-cycling protocols provide a minimum of one sleep and one wakeup period to define their transitions between states, as illustrated by Figure 2.4.

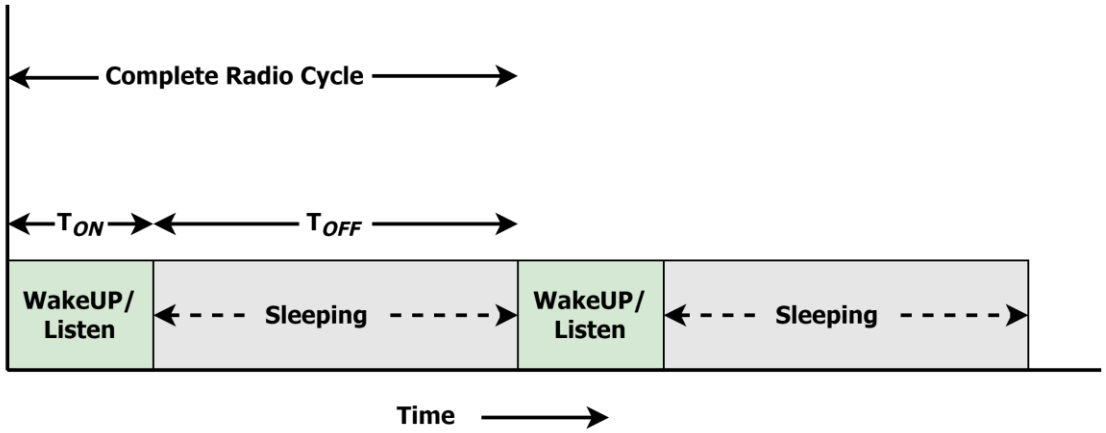



Figure 2.4: An illustration of radio duty cycling mechanism

Similarly, data-driven techniques are also designed to tap into data filtering, sampling ratios, local processing, aggregation, and in-network packet processing to further achieve efficient energy design applications for WSN. Although not directly related to the research objectives of this dissertation, data-driven techniques were still studied as it was imperative to study the role of data aggregation (or router/sink node) nodes. However, as this research focuses on fixed, low-data rate WSN applications, mobility-based schemes for energy conservation in WSN were not considered. In this regard, first the duty-cycled MAC protocols are studied, followed by the data-driven approaches (as illustrated and highlighted in Figure 2.3).

### 2.3.3 Duty cycled schemes

In general, power management schemes for MAC and routing protocols are divided into two categories: independent sleep/wakeup schemes and integrated sleep/wakeup schemes. Independent sleep and wake-up schemes are those that do not require any external power. Independent schemes, as the name implies, are on-demand strategies that are typically deployed as an abstraction layer on top of MAC protocols to provide additional functionality. The integrated schemes are tightly coupled with the MAC protocol suite of protocols. Therefore, dedicated memory access control (MAC) protocols have been developed to implement integrated sleep/wakeup duty cycling environments. These duty cycling-based schemes are being investigated in greater depth. To conserve energy, the on-demand protocols manage to keep the node in a sleep state until a packet from a neighbouring node is received by the node in question. In order to carry out this operation, the nodes must continuously monitor the channel for incoming transmissions while remaining in idle-listening modes. Immediately upon receipt of a packet, the node is triggered into its active state, which ensures successful communication.

Typically, on-demand schemes necessitate the use of two separate channels for the management of node activity and communication. One dedicated channel is reserved for communication between the nodes, whereas the other channel actively monitors the channel to ensure that the wakeup schedule is updated as frequently as possible. There are two major drawbacks to on-demand based schemes that should be mentioned. First and foremost, there is an increase in resource overhead to manage a dedicated channel. The nodes also consume a significant amount of power in order to actively monitor the channel for incoming packets, which is inefficient. Although the power consumption in idle listening is significantly lower than that in the radio

transceiver's transmit or receive states, it is still significant when compared to the power consumption in the sleep state. Following that, because all the nodes use a dedicated channel for wakeup activity, there is an increase in latency because of packet collisions. Figure 2.5 classifies some of the most studied MAC protocols in these categories.

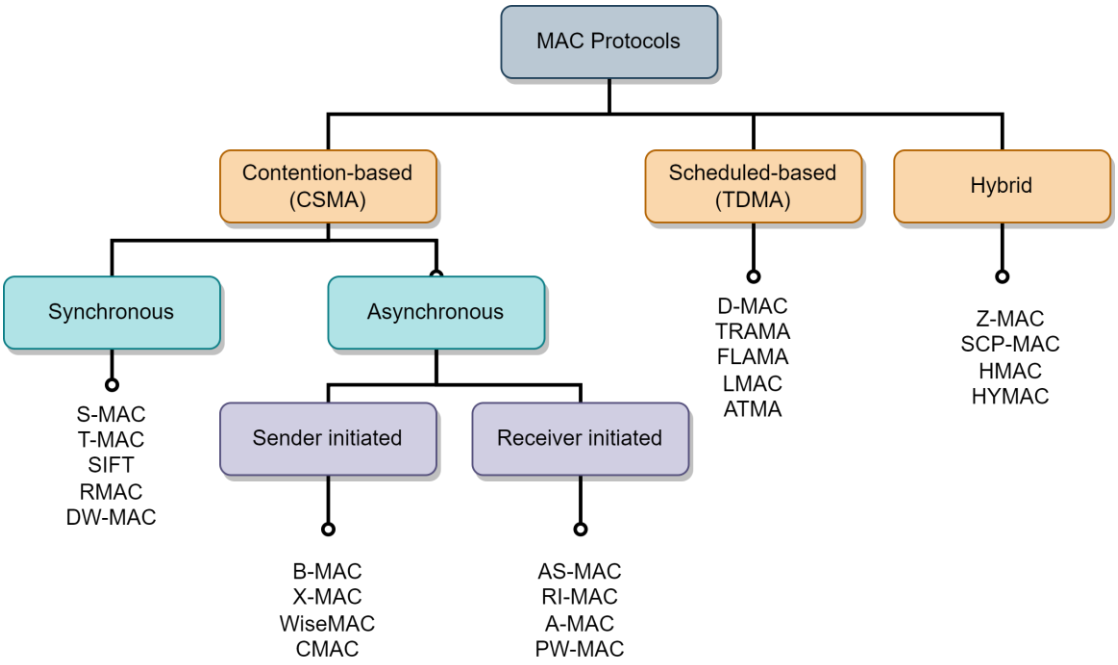



Figure 2.5: Classification of MAC protocols for WSNs.

In (Schurgers, Tsiatsis, & Srivastava, 2002) authors came up with the Sparse Topology and Energy Management (STEM) protocol, which uses two radio channels to send data packets and wake-up signals. There are periodic beacon messages sent through the wake-up channel by senders when they have a packet they want to send to their neighbor. To acknowledge the sender, it sends back a message and changes its state so that it can get the data packet through the communication channel. In the event of a wakeup channel collision, the corresponding node activates its data radio without sending an acknowledgement. Until and unless the initiator receives a wakeup acknowledgement from the target, the initiator repeats the beacon transmission in the wakeup. The STEM protocol was modified to achieve a completely different approach,

where instead of a single neighboring node being awakened all neighboring nodes are awakened (Schurgers, Tsiatsis, Ganeriwal, & Srivastava, 2002).

A Pipelined Tone Wakeup (PTW) protocol is also proposed in (X. Yang & Vaidya, 2004), which is predicated on transmitting a tone message from an initiator rather than a beacon message. PTW manages to strike an acceptable balance between power consumption and wakeup latency. Combining topology control protocols with on-demand protocols has been shown in the literature to reduce wakeup latency and thus power consumption by about 1%. Furthermore, reducing the wakeup latency can be accomplished by using a low-power radio for the wakeup channel (Srivastava, Singh, & Gupta, 2018; T. Zhang et al., 2020). While such techniques may reduce the wakeup latency, the wakeup radio's transmission range is reduced. Furthermore, Adam et al. (Adam et al., 2019), proposed an adaptive wakeup internal mechanism to mitigate the synchronization issues between radio wakeup intervals. The results of the simulation show that the proposed method can limit the false traffic convergence problem, which makes the receiver node wake up more often than the transmitter node. Furthermore, simulation results reveal that the improved technique may save energy, have less latency, improve throughput, and have a better packet delivery ratio than other similar works.

Similarly, in (Dinh, Kim, Gu, & Vasilakos, 2016) wake-up time self-learning MAC protocol was proposed that re-uses the receiver beacon messages to coordinate and synchronize the wakeup times of the nodes. Beacon messages sent by receiver-initiated MAC protocols can be reused to make it easier for nodes to wake up at the same time as their parent nodes without having to send extra messages. Based on the self-learning mechanism, L-MAC creates an on-demand staggered scheduler that lets any node send packets to the sink node at any time. This allows any node to send packets

to the sink node at any time. Experiments indicated that L-MAC is more energy efficient than current asynchronous MAC protocols and has the same latency as synchronous MAC protocols, which are more energy efficient than current asynchronous MAC protocols. Results show that the L-MAC protocol achieves better duty cycle as well as reduced latency.

In addition to the previously discussed independent sleep/active protocols, the scheduled rendezvous protocols have also been extensively studied for energy conservation in WSNs. Unlike the on-demand schemes where nodes continuously monitor the channel and wakeup for activity, the scheduled schemes tend to synchronize the nodes waking them up at a prescribed periodic timeslot. These techniques are specifically implemented as an abstraction layer on top of contention-based MAC protocols for better collision avoidance. As the nodes wake up periodically at the specified timeslot, this ensures that all neighboring nodes are always available in active states when needed, thus eliminating the need for continuous channel monitoring. Furthermore, broadcast wakeup message overhead is also reduced as the nodes are tightly synchronized and orchestrated. On the flip side, one of the most critical and challenging requirements for these techniques is the need of clock synchronization for simultaneous wakeup scheduling. A drift in clock would result in a complete out-of-sync network or segmented networks, that may increase collisions, retransmissions and in worst case scenario a time synchronization malfunction resulting in no communication at all.

In (Kaswan, Nitesh, & Jana, 2017), the authors proposed a k-means clustering based technique to identify the rendezvous points (RPs) to gather data. The data collection to RP, time sync information, and network parameters were used as a weight function to design a scheduling algorithm that covers the entire network, reducing

segmentation and hotspot issues. The authors (Nazir, Hasbullah, & Madani, 2011) presented a sleep/wake scheduling protocol for event-driven multi-hop wireless sensor network. One significant design consideration was to reduce the end-to-end transmission delay. In comparison to generic sleep/wake scheduling strategies, the proposed scheme implemented node scheduling based on traffic patterns. Nodes adjust their sleep/wake schedules in response to traffic loads in three critical ways: (a) the distance between the node and the sink node, (b) the importance of the node's location in terms of connectivity, and (c) if the node is in the vicinity of an event. The experimental results reported a significant decrease in average delay, PLR, and at the same time improved QoS. However, the results are obtained from simulations that may not completely emulate a real-world multi-hop wireless communication network. Furthermore, the distance calculations were fixed, and the target events in the node that are important for this algorithm's decision making were all fixed simulated values.

Along the same lines, Abtin et al. (Keshavarzian, Lee, & Venkatraman, 2006), presented several wakeup scheduling techniques for WSNs. The proposed techniques were computationally efficient as all nodes followed a similar active and sleep schedule, governed by ($T_{active}$) and ($T_{wakeup}$) timestamps, respectively. This enabled the entire network to synchronize the sleep/wakeup cycle with each other, however, still requiring a clock synchronization between the entire networks. These techniques were later implemented in several contention-based MAC protocols including S-MAC, and T-MAC to name a few. Although computationally efficient and easier to implement, these algorithms are prone to clock drift that may result in complete network segmentation and collision. Furthermore, the duty cycling approach incurs a trade-off between energy saving versus packet delivery delays. Dayong et al. (D. Ye & Zhang, 2017), presented a self-adaptive scheduling technique to address these problems. Unlike conventional

duty cycling techniques where sleep/wakeup periods are fixed, the proposed scheme utilized reinforcement learning strategy where every node adaptively decides its own working state. The simulation results highlight a very prominent limitation factor of the proposed technique, where dense networks and heavy traffic network's performance may degrade. As the learning process is based on estimation, it is always difficult to adjust the learning sample size and period which results in poor approximations for fast-changing networks. Similarly, another interesting approach is based on game theory where (Rose & Banu, 2019) presented an enhancement to the previously mentioned technique by reducing the idle-listening time of sender using reinforcement learning. As for effective communication both sender and receiver must be synchronized, the sender adjusts its time based on the receiver, thus only waking up when needed to transmit. This resulted in significant performance improvements and enhanced lifetime.

Asynchronous protocols allow neighboring nodes to have their own separate active intervals that can overlap for a certain number of cycles. While such protocols enable each node to communicate with its neighbors in a finite period of time, packet delay can be significant, particularly in multi-hop WSNs. Additionally, because it is not possible for all nearby nodes to be active at the same time (i.e., subset of them can be active concurrently), communication among nodes via broadcast messages is ineffective.

The authors (Paruchuri, Basavaraju, Durresi, Kannan, & Iyengar, 2004) presented a Random Asynchronous Wakeup (RAW) protocol for achieving low network latency by combining it with any geographical routing mechanism. The concept is to transmit packets to a neighboring active node (nodes alternate between sleep and active states at random) that the geographical routing protocol selects. Low network latency can be accomplished by routing a transmission to the node that is

closest to the target. Due to the random nature of the wake-up and sleep stages, there is no certainty that a node will discover surrounding active nodes. Therefore, this method is unsuitable for WSNs with a high density. Various techniques, such as (Schurgers, Tsiatsis, & Srivastava, 2002) and (Kansal, Somasundara, Jea, Srivastava, & Estrin, 2004), have been examined in the literature to ensure that when a node sends a packet to its neighbors, at least one node must be in an active state. The concepts are based on doing periodic listening to ascertain the existence of an asynchronous sender. While these systems are acceptable for high-density WSNs, they require a high level of complexity in their implementation and sensing procedures, which consume a significant amount of power.

Along the same lines, some researchers proposed asynchronous wakeup strategies for underwater wireless sensor networks (UWSNs) that helps to extend the device lifetime. In this regard, the authors (Su, Venkatesan, Li, & Computing, 2016) presented a combinatorial approach to reduce node duty cycle and prolong the lifetime. In this research the use of multiple sink nodes can effectively reduce the impact of high traffic workload. In addition, the authors (Su et al., 2021) extended their work by utilizing deep reinforcement learning and Long short-term memory (LSTM) networks to estimate idle-listening periods and network traffic (C. Li, Zhu, Xu, Gao, & Xu, 2018). This information then adaptively helps to adjust the wakeup schedules of the underwater sensor nodes.

Extending the literature to the next power management technique involving integrated sleep/active schemes, the MAC protocols that directly implement these algorithms were investigated (A. Bhatia & Hansdah, 2020). Typically, MAC protocols at data link-layer discuss channel access related issues. However, as the focus of this research is towards energy conservation, few MAC protocols that can be utilized in

energy constrained applications were discussed. In this regard, two of the most prominent sub-categories including Time Division Multiple Access (TDMA) based, and contention-based MAC protocols are discussed. Contention-based protocols, which are formally defined by the IEEE 802.11 standard, consume greater energy than TDMA protocols since energy is wasted in collisions and idle listening. Furthermore, they do not guarantee delivery times. Because network nodes can enter inactive states until their designated time slots, TDMA protocols consume less power. This also eliminate collisions and keep the delay to a minimum. TDMA algorithms can schedule one-hop or multi-hop. First, networks with one hop between nodes and base stations assign time slots in the reverse channel based on node allocation requests and deadlines (Kaur & Kumar, 2016). One node per slot can transmit because the base station receives all transmissions. The direct transmission of all sensor nodes to the base station is not always practical or energy efficient. This makes multi-hop TDMA scheduling more difficult than one-hop scheduling. If their receivers are not in conflict, many nodes can communicate at the same time (Ergen & Varaiya, 2010).

(Zibakalam, 2012) proposed a centralized TDMA scheduling algorithms for WSNs that accounts for node data congestion. Contrary to the conventional node-based scheduling for TDMA networks, the proposed scheme analyzes the network traffic pattern and prioritizes congestion areas. The congestion areas are then colour marked and various priority metrics then help to adjust the schedule of those areas to minimize congestion. Typically, TDMA techniques are designed to reduce clutter and network congestion by providing efficient flow of traffic from source to the sink, the proposed algorithm can be used an abstraction optimized strategy during congestion intervals. Similarly, Gilani et al. (Gilani, Sarrafi, & Abbaspour, 2013), proposed a hybrid protocol at for low-rate wireless personal area networks (LR-WPAN) to avoid congestion

problems. Typically, IEEE 802.15.4 standard regulates the use of slotted CSMA/CA based beacon-enabled schemes at data-link layer. One of the biggest downsides is a sudden energy increase and reduced throughput as the network traffic increases. The proposed scheme mitigates this by introducing TDMA based scheduling during the congestion intervals. Although this technique appears to be a smart approach for congestion avoidance, one great concern is the queue length that can exponentially grow as the network and/or the traffic increases. Furthermore, the analysis was only performed in simulated environments where channel characteristics and external parameters were kept constant, thus, real-world application and implications were not reported.

Along the same lines, in (Alvi et al., 2015) the authors proposed a TDMA based scheduling protocol for hierarchical WSNs where network traffic is prioritized based on shortest job first (SJF). The Bitmap-assisted shortest job first (BS-MAC) protocol provides greater number of short interval time slots as compared to the network nodes. This ensures that a time slot will be reserved for every member node within the network. Furthermore, the short interval time slot ensures a fair chance of channel access to neighboring nodes. Finally, with SJF algorithm, the proposed technique aims to clear the queue buffers faster, thus, suitable for low-to-medium sized network traffics. The simulation results illustrated the efficacy of the proposed scheme maintained higher throughout with significantly smaller packet delays. Ahmed et al.(A. Ahmad & Hanzálek, 2017), suggested a distributed scheduling technique to overcome the queue buffer issues within centralized scheduling techniques. Every node within the network proposed its own schedule for bi-directional traffic (sensed data, and actuation information), whereas the MAC protocol attempts to prioritize the shortest collision free path first to handle traffic congestion.

Similarly, Osamy et al.(Osamy, El-Sawy, & Khedr, 2020), proposed a genetic algorithm (GA) based strategy to minimize transmission delays within multi-hop WSNs. WSN has used GAs to solve a variety of problems. The most typical application of GAs in WSN is to achieve an energy-aware network under particular network conditions, as described in (T. Bhatia, Kansal, Goel, Verma, & Engineering, 2016; J.-L. Liu, Ravishankar, & Computing, 2011), improve localization accuracy, as explained in (Sharma, Kumar, & Engineering, 2018), and implement endpoints that can provide k-coverage and m-connectivity, as described in (Gupta, Kuila, Jana, & Engineering, 2016). In previous schemes (Khamayseh, Mardini, & Halima, 2018; T. Wang, Wu, & Mao, 2007) metaheuristics techniques were employed; however, the routing information was not considered during TDMA scheduling generation. Because the routing protocol does not choose the path, their schedules may be inadequate in terms of latency reduction (Louail & Felea, 2016). The proposed technique, EDTMA-GA assigns fitness functions to network traffic to reduce latency. The simulation results demonstrated performance gains in terms of reduced network latency, and duty cycle, that can conserve energy.

Furthermore, TDMA can also be implemented in routing protocols for energy-efficient implementations. In particular, clustering algorithms are widely used to logically segment a network where a cluster head (CH) is elected to locally process and disseminate the information between neighboring CH (S. K. Singh, Kumar, & Singh, 2017). CH also assign the timing slots to other nodes in its cluster, therefore acting as a master scheduler for its sub-cluster (Arisha, Youssef, & Younis, 2002; Effatparvar, Dehghan, & Rahmani, 2016; R. A. Rashid, Embong, Zaharim, & Fisal, 2009). Other techniques such FLAMA (Rajendran, Garcia-Luna-Aveces, & Obraczka, 2005)(Flow-aware Medium Access) and TRAMA (Traffic-aware Medium Access) utilize TDMA

slots to schedule and prioritize the network traffic flow to avoid congestion. In this regard, TRAMA (X. Zhang, Ansari, & Mähönen, 2009) divides the time slots in two phases. First, slot reservations are made by choosing random access period, followed by a scheduled period for slot assignment to the node. This enables every node to be aware of its time slot and respective radio transceiver switching schedule.

The contention-based MAC protocols use IEEE 802.11 standard implementation of CSMA/CA, with the idea of resolving contention: listening before transmitting. At its core, the most important job of a contention-based MAC protocol is to sense the channel before transmitting packets so that collisions, and thus re-transmission of packets can be avoided (Al-Obaidy, Momtahen, & Mohammadi, 2019; Chaari & Kamoun, 2010; Sonkar, Sangwan, & Tripathi, 2013). Over the past two decades various implementations and enhancements of contention-based protocols that were mostly based on protocols such as S-MAC, B-MAC, T-MAC, DMAC, TEEM, and receiver-initiated protocols RI-MAC were presented. Sensor MAC (S-MAC) is a low-power RTS/CTS protocol with four distinct radio transceiver states to avoid collisions and over-hearing problems. The nodes locally synchronize their active states, which is based on a fixed size time interval. The active fixed interval is typically regulated by the duty-cycle and the overall required network throughput. S-MAC are a favorable choice to avoid energy wastage in idle-listening states, as active cycles are synchronized (W. Ye, Heidemann, & Estrin, 2002). The nodes require RTS/CTS before communication commences and respective acknowledgements ensures reliable delivery. S-MAC protocol suffers two major problems. First, as the network size and traffic increases, the node with larger packets contains the channel for a longer period, thus the scheduling is unfair. Second, with the increase in network traffic, the overhead of maintaining node schedule rises exponentially, further limiting its performance. In

this regard, with S-MAC based implementations, energy-efficiency, packet latency, and throughput always suffer a trade-off. Some recent implementations of S-MAC protocol address these issues by using adaptive listening, and active periods depending on the network traffic (Jagriti & Lobiyal, 2018; K. Wang, Zhao, Shi, Xu, & Li, 2020; Zheng, 2019).

Timeout-MAC (T-MAC) protocol was proposed to solve the timing related issues of S-MAC protocol over increased network traffic (Van Dam & Langendoen, 2003). T-MAC introduced timeout values to toggle the state of a node from active to sleep. The node actively listens for communication and if it fails to hear any incoming transmission, the timeout value immediately switches it in sleep state. T-MAC performs significantly well in situations where network traffic continuously fluctuates. The analysis of T-MAC in fluctuating network traffic reveals a one-fifth power consumption compared to S-MAC (Al-Obaidy et al., 2019; Mihai, Alina, & Ion, 2010; Staub, Bernoulli, Anwander, Waelchli, & Braun, 2006). However, this technique also suffers scalability and complexity issues as the network grows. DMAC (Lu, Krishnamachari, & Raghavendra, 2004)protocol was designed to overcome the shortcomings of S-MAC and T-MAC protocols. The DMAC can be thought of as a better version of the Slotted Aloha algorithm, wherein slots are assigned to node sets based on a collecting data tree. All of a node's leaf node have transmit periods during its receive period, and they compete for the medium. When contrasted to other sleep/listen periods, it can achieve very low latency. Collision avoidance methods are not used in DMAC. Therefore, collisions will occur if multiple nodes with the same time frame consider sending to the same node. Traffic aware energy-efficient MAC (TEEM) (Suh & Ko, 2005) modifies the existing S-MAC protocol in two important ways: first, it turns off all nodes' radios much sooner when no data packet transfer is expected, and second, it eliminates the

RTS control packet although when data traffic is expected. TEEM combines Sync and RTS packets into one SyncRTS packet. Nodes send the SyncRTS packet in their Syncdata part to communicate with other nodes. The destination node receives the packet and begins communication, while the other nodes synchronize and sleep. TEEM MAC is suitable for low-traffic networks and small networks with low retransmission rates.

Another interesting type of contention-based MAC protocols are receiver-initiated, where unlike the sender, the sink node advertises its availability and initiates the transmission schedule. In this regard, one of the most studied protocols is receiver-initiated MAC (RI-MAC). The proposed technique is specifically designed to overcome the idle-listening problem at the receiver side that consumes a significant amount of power in WSNs (Y. Sun, Gurewitz, & Johnson, 2008). In terms of how the sender and receiver arrive at a rendezvous time, RI-MAC differs from previous research in asynchronous duty cycle MAC protocols. In RI-MAC, the sender stays active and listens quietly until the receiver sends a short beacon frame indicating that data transmission is ready to begin. Because only beacon and network information dominate the medium in RI-MAC, as opposed to preamble transmissions in LPL-based protocols, the medium's occupancy is reduced significantly, allowing other nodes to exchange data. The performance analysis of RI-MAC reveals that this protocol is most suited to low-to-medium traffic environments with minimum fluctuations (Duan, Zhao, Zhang, Zhang, & Li, 2017; Jain, 2020).

Berkeley-MAC (B-MAC) is a sender initiated technique that utilized channel assessment (CA) parameters including clear channel assessment (CCA) to achieve data transmission efficiency (Polastre, Hill, & Culler, 2004). For channel arbitration, B-MAC employs clear channel assessment (CCA) and packet backoffs, as well as link

layer acknowledgments for reliability and low-power listening (LPL). To optimize energy usage, latency, throughput, fairness, or reliability, B-MAC makes continuous policy decisions. BMAC uses an adaptive preamble sample selection approach to minimize duty cycle and reduce idle listening to achieve low power operation (an adaptive rate scheme). B-MAC allows for on-the-fly adjustment and bi - directional interoperability for system services to optimize performance, whether for throughput, latency, or energy savings. BMAC resulted in improved packet delivery rates, throughput, latency, and power consumption as compared to SMAC. Furthermore, B-MAC achieve computationally efficient deployment by exploiting CCA thresholds, which is further enhanced by many researchers (I. Ramachandran & Roy, 2006).

Although CCA's implementation is done at the PHY layer, its performance and complexity have the greatest impact on MAC metrics such as throughput and energy efficiency. Because of this cross-layer dependency, it is necessary to improve MAC protocol performance assessment by considering the specific details of CCA deployment that are naturally link-dependent. Mounib et al. (Mounib, Mouhcine, & Hussein T, 2012), presented a priority based backoff algorithm that maintains a node's transmission priority if a collision occurs and reported through CCA parameters. The proposed scheme outperformed the standard IEEE 802.11 based backoff strategy in terms of channel utilization and power conservation. Son et al. (K. J. Son, Hong, Moon, Chang, & Cho, 2016), investigated the energy threshold level of CCA to address the single power threshold measurement for channel activity. The proposed scheme provided two CCA thresholds to increase the detection accuracy. Along the same lines, King et al. (King & Roedig, 2018), proposed a differentiating CCA threshold measurement strategy, that could effectively assess various channel energy levels. The proposed scheme was directed towards indoor environments where local interference

may add to the difficulty, and thus the inaccuracy of CCA. The receivers running the proposed power differentiating clear channel assessment (P-DCCA) scheme can distinguish between various transmission characteristics. Through a comprehensive and detailed analysis of the proposed scheme on ContikiMAC protocol, the authors reported enhanced range, throughput while reducing energy consumption. Similarly, authors (Amirinasab Nasab, Shamshirband, Chronopoulos, Mosavi, & Nabipour, 2020) also proposed a CCA based strategy for ContikiMAC protocol for improved chancel assessment as well as reduced energy consumption. The proposed lightweight scheme reduced false wakeups, and idle-listening to reduce the overall energy consumption of the WSN. These techniques intrigued us to design a lightweight, CCA-based method for cross-layer optimizations of an energy-efficient WSN.

### 2.3.4 Data driven techniques

When using data driven techniques, the general concept is to keep the sensing accuracy for every sensor node at a reasonable threshold for the application while reducing the amount of redundant information as much as possible. Because the spatial and temporal correlation of sampled data is extremely strong, there is no need to communicate some of the unnecessary data (i.e., superfluous information) to the sink nodes, which saves time and money. A subset of data driven techniques is power efficient data acquisition schemes, which can be further subdivided into energy efficient data reduction schemes. Additionally, they can be divided into three types of active sampling protocols: adaptive sampling, hierarchical sampling, and model driven active sampling protocols. Whereas data reduction schemes are designed to reduce the amount of data transferred.

Energy-efficient data acquisition schemes mostly rely on adaptive sampling to accurately sense and process only the most appropriate information. Adaptive sampling

with identify the likelihood is discussed in (Vuran & Akyildiz, 2006), where the authors propose a spatial Correlation-based Collaborative MAC protocol (CC-MAC) wherein the Iterative Node Selection (INS) algorithm is located on sink nodes to minimize the number of nodes that have packets to report. Two CSMA-CA components are used in the CC-MAC protocol: Event-MAC (E-MAC), which is used to minimize superfluous data transmitted to sink nodes during the channel access phase, and Network-MAC (N-MAC), which is used to manage network routing by prioritizing route through packets over newly arrived packets. Along the same lines, in (Pourpeighambar & Sabaei, 2013) rate distortion (RD) theory was investigated for intelligent data gathering to reduce energy consumption in sensor networks. Zhou et al., proposed a novel energy-efficient scheme was introduced to reduce temporal redundancy between intra-clusters transmissions. Furthermore, the authors proposed an error threshold selection scheme to accurately model the effect of spatial data reduction in terms of accuracy. Finally, in a recent study on spatial-temporal approach for data collection (Tayeh, Makhoul, Perera, & Demerjian, 2019) tayeh et al., proposed to reduce the sampling and transmission rates while maintaining the overall data quality. Although data-driven techniques are promising, the implementation on resource-constrained sensor nodes is complex and challenging task. Furthermore, the energy conservation achieved by smart data collection is many folds less as compared to the other cross-layer schemes.

## 2.4 Battery Types and the need for characterization

WSN nodes are mostly powered by batteries where only a few specialised WSN applications can harvest energy. So, in a typical WSN deployment, system designers focus on battery voltage and energy footprint to estimate the device lifetime. Inadequate knowledge of battery chemistries, discharge patterns, environmental and operational

sensitivity, and failure processes leads to erroneous system models, affecting overall device lifetime. While battery manufacturers evaluate and classify batteries before transporting them to consumers, it is prudent for customers to test and qualify their own batteries to ensure safety and performance. For material testing, manufacturers usually use advanced testing techniques. These include X-Ray Absorption Fine Structure Spectroscopy (XAFS), Transmission Electron Microscopy (TEM), Fourier Transform Infrared Spectroscopy (FTIR), and Coherent Diffraction Imaging (CDI), to name a few. It can analyze battery chemical materials, transport phenomena, state of health, and expected material life. These techniques are complex, expensive, and time consuming, and may not be applicable to all batteries produced. Before deploying batteries in real-world applications, the system designer must perform discharge and failure analysis. These analyses are also required to remove battery memory, passivation, and hysteresis effects to estimate available battery capacity compared to vendor claims. Therefore, it is crucial to first understand the battery chemistries by investigating their chemical properties and responses in WSN applications.

### 2.4.1 Classification of Batteries

A battery contains one or more cells that undergo chemical reactions to allow electrons to flow through a circuit. Electrochemical batteries can be divided into four types. These broad categories are grouped by the materials used to design specific operations. There are primary cells, secondary cells, reserve cells, and fuel cells. For military and weaponry purposes, reserve cells are used, whereas fuel cells generate energy by using external active materials. Primary and secondary batteries (also known as rechargeable cells) are widely used in consumer electronics. Consumer electronics devices such as phones, computers, smart watches, and cameras prefer rechargeable secondary batteries. In applications like WSN, where maintenance, battery

management, and physical access are impractical, a node's battery capacity defines its life. As a result, primary batteries were used in such devices. Technology advancements and lower secondary battery prices have made both cell types popular in WSN applications. Table 2.1 presents few critical battery parameters that are often considered during system design process. Furthermore, other important battery parameters are detailed in *Appendix A*.

Table 2.1: Battery types, pros, and cons for consumer electronics

| **Parameters** | **Primary Batteries** | **Secondary Batteries** |
|---|---|---|
| Initial Cost | Low | High |
| Life-cycle Cost ($/kWh) | High | Lower life-cycle cost if charging and reusability is considered |
| Recyclability | Disposable | Special attention is required |
| Size and weight | Lighter and small | Typically heavier, but recent batteries with advanced materials are becoming smaller and lighter |
| Service Life | Longer life and good charge retention | Adequate service life but batteries have inferior charge retention |
| Discharge Rate | Designed for low discharge rate applications | Designed for low to high discharge rate applications |
| Maintenance | No maintenance required | Maintenance costs may increase exponentially based on applications |
| Applications | Limited range of applications | Suitable for a broad range of applications |

### 2.4.2 Types of battery chemistries

As previously mentioned, primary batteries may be used for WSN applications, however, keeping cost, rechargeability, and longer lifetime into consideration secondary batteries are preferred. There are several battery chemistries available in the market, however, in this research the following types were utilized:

- Lithium-Ion (Li-Ion)

- Lithium-Polymer (LiPo)
- Nickel-metal hydride (Ni-MH)
- Zinc-Manganese Dioxide (Alkaline)

They come in various sizes and shapes, but system designers must consider certain properties to choose the best battery for their application. This includes topics like material selection, design philosophy, testing, use of innovative materials, and battery optimization. This study's goal is to select the best battery chemistry for the WSN node's operational and environmental parameters.

### 2.4.2(a) Lithium-Ion batteries

One of the most common forms of rechargeable batteries is lithium-ion. There are numerous varieties of lithium batteries, but lithium-ion batteries are the most often used. Lithium-ion batteries are a form of rechargeable battery in which lithium ions migrate from the negative electrode to the positive electrode during depletion and back to the negative electrode during charging. In contrast to the metallic lithium utilized in non-rechargeable lithium batteries, lithium-ion batteries use an intercalated lithium compound as one electrode material. In comparison to other battery types, lithium-ion batteries have a high energy density, little or no memory effect, and low self-discharge. The performance and cost change depending on the use case. While Li-ion batteries based on Lithium iron phosphate have a lower energy density and are therefore safer due to a lesser risk of untoward events occurring, they are commonly utilized in powering electric tools and medical equipment. Lithium-ion batteries have the highest performance-to-weight ratio, followed by lithium-Sulphur batteries.

### 2.4.2(b) Lithium-Polymer (LiPo)

Lithium-polymer batteries are sometimes called lithium-ion batteries due to their similarity. In truth, Li-Po is not a new technology; in fact, it is a material extension and upgrade to Li-Ion batteries. These developments paved the way for Li-Po batteries

to become a viable alternative to Li-Ion batteries in a variety of applications (Torabi & Ahmadi, 2020).

Li-polymer batteries work and are built like Li-ion batteries. They work by intercalating lithium from positive electrode materials to negative electrode materials. The lithium polymer battery differs from other lithium-ion batteries in that it uses an electrolyte that is a polymer. Modern lithium polymer electrolytes use a gel for high room temperature conductivity. Most of these batteries have a micro-porous separator. Li-polymer cells have a thin and ‘soft' housing made of deep-drawn aluminium foil, like a pouch. The prismatic housing is easier and cheaper to build than Li-ion cell casings. The cells are light and thin, and come in a variety of shapes and sizes. Using soft-pouch cells allows for greater design flexibility in terms of shape, size, and the ability to build flat cells. These truly transform the use of small electronics like WSN and smart watches. Another advantage is that these cells have a higher energy density per weight. Furthermore, they have a very low self-discharge, making them an excellent contender for WSN applications.

### 2.4.2(c) Nickel-metal Hydride (Ni-MH)

NiMH batteries have a chemistry very similar to NiCad batteries, and their operating voltage is around 1.2V. Because NiMH batteries have a higher energy content than NiCad batteries, they are used in portable devices like phones, computers, and shavers. Although Ni-MH cells powered the first generation of electric vehicles, lithium batteries are now the industry standard. The negative electrode of Ni-MH batteries is made of metal hydride rather than cadmium. The electrolyte is the same as in NiCad cells. The negative electrode is made of two metal groups: AB2 metals like titanium and zirconium, and AB5 metals like nickel and lanthanum. All Ni-MH batteries now use AB5 metals due to their superior performance. They are used in devices that require

a lot of power. For the same size, a Ni-MH battery has two to three times the capacity and energy density of a lithium-ion battery. Unlike NiCad batteries, Ni-MH batteries do not have the "memory" effect. Their high self-discharge rate necessitates pre-conditioning before use.

### 2.4.2(d) Zinc-Manganese Dioxide (Alkaline)

It is a primary battery that gets its energy from zinc and manganese dioxide reactions. Less expensive than zinc–carbon or zinc chloride batteries, alkaline batteries have higher energy density and shelf life. Contrary to the zinc–carbon batteries' acidic ammonium chloride or zinc chloride electrolytes, the alkaline battery's electrolyte is alkaline. The capacity of an alkaline battery is three to five times that of an acidic battery due to the purer and denser manganese dioxide in an alkaline battery. The capacity of an alkaline battery is load dependent. The total usable capacity is determined by the application's cut-off voltage. These cells have roughly the same capacity as other acid-based cells under intermittent or continuous light loads. Continuous discharge capacity is lower on heavy loads. A typical alkaline battery provides nearly 1.5V, necessitating a series connection to raise the output voltage above 3V. For low-power applications like WSN, an alkaline cell can deliver up to 700mA without overheating. Estimating battery capacity and voltage per cell is difficult due to the reliance on load and lower voltage per cell.

### 2.4.3 Battery characterization for small-load applications

Most of the battery characterization focuses on material testing techniques that employ chemical and electrochemical testing, as discussed in Section 2.4. Most of these characterization methodologies are specific to industrial batteries that offer high energy density over higher discharge rates. However, this study aimed at finding the discharge behaviour of several battery chemistries over a range of temperatures under IEEE

802.15.4 prescribed load. Therefore, a review of battery characterization research is presented in Table 2.2 , that is applicable to small-load discharge characterization.

Table 2.2: Recent literature on battery discharge characterization.

| Reference (s) | Domain | Limitations |
|---|---|---|
| (da Cunha & da Silva, 2012) | Discharge pattern modelling | Limited to alkaline batteries with controlled parametrization of discharge patterns |
| (K. Chen & Li, 2014) | Discharge characterization, Controlled temperature test | Fixed temperature and C-rates |
| (A. Yang et al., 2019) | Degradation analysis, dynamic Peukert coefficient | Limited to lithium-based batteries, limited discharge rates |
| (Chin, Gao, & Zhang, 2020) | Electrothermal analysis, equivalent modelling circuits | Limited to Lithium-Iron Phosphate batteries for EV applications. |
| (L. Wang et al., 2020) | Discharge characteristics, statistical modelling | Limited to EV applications featuring high C-rates at ambient temperatures. |
| (Fru, Tsafack, & Tanyi, 2021) | Recovery effect, SOH estimation | Specific to lead-acid batteries, require extensive laboratory experimentation |
| (B. Xu et al., 2021) | Discharge characteristics, thermal characterization | Suitable for high discharge rate applications, limited C-rates and temperatures for characterization |
| (Xia & Qahouq, 2021) | Discharge pattern characterization | Limited to Lithium-based batteries operated at fixed loads under controlled environments |
| (Gungor, Cetkin, & Lorente, 2022) | Thermal and electrical characterization | Suitable for high discharge rate applications |
| (Ahmeid, Muhammad, Lambert, Attidekou, & Milojevic, 2022) | Discharge characteristics, SOH, EOL, and capacity estimation | Specific to high discharge rate applications |
| (J. Sun et al., 2022) | Battery ageing, initial discharge quantity estimation | Specific to Lithium-based batteries, complex computations not suitable for embedded platform |

The battery discharge capacity is highly correlated to the associated discharge rates which can be modeled using the Peukert coefficient (A. Yang et al., 2019). Therefore, the effect of discharge rates and ambient temperatures were investigated in this research. Finally, as described by (K. Chen & Li, 2014; Chin et al., 2020) the effect of temperature range at various SoC levels were investigated to report the battery capacity as reported in Chapter CHAPTER 3.

## 2.5 Battery performance and capacity estimation

A battery's performance is influenced by operational and environmental factors. The state-of-charge (SoC) of a battery is used to estimate its remaining capacity. Accurate SoC estimation ensures safe battery operation by preventing over-charging/discharging (Y. Wang et al., 2020). Although this information is an indispensable component of typical Battery Management Systems (BMS), WSN mostly require SoC information for device lifetime prediction. Many studies have used SoC calculations to predict remaining battery capacity at ambient temperatures and constant loads, resulting in similar battery discharge trends (Ali, Ishak, Bhatti, & Communications, 2021; Dunkels, Osterlind, Tsiftes, & He, 2007; Nguyen, Förster, Puccinelli, & Giordano, 2011; Rukpakavong, Phillips, & Guan, 2012). However, WSN deployments exhibit dynamic application-specific loads and environmental conditions that must be accounted for battery capacity estimation. Therefore, SoC and lifetime estimation in WSN should be calculated based on the tasks performed by the nodes (Piller, Perrin, & Jossen, 2001).

SoC is a relative measure of available battery capacity in relation to its rated nominal capacity. Therefore, it cannot be directly measured using electrical sensors. Furthermore, the relation between the observed battery parameters and estimated SoC

are highly non-linear, that are sensitive to temperature and current changes, thus affecting SoC estimation accuracy. In this regard, a lot of effort has been made to design robust, accurate, and computationally efficient SoC estimation algorithms (M.-F. Ng, J. Zhao, Q. Yan, G. J. Conduit, & Z. W. Seh, 2020). These methods can be generally classified into five categories, including: open-circuit voltage (OCV) based method, Ampere-hour (Ah) integration or Coulomb counting methods, electrical circuit models (ECM), filter-based methods, and data-driven based methods (Xiong, Cao, Yu, He, & Sun, 2018).

OCV is the most used method for SoC estimation in the laboratory environment. This method utilizes the open circuit voltages of the batteries at steady state to estimate the remaining battery capacity (Chaoui & Mandalapu, 2017; Gholizadeh & Salmasi, 2013; Y. Li, Guo, Qi, Guo, & Li, 2020; Z. Liu, Dang, & Jing, 2019). The one-to-one SoC-OCV maps the mathematical relation between battery capacity and available voltage that can be recorded at any temperature. However, the extended relaxation time (typically reported between 1h to 4h (A. B. Ahmad, Ooi, Ishak, & Teh, 2019; Ding et al., 2020)) required to reach steady state makes it unfit for real-time SoC estimations (T. Liu, Ge, Yang, & Wang, 2021; K. Singh, Tjahjowidodo, Boulon, & Feroskhan, 2022; C. Zhang et al., 2017). Ah integration method integrates the discharge current of the batteries over a period to estimate its capacity making it the most computationally efficient method (Duryea, Islam, & Lawrance, 1999; Hansen & Wang, 2005; He & Guo, 2019). However, it requires highly sensitive sensors to measure charge flowing through the battery, as well as a precise initial battery capacity. The SoC estimation accuracy reduces due to initial capacity estimation inaccuracies and battery charge flow variations. (Z. Li, Huang, Liaw, & Zhang, 2017; L. Liu, Chen, Wang, Lin, & Wang, 2012). ECM based techniques compute state equations to model the behavior of the

batteries under specific loads and environment conditions. The models replicate the electrochemical behavior within the battery by mapping mathematical relations. However, due to large number of complex battery parameters these models require a great deal of parameter extraction.

In contrast to the ECM models, filter-based techniques extend ECM state equations to perform online calculations using adaptive filters. Some of the widely used adaptive filters for SoC estimation include sliding mode observer (SMO), particle filters (PF), Kalman Filter (KF), and extended Kalman Filters (EKF). These filters are capable of high accuracy SoC estimation, but at the same time they are very sensitive to system noise. A non-Gaussian noise distribution and a lack of a precise system model leads to exponential performance degradation. Data-driven techniques are model-free and require a large dataset for system identification, mapping, and predictions tasks. These schemes involve fuzzy-logic, support vector machines (SVM), neural networks (NNs), process regressions (PR), and extreme learning machines (ELM) to name a few (Chandran et al., 2021; Zhou, Zheng, Pan, & Lu, 2021). The SoC estimation methods can be implemented using hardware, software, or even hybrid-based approaches (Buchli, Aschwanden, & Beutel, 2013; Hu et al., 2019; Y. Wang et al., 2020; Y. Zhang, Xiong, He, & Shen, 2017). The hardware-based estimation requires dedicated Integrated Circuits (ICs) to characterize real-time battery data (R. Jin, Che, Wang, Zhu, & Wang, 2015; Y. Wang, Yang, Zhang, & Chen, 2016). However, hardware-based approach may significantly increase the implementation cost of a WSN node. Therefore, software-based data-driven techniques are considered as a viable implementation option for WSN applications. Table 2.3 summarizes various SoC estimation techniques that piqued our interest and facilitated in the design of lightweight and adaptive SoC estimation algorithms.

Table 2.3: A review of SoC estimation techniques.

| Reference (s) | Domain | Limitations |
|---|---|---|
| (L. Li, Wang, Chao, Zhou, & Xie, 2016) | Gaussian process modelling based capacity prediction | Highly complex model that is not feasible for embedded platform. |
| (Yu, Tang, Tang, & Liu, 2018) | Voltage prediction, RUL estimation | Very limited test cases of UAV application. The true dynamic discharge load effect was not reported. |
| (D. Ren et al., 2019) | Remaining life estimation, load forecasting | Limited to EVs, applicable to applications where known load-profile is available. |
| (Eleftheroglou et al., 2019) | Capacity and EOL estimation, hidden Markov Model | Limited to high discharge rate applications using high energy density batteries. |
| (Song, Zhang, Liang, Han, & Zhang, 2020) | SOH Estimation, Deep learning neural networks | Limited to data collected from EVs, complex computation which is not suitable for embedded platform. |
| (Niri et al., 2020) | RUL estimation, Gaussian process-based modelling, | Limited to EVs application dataset for controlled dynamic load, highly complex computational model that is unsuitable for embedded platform |
| (Y. Jin, Zhao, Li, Liu, & Wang, 2021) | EKF algorithm, approximation model | Limited to Li-Ion batteries, Randomly defined discharge rate |
| (Jia et al., 2021) | SOH estimation, Ensemble learning | Limited to Li-Ion batteries, wavelet transforms, and ensemble learning require a lot of computational power. |
| (Shen, Wu, Kang, Guan, & Peng, 2021) | Capacity estimation, Particle filtering, variable discharge current | Specific to Li-Ion batteries only. Models only the difference in discharge cycles, not the variable discharge currents as is the case of WSN. |
| (Yuanyuan Li et al., 2021) | SOH estimation, feature selection | Limited to high discharge rate applications. |
| (T. Xu, Peng, & Wu, 2021) | SOH estimation, Capacity estimation, data-driven modelling | Limited to EV applications that have random variable current load profiles. |
| (Zuo et al., 2022) | SOC Estimation, Neural Networks | Limited to specific battery chemistry, C-rates, and discharge feature set. |

The correlation factor between dataset feature can greatly impact the convergence and accuracy of the models. Therefore, careful consideration is required during the feature selection process. In this study, the feature selection is based on the works presented by (Z. Deng et al., 2020; Yuanyuan Li et al., 2021; T. Xu et al., 2021). Furthermore, the selection of variable discharge currents technique as presented by (Shen et al., 2021) was considered to design the SoC estimation models. Finally, unlike (Niri et al., 2020) where multimodal algorithms were designed. In this research, as battery charging and self-recovery effects are not considered, the multimodality trajectories were ignored. This enabled to simplify the model design and for the embedded platform. The SoC estimation model, optimization and performance validation techniques are detailed in Chapter CHAPTER 4.

## 2.6 Energy estimation and WSN architecture design

Because WSNs are made up of hundreds, thousands, and in some cases tens of thousands of devices, the system design and deployment techniques differ from those used in traditional battery-powered electronics systems. Conventional battery-powered electronics are mostly subjected to similar load and environmental conditions, allowing for predictable operation. Furthermore, every conventional device is programmed to perform similar tasks, increasing energy consumption and thus lifetime predictability. WSN devices are typically deployed in a collaborative network setting, where data collection, sourcing, aggregation, relaying, and processing are handled by separate network nodes. This necessitates a heterogeneous load and operational profile for these devices, which varies their duty cycle and payload in most settings.

Furthermore, because WSN nodes are dispersed across vast geographical landscapes, the environmental impact further varies battery behaviour, affecting the

lifetime of these devices. Another important consideration is the task of each WSN node, which prevents detailed characterization of each node. Therefore, simulators are mostly used to evaluate some of the critical node behaviours, such as networking, routing, and data propagation. Because these tasks typically make use of on-board radios, the energy consumption profiles are largely determined by the intended platform and the duty cycle of these nodes. The estimation of energy consumption based on platform (for load profiles) and network duty cycle (for node behaviour) is a good starting point for estimating node renaming energy, and thus device lifetime. However, almost all the techniques suffer from inaccuracies that arise from a variety of reasons, which are discussed below.

### 2.6.1 Initial battery voltage estimation errors

To begin calculating energy consumption, almost every simulator assumes a constant battery starting voltage (as specified in its datasheet). As previously stated, batteries are highly non-linear devices that rarely provide voltages close to their specifications, which may result in significant inaccuracies. Furthermore, because no two batteries are alike, the simulators mostly assume an average voltage and nominal capacity (as per datasheet), which may differ greatly from the actual batteries. This leads to energy estimation errors that cannot be compensated for, resulting in inaccurate WSN node lifetime estimation.

### 2.6.2 Linear analytical battery modelling

Most simulators use analytical battery models to calculate energy consumption based on attached load and duty cycle. Using the initial erroneous voltage values (as described above), these models calculate the energy consumption for each cycle where the drop in battery energy appears to be linear. It was discovered that battery discharge is a highly non-linear operation, with nearly half of the battery discharge cycle falling

into the non-linear region. Therefore, assuming the current voltage level based on the previous cycle is nearly impossible, resulting in very large energy consumption inaccuracies that rise exponentially from start to finish.

### 2.6.3 Effect of discharge rate, and self-discharge on battery capacity

The majority of WSN OS, including Contiki OS, were designed for ultra-low-power applications, with nodes expected to run for years on AA batteries. The simulators, however, do not account for battery self-discharge rates in their default configuration. For example, AA batteries that are mostly alkaline lose about 2% of their capacity each year. Ni-MH-based batteries report a significant high self-discharge rate of 20% initially, then accumulating to 1-4% in subsequent days**.** (Rukpakavong, Guan, & Phillips, 2013). Not only is the battery capacity affected, but so are the voltages. Although self-discharging is a critical phenomenon that must be accounted for in design for accurate lifetime estimations, it is simple to be programmed to interpolate the self-discharge rates on the overall battery capacity reduction.

The effect of discharge rate on battery capacity is critical because WSN nodes are always in dynamic environments and modes with constantly changing load profiles. The capacity of the batteries is linked to its discharge rate, as it affects the internal electrochemical reactions. For instance, alkaline cells report a decline in battery capacity with an increase in discharge rate, and vice-versa. Whereas Lithium and Ni-MH based batteries report a decline in capacity when the discharge rate is reduced (Energizer, 2014; Panasonic, 2007, 2020). The effect of discharge rate is a complex process and cannot be accurately mathematically programmed. Therefore, the proposed scheme which is driven by actual battery discharge models at various discharge rates overcomes this problem.

### 2.6.4 Inability to record the impact of environmental factors on batteries

Finally, the effect of environmental factors such as temperature and humidity on batteries is rarely reported. Batteries are very sensitive to temperature changes, with lower voltage levels and lower capacities at colder temperatures. At higher temperatures, however, batteries provide higher voltages and increased capacities, which may result in thermal runaways and permanent battery failures. (Park, Lahiri, & Raghunathan, 2005) reported a similar pattern, with battery capacity increasing around 0.5 percent of SoC with temperature rise (from 20˚C to 60˚C) and decreasing to nearly 1.7 percent (when the temperature was dropped from 20˚C to -10˚C).

The former effect is more serious because it can have a direct impact on WSN node lifetime, whereas the latter is more serious when energy-harvesting techniques for battery charging are considered for WSN. Contiki OS and its simulator Cooja do not directly report the temperature's effect on WSN nodes. Furthermore, the Cooja environment's energy consumption calculation is linear. Therefore, the KiBAM technique was used for accurate simulated energy modelling. T-KiBAM, a KiBAM extension, was used to study the effect of temperature on batteries, and the proposed technique was used to compute real-time on-device estimation while accounting for temperature changes. These techniques are described in detail in Chapter CHAPTER 5.

## 2.7 WSN enhancements at PHY and MAC layers

Wireless sensor networks (WSNs) are mostly deployed in harsh environments for information sensing and relaying, where they co-exist with other networks, devices, and interference sources. IEEE 802.15.4 standard based WSN is the most common choice of ultra-cost, low-power and low-data rate networks operating at 2.5GHz. This frequency spectrum is prone to interference from co-existing networks, industrial and home appliances operating at or around the same frequencies (Elshabrawy, 2015; Tytgat

et al., 2012). A typical WSN with limited on-board power mostly relies on efficient protocols and algorithms to perform its task and extend its lifetime (S. Li & Kim, 2015). The radio transceivers on a WSN device consume most power compared to other components, such as processors, sensors, and actuators (Kia & Hassanzadeh, 2019; Toor & Jain, 2019). This implies the significance of efficient MAC layer protocols that control the radio transceiver operation of a WSN.

IEEE 802.15.4 based MAC protocols operate on carrier sensing for estimating the channel condition before initiating transmission, which is performed by Clear Channel Assessment (CCA) at the Physical (PHY) layer (I. Ramachandran & Roy, 2006; Ramadan, Dessouky, Abd-Elnaby, & Abd El-Samie, 2018). At the end of each CCA stage, the state of the channel (idle or busy) is monitored and reported back to the MAC layer, which decides whether to initiate the transmission or defer the transmission for a fixed period (commonly referred to as the back-off period) (King & Roedig, 2018). CCAs are performed on both the sender and the receiver side that serve a different purpose, but both impact node's throughput and energy consumption. The sender only transmits the packet if an idle channel is available, otherwise defers the communication attempt based on the fixed back-off period (W. J. W. p. c. Kim, 2013). Whereas the receivers remain in a sleep state to conserve energy and only wakeup periodically to check for incoming packets. The radio transceivers are programmed to probe the medium to perform CCA estimation, which is significant for the MAC layer to adjust the transmission schedules and radio duty cycles. The current transceiver implementations based on IEEE 802.15.4 only indicate the network's current state (idle or busy) based on a fixed threshold of received signal strength indicator (RSSI) of the medium.

The IEEE 802.15.4 based MAC protocols rely on PHY layer to investigate the state and quality of the link after which a transmission decision is made. The channel acquisition, listening and transmission decision is mostly based on the CCA results after which a transmission window and a backoff period is selected to mitigate packet collisions or completely avoid them, in an ideal scenario. The IEEE 802.15.4 protocol suggests two consecutive CCA symbols with a CCA interval of (8 symbols length) for channel availability. However, many protocols utilize these symbols in various ways, with additional symbols to increase backoff periods and reduce packet collision. Although the fundamental aim is to reduce packet collisions that require retransmissions, however, increasing the CCA intervals and backoff period may keep the channels busy and can also put the radios in listening state for prolonged periods that may consume additional power. It is, therefore, necessary to investigate how IEEE 802.15.4 utilizes its PHY layer parameters to facilitate CCA on MAC layers. Some of the key responsibilities of PHY layer is to manage the activation and deactivation of radio transceivers, packet energy detection (ED), Link Quality Indication (LQI) for channel statistics and Clear Channel Assessment (CCA) to respond channel activity.

The receiver energy detection (ED) is significant to estimate the received signal strength within 8 symbol periods during transmission. The ED enables the channel selection algorithm which results in packet collision avoidance. As the ED values are reported in integer multiples, the channel sensitivity threshold is monitored using Received Signal Strength Indicator (RSSI) to estimate any communications above the channel sensitivity region. This also enables to record the amount of received power levels above specified receiver sensitivity. The LQI reports the quality of the received packet. Typically, LQI parameters are measured by incorporating ED, RSSI and Signal-to-noise ratios (SNRs) values. LQI values are typically used by network and higher

layers to route the network packets effectively. In most of the communications involving backoff periods, LQI values are used to observe the channel interference and therefore to increase the backoff unless the LQI values have improved. Clear Channel Assessment (CCA) helps to identify the state of the channel, before and after the communication. Typically, IEEE 802.15.4 based CCA is performed in one of the three, or the combination of the following methods (Farahani, 2008b):

- CCA reports a busy medium if the receiver ED is above a specified threshold.
- CCA reports a busy medium if the received signals exhibit IEEE 802.15.4 PHY signal characteristics
- Or a combination of the above two methods, where ED is detected above threshold and Carrier Sense (CS) is also detected.

However, the channel state is highly link dependent and can vary significantly during heavy traffic and heavy interference scenarios. In contrast, if the current traffic pattern or interference behaviour is known, it is more likely that the nodes can schedule their transmission periods resulting in improved energy consumption. For instance, a transmitting side can reduce the back-off period during high traffic and heavy interference. Whereas receivers can adaptively change their duty cycles and wake up times to conserve more energy by staying in sleep states. In narrow-band WSN, transceivers commonly deploy non-coherent energy methods for CCA. Energy detection (ED) techniques are more comfortable to configure. It records the RSSI values of the signal's energy around the carrier, which is then compared with a pre-defined threshold value to identify channel state (De, Song, Xu, Shi, & Tan, 2013; Iyappan Ramachandran & Roy, 2007). However, this makes channel interference patterns

extremely challenging for low-power transceivers, such as CC2420 and CC2650 (Cabezas & Patrón, 2018).

The proposed adaptive CCA (A-CCA) adjusts the CCA periods based on the traffic patterns by reducing and extending the CCA periods, resulting in lesser false wakeups and time wasted in idle-listening. The transmission power is adjusted to a fixed threshold, which helps ease the receiver side by easier detecting an incoming packet. The receiver then adjusts its wakeup period based on the transmission duration between high power signals above the noise floor, which further improves its energy consumption. Finally, a two consecutive symbol periods at the end of transmission ensures significant backoff period to avoid any packet collisions. The receivers adjust its CCA, as soon as an increased transmission threshold is detected, thus further reducing channel interference and at the same time increasing the packet transmission probability. The proposed scheme is evaluated in Cooja Simulator as well as Texas Instruments based MSP430 platforms using CC1352R2 development boards that feature CC2650 radio transceiver (Instrument, 2018).

## 2.8 Summary

In this chapter the current state-of-the-art of WSN architecture, advancements and technical challenges at each layer were discussed. As MAC and routing layers directly control the traffic, thus the radio transceiver, it was concluded that a cross-layer optimization technique is required for WSN device lifetime enhancement. Furthermore, the impact of battery utilization on device lifetime was reported. In this regard, current state-of-the-art of battery characterization and capacity estimation techniques were discussed. Additionally, a roadmap to design lightweight, and scalable SoC estimation

algorithms were discussed. Finally, the impact of CCA, and the performance validation strategies for the proposed A-CCA algorithm were thoroughly discussed.

# CHAPTER 3

# BATTERY CHARACTERIZATION FOR WSN

Energy-aware cross-layer optimization requires real-time accurate knowledge of device's available energy. WSNs are mostly powered by various types of small batteries with limited capacity, that regulates the device lifetime. In this chapter a novel technique to characterize various battery chemistries specifically for low-power applications such as WSN is presented. To attain the maximum battery capacity, a five-step method was defined. Four battery chemistries including (Lithium-Ion, Lithium-Polymer, Nickel-metal hydride, and Zinc-manganese dioxide based Alkaline) were used to study the discharge characteristics. The discharge effects were observed at range of temperatures including (5°C, 15°C, 25°C, 35°C, and 45°C). The batteries were subjected to IEEE 802.15.4 protocol-based transceiver load profiles where the effects of load on battery voltage, state-of-charge (SoC), capacity, and surface temperatures were observed.

## 3.1 Characterization perspective and Safe Operating Area (SOA)

Batteries must be operated with care to avoid hazard and to increase their operating life. A safe operating area (SOA) is the range of operation within which a battery can operate at maximum capacity while avoiding the risk of leakage, fire, or explosion. In contrast to fixed intrinsic battery parameters, SOA is a guideline that must be followed to avoid risks and achieve the best results by operating the batteries at their maximum power points (MPP). Battery manufacturers provide SOA guidelines for safety and longevity, which are usually based on some of the most critical operational parameters of the batteries, such as:

- Battery Voltages (OCV and the terminal voltage)
- Temperature changes (charging, and discharging)

- Current drawn (attached load, and its effect on internal battery temperature)

The SOA region serves as a boundary for a battery's safe and optimal operating area. Unlike a fuse or a switch, which turns off an output when it goes outside its logical range, SOA defines a fuzzy range that can change over time. In fact, most advanced battery management systems (BMS) track the SOA of a new cell in real time and update it based on various applications and even time as the cell deteriorates. This phenomenon is explained in Figure 3.1, where a typical Li-Ion cell is investigated for its SOA based on the above stated parameters.

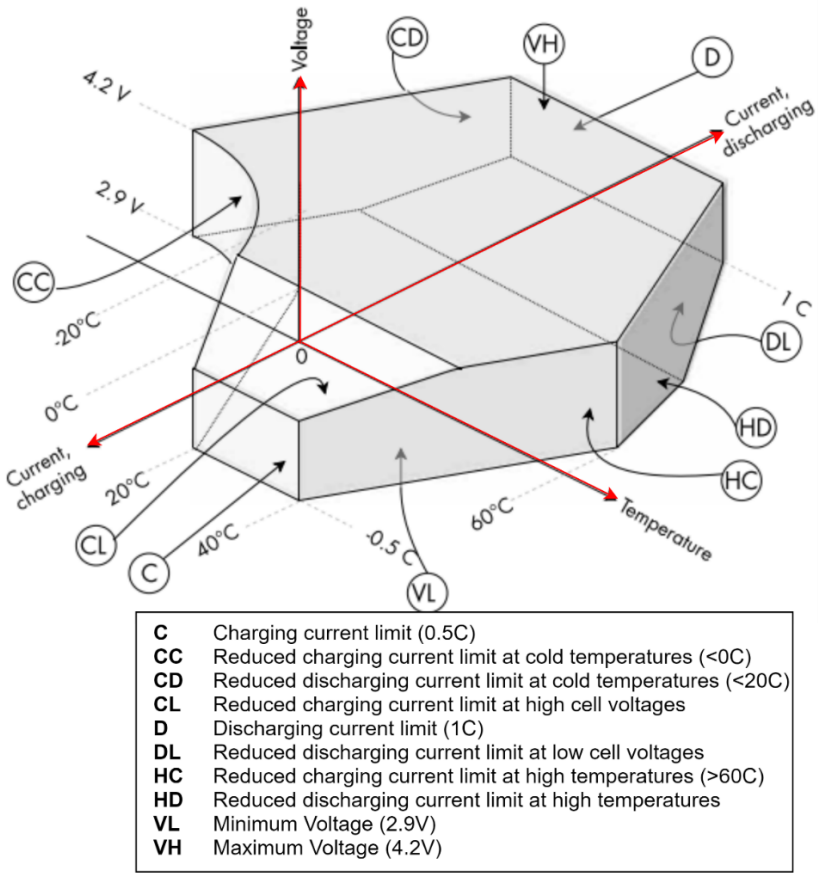



Figure 3.1: A typical operation of a Li-Ion cell under its SOA region (Andrea, 2020)

For instance, the battery under investigation can be operated at higher temperatures while still maintaining its SOA as well as pushing high discharge currents. However, exceeding this range or using the battery for an extended period of time can permanently damage the battery. These effects can gradually reduce battery life or, in some cases, like thermal runaway may destroy the cells. Identifying the end application

perspective is critical when characterization the cell. In this regard, (Andrea, 2020) presented a comprehensive cell characterization perspective, as given by Table 3.1.

Table 3.1: Application specific cell characterization perspective

| Specification | Description | Parameters |
|---|---|---|
| **Cell Model** | Nominal cell/battery characteristics and parameters as specified by the manufacturer | Capacity<br>Voltage range<br>Temperature range<br>Nominal and peak load currents<br>Discharge characteristics<br>Cycle life |
| **Battery Family characteristics** | Battery characteristics determined by the internal chemistry regardless of the physical properties including size and dimensions | Energy density<br>Specific energy<br>OCV Vs. SoC<br>Cycle life |
| **Operational characteristics** | Application specific operational range | SoC<br>Voltage range<br>Effective capacity<br>Cycle life |
| **Individual cell state** | Cell conditions during operation at a given moment in time | Voltage<br>SoC<br>Surface Temperature |
| Individual cell conditions | The current state of health of the battery due to long-term operation | Actual capacity<br>State of Health (SOH)<br>Current range |

In this study, several application-specific characteristics (as highlighted in Table 3.1) to see how four different battery chemistries (as given by Section 3.1) performed in terms of fundamental characteristics, operational requirements, and responses. For two reasons, individual cell conditions during operation were not monitored. For starters, SOH and DOD monitoring necessitates a costly and complex laboratory setup that may not be suitable for real-time applications. Second, because the end of a battery's capacity means the end of a WSN node's life, and there's usually no way to recharge it, investigating the battery's health isn't feasible.

## 3.2 Battery Characterization System Overview

Figure 3.2 depicts the experiment design and system flow for battery characterization experiments. The most critical aspect of this research was choosing two batteries. Most WSN research uses primary alkaline batteries, but alkaline based cells are known to be highly dependent on discharge current, which affects capacity. A slow discharge can extract almost all the battery's charge and energy. However, with an increase in the discharge rates, battery kinetics increases resulting in an abrupt loss of energy. Therefore, instead of using non-rechargeable alkaline cells, three other more recent battery chemistries were investigated. The details of the selected battery chemistries are presented in Table 3.2.

Table 3.2: Battery specifications used for characterization in the study.

| Tags | Manufacturer | Model | Type | Capacity (mAh) | Nominal Voltage | C-Rate |
|---|---|---|---|---|---|---|
| Batt1 | Powerizer | MH-AAA1000APZ (Corporation, 2015) | Nickel-Metal Hydride (Ni-MH) | 1000 | 1.2 V | 0.3-1C |
| Batt2 | Data Power Technology | DTP603450 (DPTL, 2017) | Polymer Lithium-Ion (LiPo) | 1000 | 3.7V | 0.2-1C |
| Batt3 | Panasonic | UF553443ZU (Panasonic, 2007) | Lithium-Ion (Li-ion) | 1000 | 3.6V | 0.2-1C |
| Batt4 | Energizer | LR-6 (Energizer, 2014) | Alkaline (Zinc, Magnesium Dioxide) | Variable, load dependent | 1.5V | 0.5-2C |

Various battery chemistries with equivalent capacity were used to establish similar discharge timings. However, because Ni-MH and Alkaline batteries have lower nominal voltages per battery, a battery pack with three cells each was built to match the nominal voltage of other batteries.

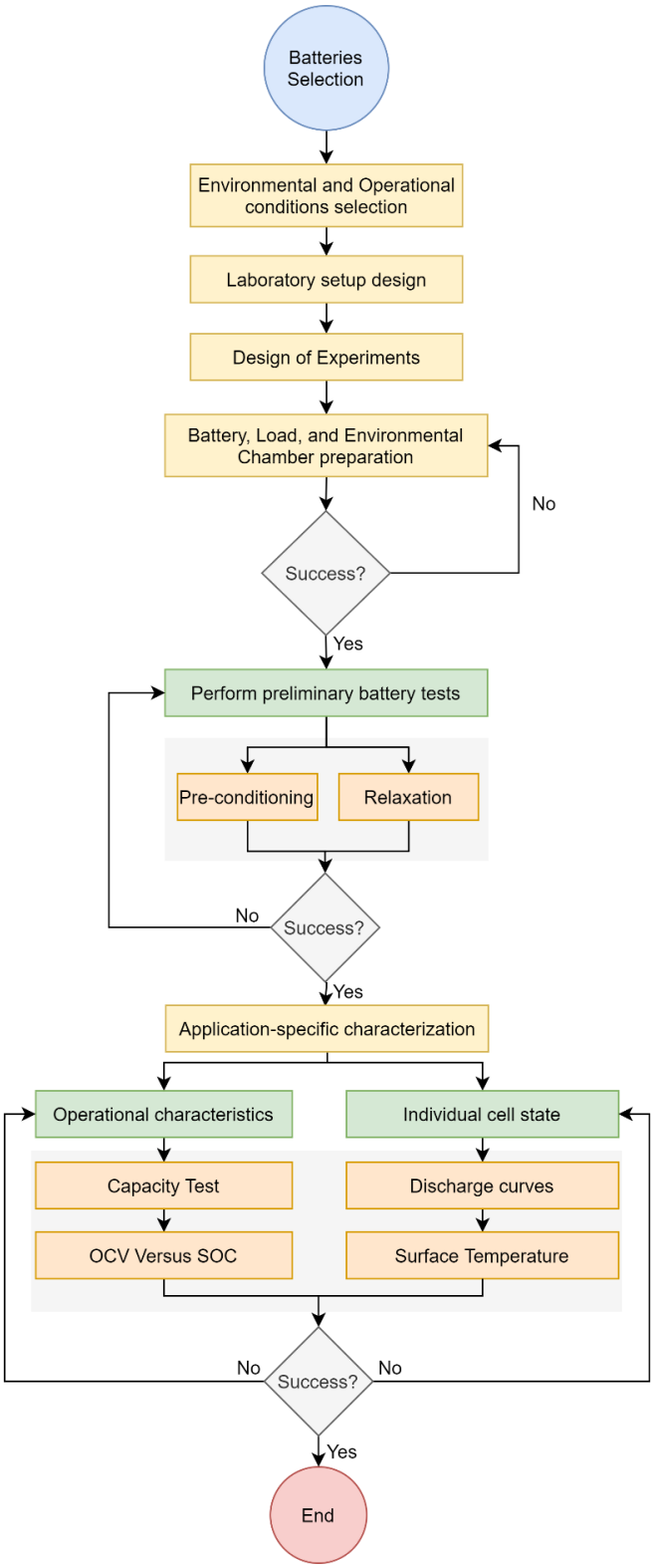



Figure 3.2: System design and experiment flow chart for battery characterization

### 3.2.1 Device Discharge Profiles

As previously stated, battery capacity is usually expressed in Ampere x Hours (Ah) or milliamps x Hours (mAh). Another intriguing method of gauging battery

capacity is to do so as a function of the time it takes to fully discharge a battery at a given discharge load. The battery capacity is usually expressed as C$x$ in these calculations, where $x$ represents the time in hours it takes to discharge a fully charged battery. C10, for example, would represent a specific battery capacity when discharged for 10 hours. When a battery is discharged at the manufacturer's recommended rate, the battery is more capable of delivering close to its nominal capacity. However, the result cannot be explicitly extended to other discharge profiles. Similarly, this notation is commonly used to calculate a battery's discharge rate. For example, dividing the battery capacity by time yields a discharge rate of C/10 or 0.1C (Farahani, 2008a).

### 3.2.2 Standards and protocols for Low-power and Low-rate applications

While the average device current consumption in short-range wireless sensor networks is low, the peak current can be quite high. The frequency of operation (duty-cycle) can also affect recovery times and battery capacity. To investigate the effect on all battery chemistries, standard device discharge profiles must be used. In this research, IEEE 802.15.4 protocol that was designed for low-power, low-rate application, such as WSN was investigated. The IEEE 802.15.4 protocol operates at 2.5 GHz and focuses on the Physical (PHY) and Medium Access Control (MAC) layers of the Open Standards International (OSI) stack. In theory, this protocol allows for data rates of up to 250 kbps, but in practice, peak currents of up to 50 kbps are easily achieved depending on the symbol rate. IEEE 802.15.4 also includes real-time guaranteed time slots (GTS), collision avoidance via CSMA/CA, and power management via link quality energy detection, all of which are important features for designing energy-conscious low-power applications.

In a typical WSN, the radio transceiver consumes the most amount of power as compared to the processing unit, and other peripherals. Therefore, IEEE 802.15.4 compliant radio transceiver were required that can be programmed to operate in low-power mode to conserve energy. In this regard, Texas Instrument's MSP430 platform based CC1352R2 development boards were used. These boards are powered by powerful 48-MHz ARM Cortex-M4 processors that can deliver ultra-low-rate applications to real-time multimedia IoT applications. These platforms have integrated IEEE 802.15.4 compliant CC2652 radio transceiver that can be programmed to operate in various power settings (Instrument, 2020a). Furthermore, the on-board 12-bit Analog-to-Digital (ADC) converter provides sensitive and accurate on-board measurements. The ARM Cortex M4 micro-controller (MCU) operates in two states: Active, and low power, whereas the radio transceiver operates in Idle (low-power listening), Transmit (*Tx*), Receive (*Rx*), and Sleep modes (Instrument, 2021). Therefore, four radio transceiver, and two MCU power states must be accounted for in order to investigate the battery discharge behavior under these loads. However, it can be argued that in most of the situations, no two states can be simultaneously in operation. For instance, a radio cannot be in Tx and Idle state in the same moment of time. Similarly, it is unlikely to observe an MCU in Active and sleep modes simultaneously. Therefore, some of the most important power consumption metrics for the above-mentioned states are given in Table 3.3. The manufacturer's current consumption profiles are unlikely to be achieved due to component integration, bus bar noise, and passive element deterioration (such as resistors). Sleep states with low micro-ampere current utilisation were also ignored, as further improvement would necessitate IC design optimization. So, a variety of power consumption scenarios were chosen to match the device's power profiles.

Table 3.3: CC1352R2 platform power consumption specifications.

| Working State | Value | Unit |
|---|---|---|
| CC2652 radio transceiver current consumption | | |
| Transmission State (Tx) | | |
| Power = 0 dBm | 8.1 | mA |
| Power = 10 dBm | 14.3 | mA |
| Power = 14 dBm | 24.9 | mA |
| Receive State (Rx) | | |
| Power = 0 dBm | 15.8 | mA |
| Texas Instrument CC1352 R2 Development board current consumption | | |
| Sleep Mode | 110 | µA |
| Active Mode (at 48 MHz) while using on-board sensors | 2.9 | mA |
| Active Mode without using on-board sensors | 50 | µA |
| Discharge currents used for battery characterization | | |
| Radio Receiving at normal power levels with MCU in Active state | 20 | mA |
| Radio Transmitting at normal power levels with MCU in Active state | 30 | mA |
| Radio Transmitting at boosted power levels with MCU in Active state with all on-board sensors running | 50 | mA |

### 3.2.3 Environmental Chamber

Environmental testing determines a component's environmental performance. Temperature, humidity, and other variables can cause device failure. Batteries are the future of energy storage, and their safety and reliability must be tested. Batteries can fail due to overcharging, overheating, or membrane rupture. Each of these failures carries specific safety risks to the product being tested. Extreme temperatures must not cause leakage or explosion of batteries. Temperature fluctuations can reduce cell voltage and even rated capacity. Thermal chamber testing can identify a battery's heat and stress tolerance. Environmental chambers must have safety features to prevent

explosions and protect operators if batteries fail. Therefore, thermal chambers with thermal runaway detection are preferred to avoid explosion or fire hazards. In this perspective, the Memmert Eco Climate Test Chamber (Memmert, 2021) was used in this research that is designed to produce reproducible and accurate test results. Application-specific heating enables tailored temperature change rate performance to match the needs of individual testing. The battery discharge profiles were observed at five different temperature ranges (including 5°C, 15°C, 25°C, 35°C, and 45°C). For temperature and moisture-controlled testing near and below 0°C requires the use of special purpose batteries, and node housing to prevent from cold temperature and moisture, which is beyond the scope of this work. The discharge setup for preliminary battery testing under controlled environmental conditions is presented in Figure 3.3.




Figure 3.3: Battery pre-conditioning tests in a thermal chamber

### 3.2.4 Battery discharge system

The CC1352R2 (Instrument, 2020b) development board from Texas Instruments was utilized to configure the discharge current profiles and record the battery statistics. The discharge currents utilized in these tests were 20mA, 30mA, and

50mA to match the operating states of the IEEE 802.15.4 protocol-based CC2652 radio transceiver (Instruments, 2009), as previously mentioned in Table 3.3. These results correspond to discharge rates of 0.02C, 0.03C, and 0.05C respectively. A discharge pulse at every 15 seconds enabled the load, and the analog voltage changes were then transformed to digital via the on-board 12-bit Analog to Digital converter (ADC). The battery discharging setup is illustrated in Figure 3.4.

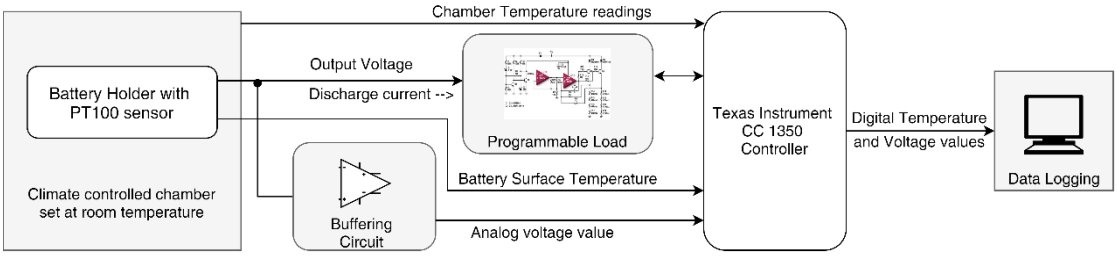


Figure 3.4: Battery discharge cycling and data logging schematic

The batteries were driven according to the IEEE 802.15.4 device discharge profiles. This required 15 drive cycles per battery at varying loads and temperatures to extract features. Three extra drive cycles were used to enhance the effect of noisy measurements, which are common in real-world deployments. An additive Gaussian noise with a mean of 0 and a standard deviation of 1 to 2 percent was applied to voltage and temperature measurements in this case. This added three drive cycles per battery. To estimate network response in a noisy environment caused by instrument or battery changes, the augmented dataset can be used. These datasets are required to train the Machine Learning (ML) models that are further detailed in Chapter 4. The drive cycles used in this research are presented in Table 3.4.

Table 3.4: Tested drive cycles and corresponding discharge profiles

| Test | Use | Discharge profile | Temp. | Battery |
|---|---|---|---|---|
| Cycle 1 | Training/Validation | 20mA pulse at every 15 seconds | 5°C | Lithium-Ion, Lithium-Polymer, and Nickel-metal hydride |
| Cycle 2 | Training/Validation | 20mA pulse at every 15 seconds | 15°C | |
| Cycle 3 | Training/Validation | 20mA pulse at every 15 seconds | 25°C | |
| Cycle 4 | Training/Validation | 20mA pulse at every 15 seconds | 35°C | |
| Cycle 5 | Training/Validation | 20mA pulse at every 15 seconds | 45°C | |
| Cycle 6 | Training/Validation | 30mA pulse at every 15 seconds | 5°C | |
| Cycle 7 | Training/Validation | 30mA pulse at every 15 seconds | 15°C | |
| Cycle 8 | Training/Validation | 30mA pulse at every 15 seconds | 25°C | |
| Cycle 9 | Training/Validation | 30mA pulse at every 15 seconds | 35°C | |
| Cycle 10 | Training/Validation | 30mA pulse at every 15 seconds | 45°C | |
| Cycle 11 | Training/Validation | 50mA pulse at every 15 seconds | 5°C | |
| Cycle 12 | Training/Validation | 50mA pulse at every 15 seconds | 15°C | |
| Cycle 13 | Training/Validation | 50mA pulse at every 15 seconds | 25°C | |
| Cycle 14 | Training/Validation | 50mA pulse at every 15 seconds | 35°C | |
| Cycle 15 | Training/Validation | 50mA pulse at every 15 seconds | 45°C | |
| Augmented 1 | Testing | 20mA pulse at every 15 seconds with added Gaussian noise | 25°C | |
| Augmented 2 | Testing | 30mA pulse at every 15 seconds with added Gaussian noise | 25°C | |
| Augmented 3 | Testing | 50mA pulse at every 15 seconds with added Gaussian noise | 25°C | |

### 3.2.5 Battery Gauging – Test Procedure, Cycling and SoC Measurement

The test procedure used during the entire battery discharge measurements phase is as follows: (i) Fully charge the battery based on the prescribed charging mechanism (ii) allow the batteries to rest for a period of 4 hours to achieve its steady state (iii) set the temperature of thermal chamber based on the drive cycle requirements (iv) place the battery inside the thermal chamber for 30 minutes for its temperature to be stabilized

(v) perform the drive cycles until the battery reaches its cut-off voltage. The experiments were repeated for all individual drive cycle settings. For room temperature and higher values, the battery internal resistance is low, resulting in a very efficient discharge operation. However, at lower temperature, especially 5°C, battery voltages periodically fall to its cut-off voltage due to higher internal resistance. To avoid any faulty observations, the experiments were repeated with additional batteries, following the above-mentioned procedure. To estimate the current flowing out of the battery, and the voltage across its terminals, LTC4150 (Technology, 2014) integrated circuit was used. LTC4150 is a coulomb counter battery gauge designed for low-power applications. The device measures the current flowing out of the battery by monitoring and integrating the voltage developed across a sensing resistor. An integrated voltage-to-frequency converter transforms this voltage into pulses, that record fixed quantities of charge flowing out of the battery and drives the $\overline{\text{INT}}$ pin to communicate with CC1352R2's ADC. In addition, a PT100 thermal sensor was mounted to the battery surface that recorded temperature values during the entire discharge cycle.

The SoC and remaining battery capacities were computed after every $\overline{\text{INT}}$ pulse until the battery is completely discharged (set to a cut-off voltage of 2.5V). In these experiments, the voltage-to-frequency gain of LTC4150 ($G_{VF}$) was used to compute the amount of intended charge flow per (Ah). The device frequency can thus be calculated using Equation (1).

$$f = G_{VF} \times |V_{SENSE}| \quad (1)$$

where output frequency (f) can be measured by interrupts per second, and $V_{SENSE}$ is the recorded potential drop across the sensing resistor. Using Ohm's Law, ($V_{SENSE} = I_{Batt} \times R_{SENSE}$), the frequency of LTC4150 can be given as Equation (2).

$$f = G_{VF} \times |I_{Batt} \times R_{SENSE}| \quad (2)$$

Since the current flowing through the sense resistor is integrated over time, the amount of charge ($Q = I \times t$ coulombs) per $\overline{\text{INT}}$, can be derived as per Equation (3).

$$One\ INT = {}^{1}/_{|G_{VF} \times R_{SENSE}|} \quad (3)$$

Battery capacities are normally computed in (Ah), which is the amount of ampere current flowing through over an hour. Therefore, a 1Ah battery can provide 3600 Coulombs of charge. This charge information was used to compute one quantum of charge flowing out of the battery at each $\overline{\text{INT}}$. Substituting this into Equation (2), yields Equation (4).

$$One\ INT = {}^{1}/_{3600 \times |G_{VF} \times R_{SENSE}|} \quad Ah \quad (4)$$

Using the LTC4150 specifications for device gain and Equation 4, a charge quantum flowing out of the battery was computed ($One\ INT = 0.6144C, or\ 0.1707mAh$). In other words, for 1000mAh batteries, a total of 5859 $\overline{\text{INT}}$ was recorded by the microcontroller. In a typical battery discharge experiment, the battery discharge curves depict an exponential, followed by a flat, and another exponential region. It is due to this non-linearity, SoC estimation is preferred over available voltage measurements. As the battery discharges, the SoC is estimated along with the precise remaining capacity enabling it to focus on a single data point in time.

## 3.3 Experiment Analysis

### 3.3.1 Battery Characterization procedure

The proposed characterization method was designed to obtain performance parameters for specific battery chemistries. Using load currents and the effect on battery

surface temperature were also studied. Figure 3.5 illustrates the proposed methodology steps that were used in this analysis. The pre-conditioning and relaxation tests were required prior to each discharge process, hence they are addressed here, whereas the remaining tests are discussed in the part that follows.

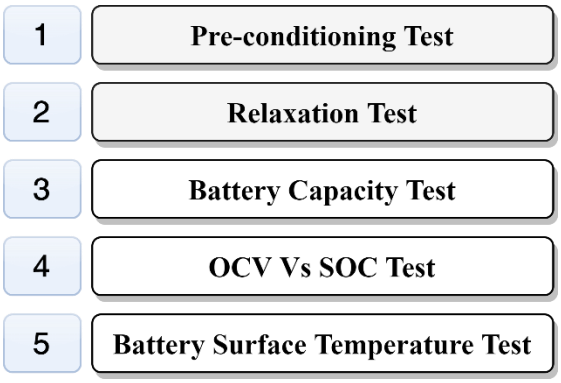



Figure 3.5: Five-step test sequence for effective battery characterization.

### 3.3.2 Pre-conditioning Test

In battery testing, current is usually expressed as a function of capacity. This is because the capacity is affected by the discharge rates. A constant current discharge is commonly used to quickly drain a battery. This is rated potential Cn, where n is the time base. To compare test results accurately, all measured cells must have the same cell potential. Preconditioning serves two purposes in this study. First, it removes any cell passivation or memory effect. Second, repeated charging and discharging stabilises battery capacity. To keep electrolyte porosity during the first few high current discharge cycles. The pre-conditioning tests used five consecutive 1-C (1A) charge and discharge cycles at room temperature (Stroe, 2014). Alkaline cells were not pre-conditioned because they were not rechargeable and there was no knowledge of their nominal capacity. Figure 3.6 discusses the pre-conditioning of the batteries used in this experiment (further referred to using their Tag names as given by Table 3.2).

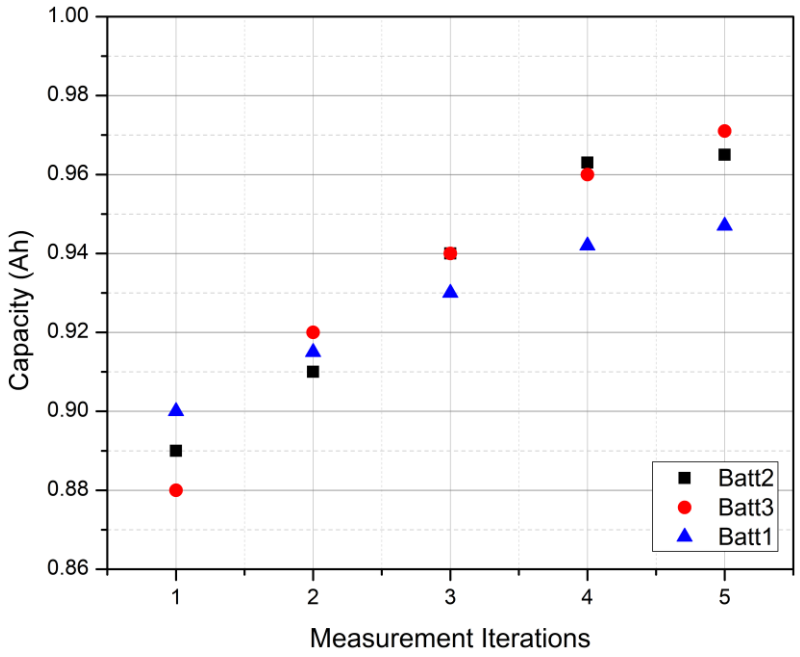


Figure 3.6: Discharging Capacity of batteries during preconditioning tests.

Initially, the lithium-based cells had a lower capacity, but after the third iteration, the battery capacity stabilised. It shows a small monotonic increase in battery capacity after stabilisation. Furthermore, the lithium-based cells show less than 4% heterogeneity over two consecutive discharge cycles. Also, lithium-ion batteries had a stabilised capacity of 960 mAh, 4.16 percent less than rated. Throughout the experiment, the Ni-MH cells' capacity increased monotonically. Despite this, capacity changes were less than 3%. Ni-MH chemistries had lower battery capacity than Lithium-based chemistries due to higher self-discharge rates.

### 3.3.3 Relaxation Test

Internal battery phenomenon such as polarization and mass transfer occur during the rest period and have very slow dynamics resulting in voltage fluctuations. The relaxation phase is the period following discharge during which there is no load attached and the battery voltages gradually approach a stable state (Bohlen, 2008; A. Li, Pelissier, Venet, & Gyan, 2016). Precise OCVs at various SOCs are required to accurately estimate battery SOC. To achieve thermodynamic stability, a relaxation time

between voltage measurements is required in no-load conditions. The longer the relaxation period, the closer the battery voltage is to OCV. The battery voltage is still changing hours later. Electrochemical Impedance Spectroscopy (EIS) measures minute changes in battery impedances during relaxation. In (Michael A Roscher & Sauer, 2011), the battery voltage can be stabilized beyond 24 hours relaxation period. Many studies found that after 5 hours, the change in battery impedance had slowed dramatically. However, it is uncertain whether it has slowed considerably enough that any future impedance changes are inconsequential (Messing, Shoa, & Habibi, 2019; X. Xu et al., 2019).

However, no single relaxation period for different battery chemistries has been documented. To that end, the batteries were fully charged before being discharged at three separate SOC levels (corresponding to 90%, 50%, and 10%, respectively). For each set of battery discharge trials, a variety of specified relaxation times (10 min, 1h, 2h, 4h, 8h, and 24h) were used. During this relaxation period, one-minute resolution voltages were compared to OCV measurements. During the relaxation stage, the batteries were considered quasi-stabilized. These trials helped us find the optimal relaxation time for future low-current discharge experiments. The OCV error between stabilized battery states can be computed from Equation (5).

$$\delta_{OCV} = \frac{|V_{\delta t} - OCV|}{OCV} \times 100 \tag{5}$$

Where $\delta_{OCV}$ and $V_{\delta t}$ represents OCV error and measured voltages over time intervals. The obtained results, along with the OCV errors are reported in Table 3.5. Shorter relaxation periods have a significant impact on Lithium-based battery chemistries, where OCV voltages are significant.

Table 3.5: Recorded voltage measurements at SOC (10%, 50%, and 90%) after relaxation period.

| Battery Relaxation Time | SOC (10%) | | SOC (50%) | | SOC (90%) | |
|---|---|---|---|---|---|---|
| | $V_{\delta t}$ (V) | $\delta_{OCV}$ (%) | $V_{\delta t}$ (V) | $\delta_{OCV}$ (%) | $V_{\delta t}$ (V) | $\delta_{OCV}$ (%) |
| **Li-Ion (Batt3)** | | | | | | |
| 10 min | 3.68 | 5.154 | 3.7 | 4.639 | 3.73 | 3.865 |
| 1 h | 3.72 | 4.123 | 3.75 | 3.350 | 3.78 | 2.577 |
| 2 h | 3.758 | 3.144 | 3.792 | 2.268 | 3.81 | 1.804 |
| 4 h | 3.798 | 2.113 | 3.816 | 1.649 | 3.847 | 0.850 |
| 8 h | 3.867 | 0.335 | 3.871 | 0.231 | 3.875 | 0.128 |
| 24 h | 3.88 | 0 | 3.88 | 0 | 3.88 | 0 |
| **Li-Po (Batt2)** | | | | | | |
| 10 min | 3.71 | 5.597 | 3.76 | 4.325 | 3.79 | 3.562 |
| 1 h | 3.771 | 4.045 | 3.79 | 3.562 | 3.801 | 3.282 |
| 2 h | 3.891 | 0.992 | 3.847 | 2.111 | 3.847 | 2.112 |
| 4 h | 3.899 | 0.7888 | 3.882 | 1.221 | 3.889 | 1.043 |
| 8 h | 3.92 | 0.254 | 3.921 | 0.229 | 3.93 | 0 |
| 24 h | 3.93 | 0 | 3.93 | 0 | 3.93 | 0 |
| **Ni-MH (Batt1)** | | | | | | |
| 10 min | 3.842 | 3.467 | 3.871 | 2.738 | 3.879 | 2.537 |
| 1 h | 3.876 | 2.613 | 3.882 | 2.462 | 3.894 | 2.160 |
| 2 h | 3.934 | 1.155 | 3.883 | 2.437 | 3.929 | 1.281 |
| 4 h | 3.9718 | 0.226 | 3.973 | 0.175 | 3.979 | 0.025 |
| 8 h | 3.98 | 0 | 3.98 | 0 | 3.98 | 0 |
| 24 h | 3.98 | 0 | 3.98 | 0 | 3.98 | 0 |

The computed errors increase between 10 min and 4 hours. However, between 8 and 24 hours, the batteries appear to be quasi-stabilized. The OCV measurements of Ni-MH batteries were slightly less variable, reaching steady-state in 4 hours. So, for Lithium and Ni-MH batteries, a relaxation time of 8 and 4 hours was chosen. It was later used to characterise battery discharge performances.

### 3.3.4 Battery Capacity Test

The capacity of a battery is affected by both load and temperature. Furthermore, deep charge and discharge cycles degrade battery capacity. Since the end of life of WSN nodes is determined by a fully discharged battery, this work does not address battery health and ageing. The capacity test approximated discharge capacity under various loads (0.02C, 0.03C, and 0.05C). These tests were done in a climate chamber at room

temperature to avoid affecting the batteries. The dependence of discharge capacity on C-rates is reported in Figure 3.7. It was observed that the discharge capacity of all the batteries decrease with the addition of load. Other than Alkaline cells, the discharge profile of the batteries shows a small monotonic decrease in capacity. This effect was investigated further by measuring the Peukert coefficient, which is given by Equations (6) and (7).

$$t = H\left(\frac{C}{IH}\right)^{k} \tag{6}$$

$$C_p = I^k t \tag{7}$$

Where $t$, $H$, $C$, $I$, $k$, and $C_p$ represents time in hours, the discharge time based on ampere-hour specifications, battery capacity, discharge current, Peukert exponent, and nominal discharge capacity at one-ampere hour respectively. The Peukert number was found to be close to 1 for Lithium-ion and lithium-based batteries, exhibiting almost less than 1% reduction in discharge capacity. However, Ni-MH battery reported a Peukert number between 1.4%, which agrees with capacity reduction as the load increases. Furthermore, it is important to consider that the Peukert Law does not account for self-discharge rate, which was found negligible after each relaxation period.

The Ni-MH battery had a nearly 80% flat region for 0.02 and 0.03C discharge rates. Its capacity decreases with increasing load. NiMH batteries' discharge behaviour is well-suited to today's consumer devices, especially those requiring a constant voltage for extended periods of time or rapid rate discharge. The capacity of a new, properly conditioned battery discharged at room temperature with constant current helps normalise capacity calculations. Charge rate affects capacity ratings because capacity depends on discharge rate. Ni-MH batteries have a low internal resistance due to their wound construction, improved connections, and large electrode surface area (IR). Ni-MH batteries have excellent high-rate performance due to low battery IR.

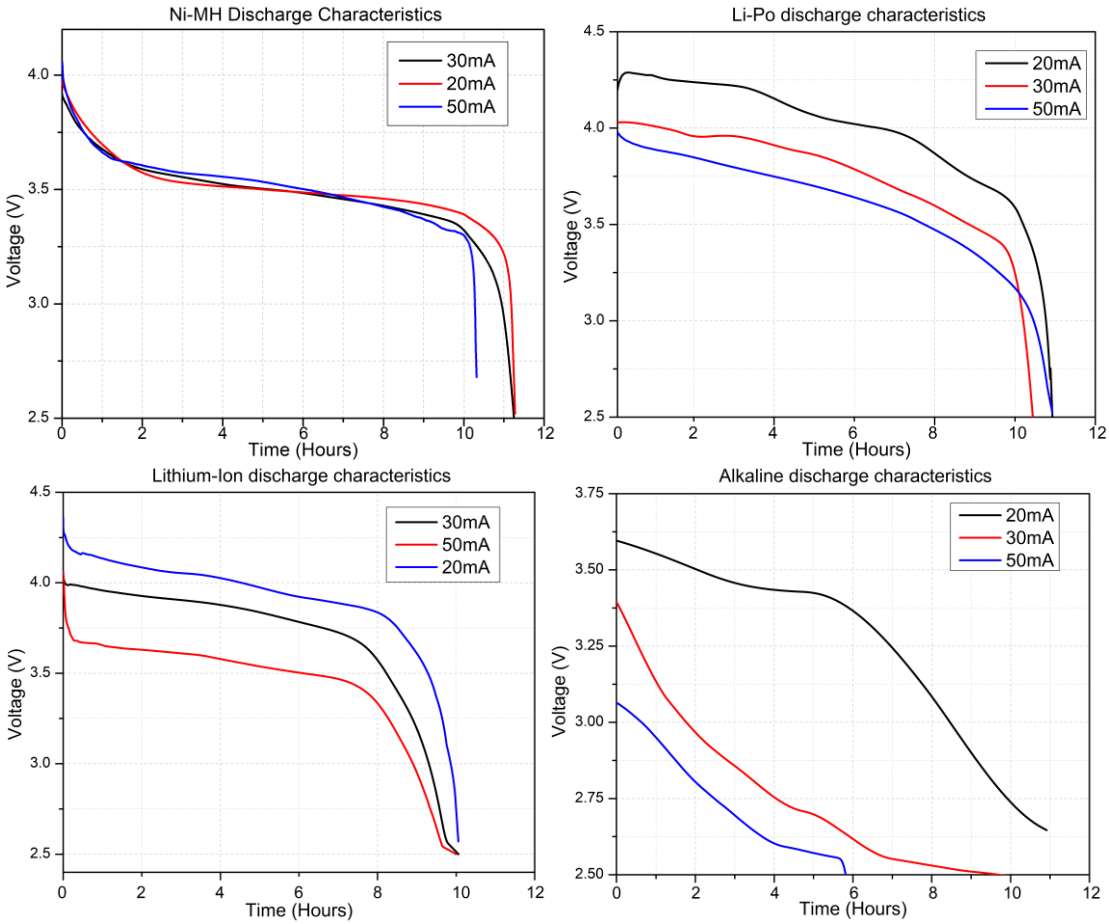



Figure 3.7: Discharge characteristics of the batteries under IEEE 802.15.4 discharge profiles.

The discharge voltage profile is affected by environmental factors such as temperature and discharge rate. Normal conditions result in a flat voltage curve suitable for electronics applications. This explains the flat discharge curves. Like lithium batteries, nickel-metal hydride batteries have a sharp "knee" near the end of discharge. Furthermore, rates less than 1C have no effect on discharge curve shape; rates greater than 1C cause both the beginning and ending transients to consume more discharge length. Finally, discharge rates less than 0.5C have no effect on capacity. Significant voltage reductions occurred at 0.5C and above. Due to a voltage drop, the capacity may be reduced.

Moreover, both lithium-based battery chemistries provided a flat area while keeping capacity. Unlike lithium-ion batteries, lithium-polymer batteries lose capacity

as the load increases. The capacity reduction is marginal for WSN application load profiles due to this load dependency. The capacity of alkaline-based cells decreases significantly as the applied load increases. Also, the non-linearity of the discharge curve makes estimating available energy, SOC, and capacity difficult.

Small cell packs are designed for low-power Ni-MH and Alkaline batteries. Providing the required device voltages requires charge balancing, which is difficult and inefficient for WSN applications. Even a single cell exceeding cut-off voltage causes premature battery cut-off. System designers must accurately estimate available battery energy and its discharge profile. Because WSNs have limited energy, SOC estimation is critical for energy-conscious applications. Because the supply voltage remains constant during discharge, a flat discharge curve simplifies device configuration. Cells with a sloping discharge curve deliver less power over time. End-of-cycle issues for high-power applications. A voltage regulator may be required if the slope is too steep for low-power applications that require a constant supply voltage.

### 3.3.5 Open Circuit Voltage Versus State-of-Charge

OCV, like most electrochemical battery parameters, is affected by operating conditions such as (temperature, SOC, etc). Therefore, the primary aim of these experiments was to determine the OCV dependence on battery SOC while maintaining a constant temperature during discharge. The OCV was calculated in 5% increments using the methods described in (Michael A. Roscher, Bohlen, & Vetter, 2011) and is reported in Figure 3.8. The OCV is affected by thermodynamical processes in addition to its utilizable ability. Overvoltage and OCV hysteresis found during charging and discharging are attributed to battery losses. OCV hysteresis is a common occurrence in Ni-MH and Lithium-ion batteries. Thermodynamic entropic effects, mechanical stress,

and microscopic distortions within the active material particles all contribute to hysteresis (Ovejas & Cuadras, 2019).

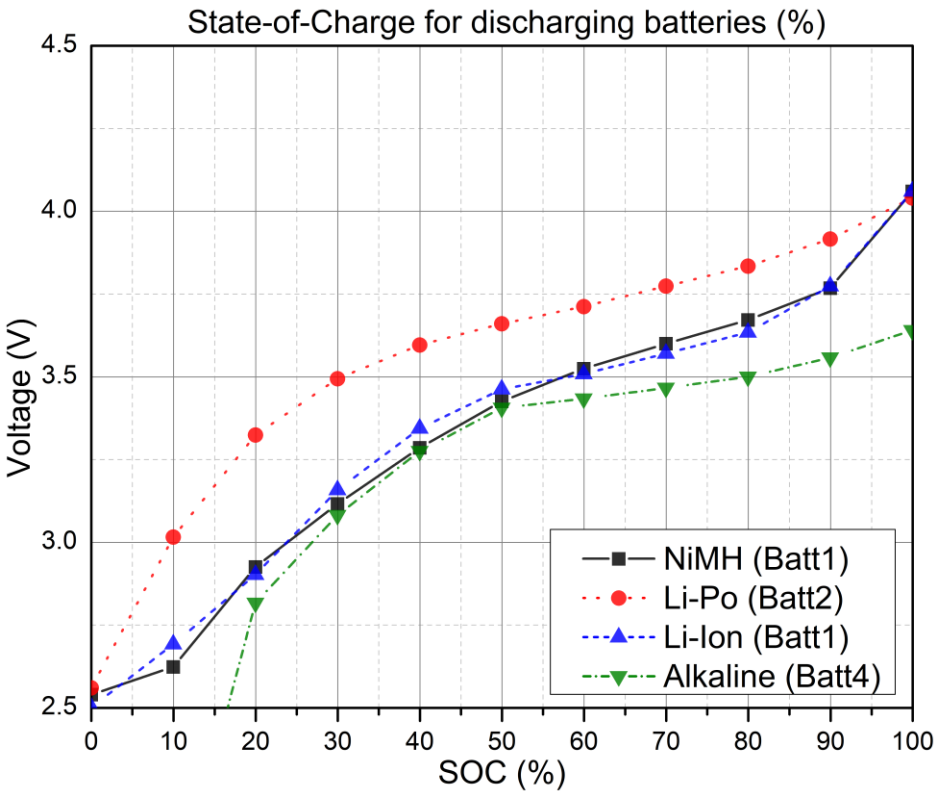



Figure 3.8: OCV measurements against battery SOC while discharging.

Lithium and Ni-MH batteries exhibited hysteresis around their flat SOC area. This is mostly due to the inherent structure of active battery materials, which undergo chemical transformations during operation. The hysteresis effect is studied in charge balancing systems for high precision, high power batteries. For two reasons, it is largely ignored in WSN implementations. First, hysteresis is minor. As a result, complicated and costly charge balance circuits are avoided. The OCV measurements are also important in the estimation of SOC. Each chemistry's discharge fingerprint is unique. The flat discharge curve of nickel and lithium-based batteries renders voltage-based SOC ineffective. Li-Ion, Li-Po, and Ni-MH batteries have linear discharge voltage curves, with nearly 80% of stored energy remaining in the flat voltage profile. In the flat zone, a small change in voltage can result in a large change in estimated SOC.

### 3.3.6 Battery Surface Temperatures under Load

The thermal effect of applied load on batteries has been extensively studied. EVs, solar energy, and grid storage all show an increase in battery temperature. Most of these thermal effects are studied in the lab by measuring the battery's core temperature. To avoid thermal runaway, most low-cost consumer products use battery surface temperature to regulate thermal impact. The effect of a lower attached load on battery temperature has not been reported.

In this regard, all the batteries were evaluated at ambient temperatures inside environment-controlled chambers with surface-mounted temperature sensors. The batteries were individually depleted at 0.02C, 0.03C, and 0.05C to examine temperature changes on the surface of the batteries. The recorded temperature variations on battery surfaces are presented in Figure 3.9. The batteries were fully discharged, and the surface temperature differences for each load profile were reported. It was observed that all the batteries had a small temperature rise as compared to the ambient temperature. Throughout all the trials, lithium-ion batteries showed a slight temperature increase of (0.003 - 0.004 °C).

Li-Po and Ni-MH batteries had 0.003°C and 0.008°C monotonic temperature differences, respectively. In the same vein, alkaline batteries showed temperature differences of 0.004 to 0.009°C. The attached load has no effect on the internal temperature of the battery. Thus, lower currents, corresponding to WSN application load profiles, have little effect on internal battery temperatures. Battery energy is now solely dependent on load and SOC, simplifying calculations. In tests, these attached loads had no effect on high precision low power SOC applications. Due to WSN deployments, the ambient temperature can range from -20°C to 60°C. So only the effect

of ambient temperature on battery power and OCV must be studied for reliable SOC measurement.

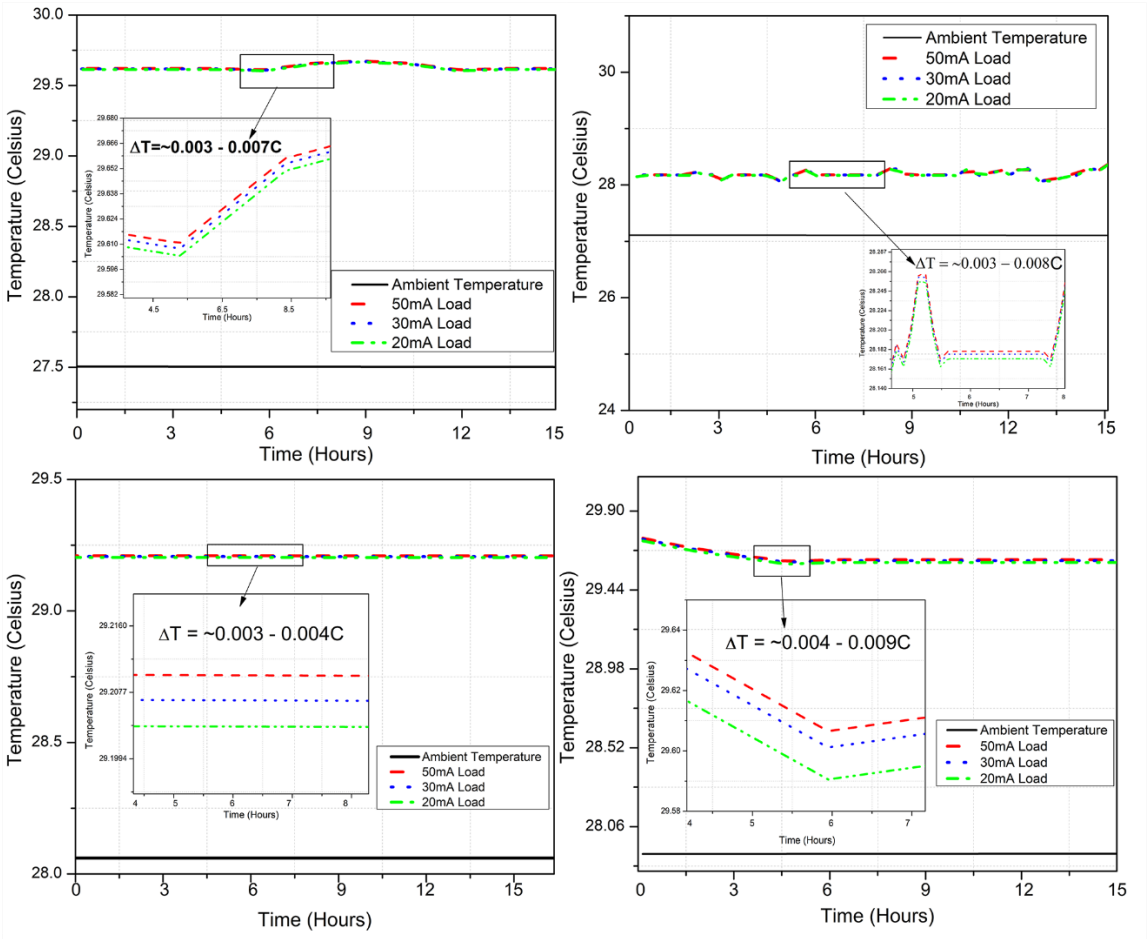



Figure 3.9: Effect of load current on battery surface temperatures. Nickel Metal Hydride (top left), Lithium Polymer (top right), lithium-Ion (bottom left), and Alkaline (bottom right).

## 3.4 Summary

In this chapter a detailed battery characterization and parameter extraction method for small load applications was presented. In this regard, four battery chemistries were investigated to determine the best battery for WSN applications. The pre-conditioning tests were carried out to remove battery passivation and stabilize battery capacity, recovering nearly 4% of the battery capacity. Furthermore, the relaxation tests enabled the calculation of the optimum time for battery energy recovery for all the battery types used in this study. It was observed that an optimum period

ranging between 8 and 24 hours is required for the batteries to fully regain their capacity, thus a critical factor for WSN application. Furthermore, the capacity approximation errors were studied using OCV and Ah method, where OCV was found more accurate but time and resource consuming. Finally, the effect of load current on battery temperature was observed, where the typical smaller WSN currents exhibited no significant effect on battery surface temperatures. The results concluded that lithium-ion batteries provided higher voltages and larger flat regions across all discharge rates. Furthermore, the thermal effect of higher discharge rates on Lithium-based batteries was found to be negligible, making these batteries the best choice for WSN applications.

# CHAPTER 4

# STATE OF CHARGE (SOC) ESTIMATION

In this chapter the battery discharge dataset is thoroughly examined to build, train, and test Machine Learning (ML) models for SOC estimation. In this regard, this chapter investigates various data normalization and transformation techniques that impacts the model accuracy. Various ML models were built and analysed for comprehensive performance evaluation based on ($R^2$, MAE, and RMSE) metrics. Finally, a Gaussian Process Regression based model with hyper-tuned radial bias function (RBF) was developed that is adaptive, scalable and can accurately estimate SOC for all battery chemistries used in this research.

## 4.1 SoC Estimation using Regression based techniques

In recent years, the development of data-driven algorithms such as machine learning methods has made significant progress in improving the accuracy of SOC measurements with improved classification performance, learning capabilities for high accuracy, and convergence. The performance of the battery is estimated based on its SOH and remaining useful life (RUL) (M.-F. Ng, J. Zhao, Q. Yan, G. J. Conduit, & Z. W. J. N. M. I. Seh, 2020). Artificial intelligence, such as machine learning and the deep learning approach, which are intelligent and adaptive, can predict the SOH and RUL of the battery. However, prediction and forecasting objectives are dependent on the availability of trained data. The artificial neural network (ANN) approach has a high potential for creating a non-linear layout among input and output parameters, demonstrating the non-linear model complexity. However, the precision of the data-driven approach is dependent on the quantity and quality of the data, which can lead to

over-fitting and under-fitting issues. In this regard, various ML techniques were employed in this research while keeping the following broader objectives in mind:

- The algorithm must be light, computationally efficient, and capable of running on embedded devices.
- The algorithm must be scalable and adaptive, and it must be capable of estimating SoC for various battery chemistries without the need for re-training or re-modeling.

Therefore, four ML techniques to train and test models based on the recorded dataset were compared for performance validation as illustrated in Figure 4.1.

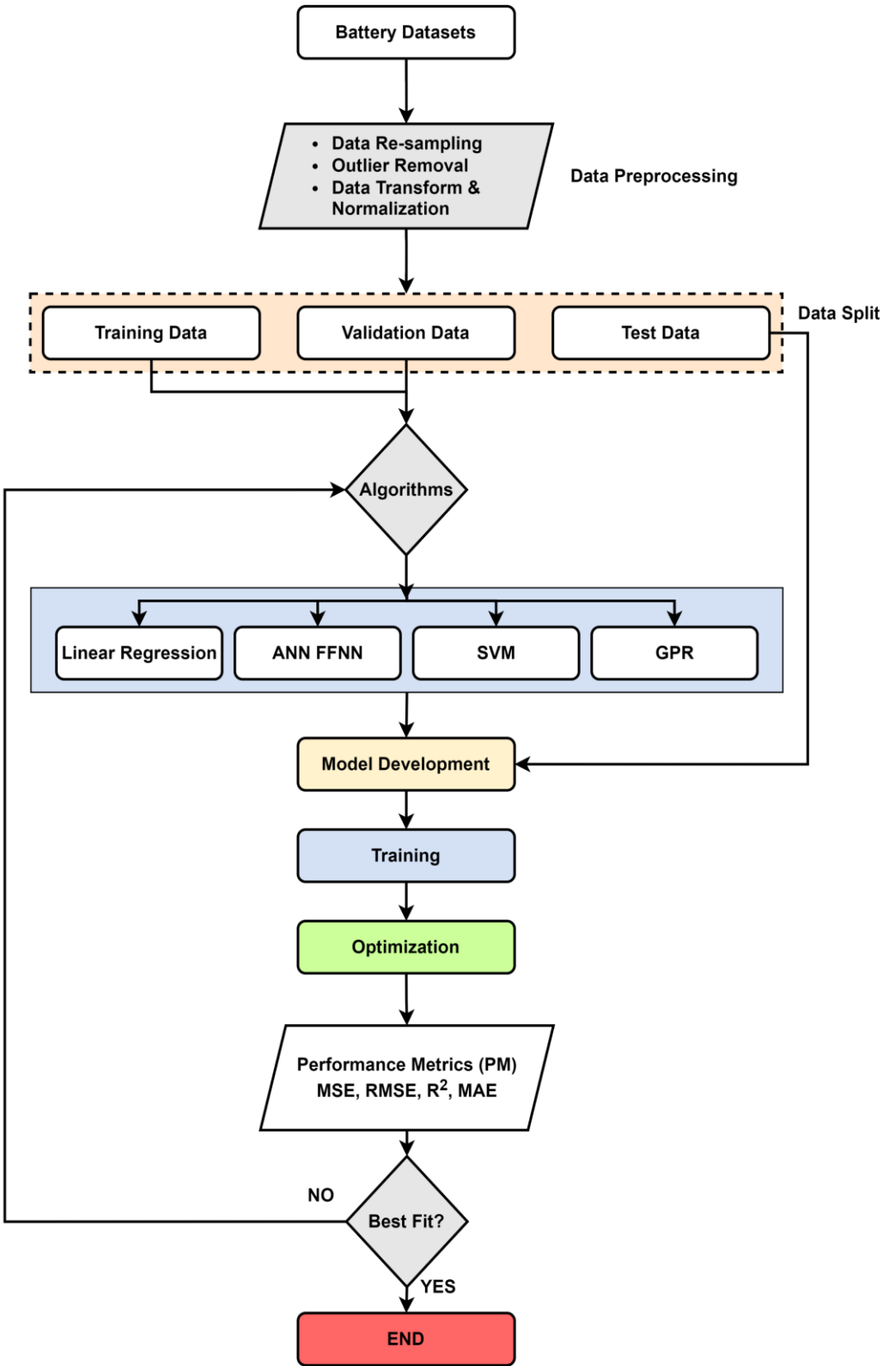



Figure 4.1: Experimental Flow diagram for ML modelling techniques

### 4.1.1 Data Filtering and Normalization

The performance of ML models greatly depends on the feature selection, the correlation between these features, feature distribution, and corresponding variances.

Therefore, a correlation matrix was computed to observe the relation between the features. The battery discharge profiles were observed at 5 different temperature ranges (including 5°C, 15°C, 25°C, 35°C, and 45°C). In this regard, battery capacity, voltage, and temperature range were selected as predictor variables, whereas SoC was dependent or target variable. It is important to understand the feature set distribution, which in some cases could affect up to 15% model accuracy (R.-C. Chen, Dewi, Huang, & Caraka, 2020; Karabulut, Özel, & İbrikçi, 2012; McCrae, 2018). Typically, scaling, normalization, and logarithmic transformation were performed to achieve a normal distribution on the feature set. In addition, feature representation in a similar scale not only helps to speed up ML model training but also improves the accuracy. Temperature and battery capacity exhibited a normal distribution that is closely related to the Gaussian distribution. Whereas battery voltages had many spikes (due to non-linear discharge and noise), as well as outliers resulting in a skewed distribution. Therefore, two separate transformations were performed on the feature set. In this regard, a standard scaling technique was implemented on battery capacity and temperature ranges values. The normalization or standard scaling transforms the features independently to achieve a (μ=0 and σ=1), this could be explained from Equation (8).

$$Z = \frac{x - \mu}{\sigma} \tag{8}$$

Where, $\mu = \frac{1}{N}\Sigma_{i=1}^{N}(x)_i$ is the mean, and $\sigma = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(x_i - \mu^2)}$ is the standard deviation. This results in a normalized feature (Z) for every input number of samples (x). Standard scaling is preferred with feature sets having normal distribution and very few outliers. Whereas the recorded voltage values had a skewed distribution therefore standard scaling technique was avoided. Consequently, the power transformation technique was implemented to reduce the skewness to normality. Typically, power

transformation is applied by directly calculating the log or square root of the predictor variable, although this is a straightforward implementation but may not account for every variable. Therefore, Yeo-Johnson power transform (Yeo & Johnson, 2000) was implemented that simultaneously performs convex-concave transforms on the feature set. This transformation also helps to normalize negative values; however, negative voltage values were not recorded, and any such value was considered as an anomaly and subsequently dropped. The power transformation normalizes a set of (x) input samples while estimating the scaling function (λ), as given by Equation (9).

$$\psi(\lambda, x) = \begin{cases} ((x+1)^{\lambda} - 1)D/\lambda & (x \geq 0, \lambda \neq 0), \\ \log(x+1) & (x \geq 0, \lambda = 0), \\ -((-x+1)^{2-\lambda} - 1)/(2-\lambda) & (x < 0, \lambda \neq 2), \\ -\log(-x+1) & (x < 0, \lambda = 2) \end{cases} \tag{9}$$

Where $\psi(\lambda, x)$ solves for concave value for ($\lambda < 1$ ) and convex values for ($\lambda > 1$ ), respectively. The power transformation is extremely useful for regression-based ML models, where the process kernel assumes a Gaussian distribution function for input variables.

### 4.1.2 Data Partitioning

Data partitioning is important in testing and validation of a trained model with a subset of data out of the analysis. It is typically used in supervised ML techniques where prediction accuracy on the training data set is needed to be analyzed based on the test and validation datasets. Another important aspect of data partitioning is to equally distribute the feature set within the desired threshold. This ensures that the model eliminates over and under-fitting which adds bias and variance to the estimation, thus affecting overall accuracy. However, this must not be confused with stratified models where predictors vary across different subgroups of partitioned datasets. In this regard,

the training and test dataset followed an 80-20% split ratio. To avoid over-fitting, the k-fold (5-folds) cross-validation technique was used to further split the training dataset into a validation subset. The iterative process of k-fold cross-validation ensured minimum data leakage and optimum subset threshold to hyper-tune model parameters. In addition, all ML models were further tuned to observe the impact of the train-test split ratio on model performance. Finally, the trained ML models were evaluated using the test dataset to record performance metrics, as per the drive cycles mentioned in Table 3.4.

### 4.1.3 Performance Metrics (PM)

Model performance evaluation varies from types of functions and intended applications. For instance, classification-based models are generally evaluated using Classification accuracy, Logarithmic Loss, Area under the curves (AUC), and F-measures. Whereas regression-based models that are generally designed for prediction or estimation are evaluated using $R^2$, MAE, and RMSE values. These performance metrics individually provide statistical observations of its accuracy and proportions of variance between predictors and target variables. Typically, the combined observation of these metrics provides a comprehensive model performance estimation. These performance metrics are detailed in *Appendix B*.

### 4.1.4 Linear Regression (LR)

Regression analysis fits a data model as a mathematical function. These functions usually relate two variables in a curved way. The mathematical function that maps predicted outputs to inputs computes prediction sum of squares. These estimations compare a set of Y observations to the original function values used to predict. It is found by subtracting the fitted nonlinear function from each Y point in the data set. The variances are then squared. The squared numbers are then added. The function fits the

data points better if the sum of these squared numbers is low. Nonlinear regression uses logarithmic, trigonometric, exponential, power, and Gaussian functions. Nonlinear models require more iterations (heuristics) than linear models because the operation is developed through trial and error.

Regression models that appear nonlinear at first glance are frequently linear. The curve estimation procedure can be used to determine the nature of the functional relationships present in the data, allowing you to select the appropriate estimation technique, whether linear or nonlinear. Therefore, these models are initially mathematically investigated by using the regression Equation (10).

$$\hat{y} = \begin{cases} b_0 + b_1 X_i + b_2 X_i + b_3 X_i + b_4 X_i \\ b_{(1,2,3,4\ldots n)}\ are\ bias\ values \\ X_i\ are\ the\ input\ features \end{cases} \quad (10)$$

As mentioned in CHAPTER 3, the effect of IEEE 802.15.4 protocol-based discharge currents resulted in negligible change in temperature rise at the battery surface. In addition, the temperature was maintained at a fixed value for each discharge cycle, therefore, due to its fixed value it was not used as a feature or coefficient for model fitting. This resulted in the use of *SOC*, *capacity,* and *voltage* as the coefficients for model fitting, resulting in Equation (11).

$$\begin{aligned} soc \sim 1 + capacity^2 + capacity \times voltage + voltage^2 \\ + capacity^3 + (capacity^2) : voltage \\ + capacity : (voltage^2) \end{aligned} \quad (11)$$

MATLAB's non-linear regression fit model was used to estimate the polynomial co-efficient fitting. They are needed to provide a fitted line across the battery discharge curves (Voltage vs SOC). Constructing the polynomial order and thus the polynomial equation required coefficient estimation. This helps map the data and improve detection accuracy. The *fitlm* function in MATLAB was used to parameterize the dataset in this case. *Poly32*, which corresponds to a cubic order polynomial equation, performed the best and was thus applied to each battery dataset separately.

### 4.1.5 LR Model Results

Initially the LR model was applied to the datasets corresponding to all batteries at room temperature operating at (20mA) discharge current. As the battery discharge curves follow a similar curve shape for all battery types, a typical LR fitting equation predicts the responses. However, the variations in datasets (range of temperature and discharge current values) result in a larger variation in predicted values resulting in residuals. The residuals are the difference between predicted and actual values and may affect the model progress. In this regard, Table 4.1 describes distinct statistical outcomes of LR model on the batteries. There were approximately 6000 data points for each dataset that was fitted using the *fitlm* function from MATLAB statistics toolbox.

The calculated values for all eight coefficients for each battery type reflects values converging to zero, which reflects the model convergence. However, generally RMSE and $R^2$ metrics are preferred to estimate the overall accuracy of the model. RMSE is a good measure of accuracy, but it should only be used to compare prediction errors of different models or model combinations for a single variable, as it is scale dependent. For instance, an RMSE of 0.6 is considered small for a single variable datum with a range of 0 to 1000, but it is no longer a negligible value when the range is changed to 0 to 10. As per a rule of thumb, RMSE values between 0.2 and 0.5 indicate that the model can reasonably predict the data. $R^2$ metric provides goodness of fit for a prediction model. For linear regression models, R-squared is a goodness-of-fit metric. This measure shows the percentage of variation of the dependent variable that the independent variables account for when taken together. On a compact 0–100 percent scale, R-squared quantifies the relationship strength between the model and the predictor variables. Typically, $R^2$ values above 0.9 shows a strong goodness of fit.

Table 4.1: LR model statistical output of various battery chemistries

| Lithium-Ion | | | | |
|---|---|---|---|---|
| **Performance Metrics** | | **Values** | | |
| No. of estimated coefficients | | 8 | | |
| RMSE | | 0.508 | | |
| $R^2$ | | 1 | | |
| Parameter estimation using model fitting equation | | | | |
| **Estimated Coefficients** | **Estimate** | **SE value** | **tStat** | **p-value** |
| Capacity | 0.7066 | 0.0323 | 21.829 | 1.033e-9 |
| Voltage | 0.74 | 1.729 | -69.809 | 0.0113 |
| $Capacity^2$ | -0.0009092 | 1.705e-5 | -53.311 | 0.0147 |
| Capacity:Voltage | 0.02989 | 0.01868 | 1.6003 | 0.109 |
| $Voltage^2$ | 34.261 | 0.5013 | 68.345 | 0.0327 |
| $Capacity^3$ | -7.17e-8 | 1.714e-9 | -41.849 | 0.6158 |
| $Capacity^2$:Voltage | 0.000286 | 5.397e-6 | 53.111 | 0.01061 |
| Capacity:$Voltage^2$ | -0.06050 | 0.00292 | -20.673 | 9.04e-9 |
| Lithium-Polymer | | | | |
| **Performance Metrics** | | **Values** | | |
| No. of estimated coefficients | | 8 | | |
| RMSE | | 0.291 | | |
| $R^2$ | | 1 | | |
| Parameter estimation using model fitting equation | | | | |
| **Estimated Coefficients** | **Estimate** | **SE value** | **tStat** | **p-value** |
| Capacity | 0.10054 | 0.00190 | 528.58 | 0 |
| Voltage | 0.0424 | 0.0164 | -2.5347 | 0.0112 |
| $Capacity^2$ | -1.4661e-8 | 4.229e-8 | -0.34662 | 0.7288 |
| Capacity:Voltage | -0.002443 | 9.646e-5 | -2.5332 | 0.0113 |
| $Voltage^2$ | 0.00799 | 0.0027 | 3.0242 | 0.00250 |
| $Capacity^3$ | -4.817e-1 | 1.057e-1 | -4.5546 | 5.362e-6 |
| $Capacity^2$:Voltage | 3.082e-8 | 1.392e-8 | 2.2129 | 0.0269 |
| Capacity:$Voltage^2$ | 2.274e-5 | 1.254e-5 | 1.8137 | 0.0697 |
| Nickel-metal hydride | | | | |
| **Performance Metrics** | | **Values** | | |
| No. of estimated coefficients | | 8 | | |
| RMSE | | 0.293 | | |
| $R^2$ | | 1 | | |
| Parameter estimation using model fitting equation | | | | |
| **Estimated Coefficients** | **Estimate** | **SE value** | **tStat** | **p-value** |
| Capacity | 0.0831 | 8.441e-5 | 984.48 | 0 |
| Voltage | 0.002899 | 0.00748 | 0.3872 | 0.6985 |
| $Capacity^2$ | -3.740e-8 | 8.458e-8 | -0.4421 | 0.6583 |
| Capacity:Voltage | 0.000127 | 4.4801e-5 | 2.8431 | 0.00448 |
| $Voltage^2$ | -0.0009148 | 0.002274 | -0.4021 | 0.6875 |
| $Capacity^3$ | -6.814e-11 | 8.261e-1 | -8.248 | 1.986e-6 |
| $Capacity^2$:Voltage | 4.545e-8 | 2.823e-8 | 1.609 | 0.10749 |
| Capacity:$Voltage^2$ | -2.2023e-5 | 9.128e-6 | -2.412 | 0.01587 |

The RMSE values of all battery types using the similar regression equation falls under acceptable limits (0.2 – 0.5). Similarly, the $R^2$ values of 1 were reported for all the battery types, concluding that the predicted model can be applied to all battery types. However, a closer look at the estimated coefficient values suggests that the predicted model is sensitive to input parameters, particularly for *voltage* values, as given by estimates and standard error (SE) parameters. In general, the model provides a considerable fit which can be used for SOC prediction, as illustrated in Figure 4.2.

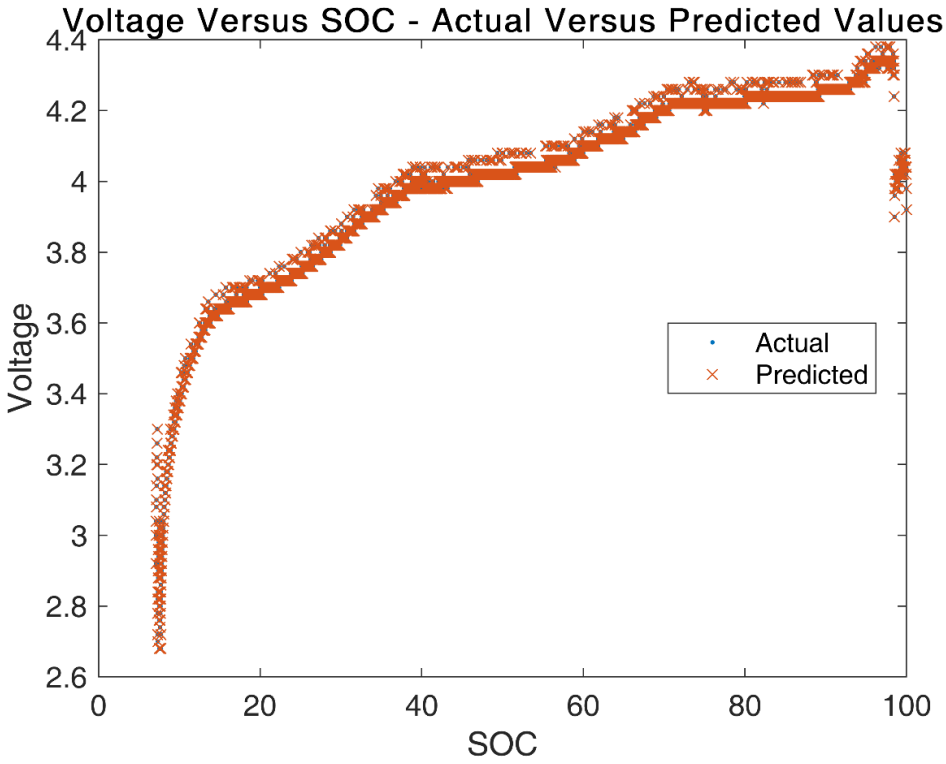



Figure 4.2: A comparison of actual versus predicted SOC values using LR model

The LR model initially fits almost perfectly, resulting in close SOC values approximation over time. In later problems, the cubic polynomial was used to approximate actual voltage discharge curves. However, looking at the residuals can help assess the fit's accuracy. A residual value indicates the degree of variance from the calculated mean value at a specific instance of a dataset. In other words, a model with lower residuals fits a data point better. A residual histogram is presented in Figure 4.3 that illustrates the degree of variance at specific data points.

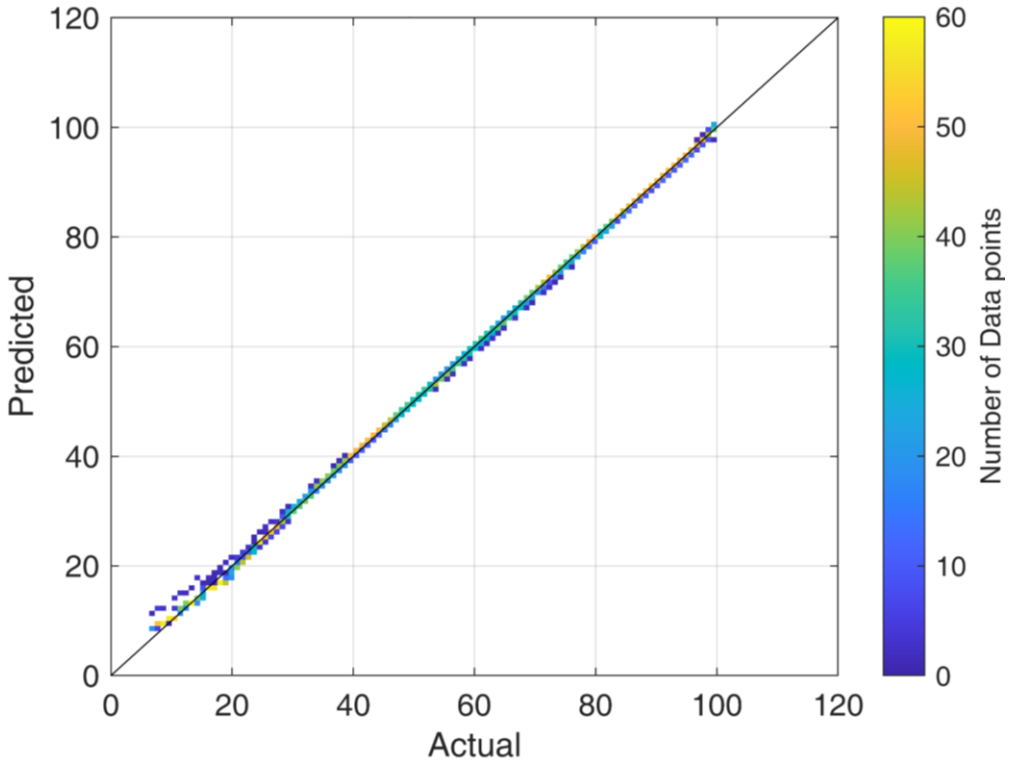


Figure 4.3: Residual histogram for LR model

As shown, the model accurately predicts data points across the region. During the linear discharge region of the curve, a few residuals are required. However, as the battery's SOC drops below 30%, the residual values spike. This trend continues around 20% and below SOC. As the battery nears its charge capacity, the voltage drops, causing frequent voltage fluctuations. As a result, the model's accuracy suffers. In this regard, the residual distribution must be observed to further investigation as given by Figure 4.4. The residuals have a Gaussian-based normal distribution with most values around the mean. In order to select the best prediction model, LR model provided an initial estimate of polynomial equation order. However, the model had several flaws that hampered its performance. As a result of this, simple polynomial regression models have been found to be highly sensitive to coefficient values. A change in dataset variables (for example, discharge rates, temperature) can affect model accuracy.

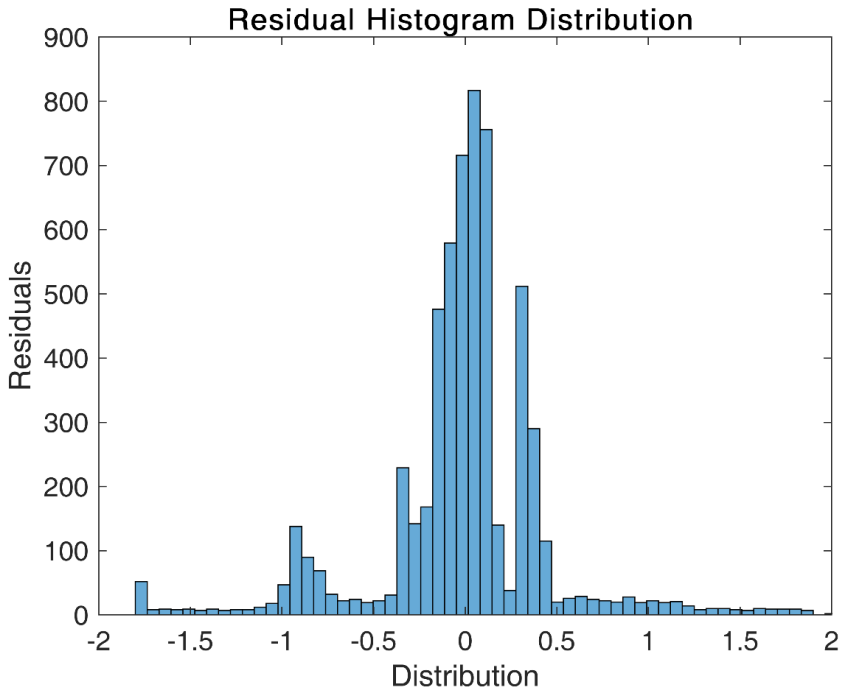


Figure 4.4: Residual histogram distribution for LR model

The model must be tuned for each dataset and thus cannot adapt to all applications. The model would also become more complex as the feature set and dataset grew. The model must also estimate the entire range for each new dataset, making it slow and non-scalable. Finally, due to limited on-board storage, these models require a complete working dataset for estimation.

## 4.2 Machine Learning (ML) Models

Machine learning is divided into two types: supervised and unsupervised machine learning. Large, labelled datasets with categorized input and output parameters are used to train the machine in supervised learning problems. These algorithms are typically used to solve classification and regression (also known as estimation or prediction) problems. Unsupervised learning focuses on patterns or trends found in the dataset used to train the machine. Clustering and association-based problems are the most common unsupervised learning problems. In this research supervised learning technique was used that utilized the recorded battery dataset as described in Chapter

CHAPTER 3. The fundamental goal is the SOC estimation based on the large, labelled dataset, therefore this technique naturally becomes a regression problem. It is important to know that within the context of ML and SL in particularly, there are various techniques that can be used for regression-based problems. These techniques are mostly selected based on the feature-set, volume, and nature of the dataset. For instance, for a smaller labelled dataset with high correlation among predictor variables, simple ML techniques such as Feed Forward Neural Networks (FFNN) can be deployed. For larger, complex, and sequential datasets Deep Learning (DL) techniques are preferred. Furthermore, for specialized datasets such as time-series based datasets, Recurrent Neural Networks (RNN) based techniques such as Long Short-Term Memory (LSTM) are preferred for better performance.

The majority of BMS research focuses on adaptive SOC, SOH, and DOD estimation techniques. To begin, most of the publications employ the supervised learning (SL) method to train the models, which is known to necessitate a large amount of data. Even in instances where sufficient data is available, the training time for DL models is typically many hours or days. Second, models trained on one type of cell chemistry do not pertain to other types of cell chemistry. Although initial studies suggest that transfer learning is possible, more research is needed to confirm its accuracy when applied to a larger number of cells with different chemistry (Bhattacharjee, Verma, Mishra, & Saha, 2021). In most cases, a model trained on data from one Li-ion battery cell does not adopt well to data from another cell, necessitating retraining from ground - up (X. Ren, Liu, Yu, & Dong, 2021). Therefore, it is necessary to describe the ML architecture and fundamental requirements to map the desired feature set. In this regard, FFNN architecture and design requirements are explained in *Appendix C*.

### 4.2.1 MATLAB FFNN Model

As discussed in the previous section and in *Appendix B*, the FFNN architecture in this research comprised of a single hidden layer. The input layer fed two neurons accounting for the vectors (voltage, and capacity), which are then mapped to the output vector (SOC). The (Levenberg-Marquardt backpropagation) training function was used for the FFNN. The network performance with initial configuration parameters is shows in Figure 4.5.

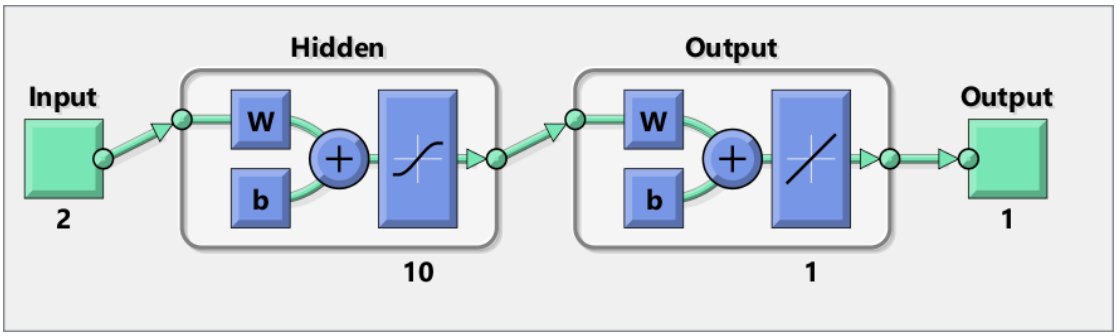



Figure 4.5: FFNN architecture in MATLB for SOC estimation.

As shown in the figure, the network has a single hidden layer that is initiated with 10 neurons. However, based on the rules discussed in the previous section, the model was tuned, and the optimal number of neurons were computed. This was required to estimate the network performance improvements and its effect on accurate SOC estimation subsequently.

### 4.2.2 ML Algorithm Results

The previously described model architecture was trained individually for each battery dataset, with the total number of datasets determined by drive cycles, as previously explained in Table 3.4. In this regard, model performance metrics including convergence time, RMSE, and $R^2$ values were observed to classify model accuracy and predictability. The models were trained and validated using several optimization strategies that reflected on its architecture.

These optimization steps included:

- *Network 1*: Design of a single layer network with 1 neuron each.
- *Network 2*: Design of a single layer network with 10 neurons each.
- *Network 3*: Design of a single layer network with optimized number of neurons (incremental linear optimization based on RMSE score).
- *Network 4*: Design of a multiple-layer network with optimized number of neurons (incremental linear optimization based on RMSE score).
- *Network 5*: Optimized neurons per layer that exhibits the best RMSE score.

The model optimization is required to identify the most efficient FFNN architecture for the observed dataset. Table 4.2 explains the model performance parameters in detail.

Table 4.2: Model performance analysis for various battery datasets

| **Battery Type: Li-Ion** | | | | | |
|---|---|---|---|---|---|
| **Parameters** | **Neurons/layers** | **Training Time (sec)** | **Epochs** | **$R^2$-value** | **RMSE** |
| **Network 1** | 1/1 | 6.987 | 27 | 0.973 | 1.684 |
| **Network 2** | 10/1 | 34.475 | 1000 | 0.99 | 0.0688 |
| **Network 3** | 50/1 | 1450 | 1000 | 0.980 | 0.0689 |
| **Network 4** | 44/2 | 357.556 | 1000 | 0.990 | 0.0096 |
| **Optimized** | 44/1 | 61.974 | 1000 | 0.998 | 0.0091 |
| **Battery Type: Li-Po** | | | | | |
| **Parameters** | **Neurons/layers** | **Training Time (sec)** | **Epochs** | **$R^2$-value** | **RMSE** |
| **Network 1** | 1/1 | 10 | 1000 | 0.947 | 0.987 |
| **Network 2** | 10/1 | 19.27 | 1000 | 0.948 | 0.33 |
| **Network 3** | 50/1 | 378 | 51 | 0.9484 | 0.032 |
| **Network 4** | 9/1 | 18.993 | 1000 | 0.9497 | 0.031 |
| **Optimized** | 9/2 | 13.45 | 410 | 0.99 | 0.029 |
| **Battery Type: Ni-MH** | | | | | |
| **Parameters** | **Neurons/layers** | **Training Time (sec)** | **Epochs** | **$R^2$-value** | **RMSE** |
| **Network 1** | 1/1 | 7.01 | 1000 | 0.986 | 0.3027 |
| **Network 2** | 10/1 | 12.117 | 995 | 0.9866 | 0.0032 |
| **Network 3** | 50/1 | 22.773 | 414 | 0.9866 | 0.0034 |
| **Network 4** | 100/1 | 158.24 | 680 | 0.9866 | 0.0187 |
| **Optimized** | 10/2 | 12.873 | 64 | 0.9877 | 0.0031 |

As previously stated, the objective was to develop a lightweight, computationally efficient FFNN network capable of accurately estimating SOC for a variety of battery types and operating conditions (temperature, and discharge currents).

The number of networks discussed in this section was necessary to gain an understanding of the model's performance and estimation accuracy to generate an optimised network architecture. As shown in Table 4.2, the architecture of the model (number of hidden layers and neurons per layer) has a significant impact on model performance. It is widely accepted that a single pass with the entire dataset should be used to accelerate training progress. However, a faster training process does not imply a faster convergence of the network, which is determined by the model accuracy score in each iteration. As model learning is an iterative process, it is common to tune the model iteratively. This typically results in a slower training process but a more rapidly converging network. Therefore, it is critical to distinguish between iterations, epochs, and batch sizes in this context.

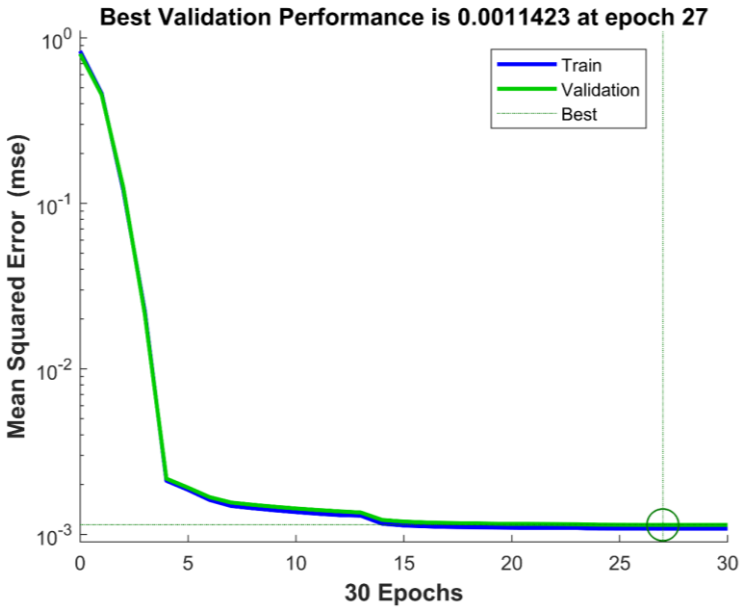
- An epoch is a descriptor being used machine learning that refers to the number of passes the machine learning algorithm must have made through the entire training dataset.
- Iteration refers to the number of instances a batch of data is processed by an algorithm. That is, in the specific instance of neural networks, it involves both the forward and backward passes. Thus, each time a batch of data is passed through the NN, an iteration is completed.
- The batch size is given by the total number of required samples in a batch.

By and large, models improve with increased training epochs, but only to a point. As they converge, their accuracy will begin to diminish. Therefore, in these experiments, a fixed data-split ratio of (70% training and 30% validated) datasets were used. The network tuning then iteratively selected number of epochs for each iteration to observe network convergence times.

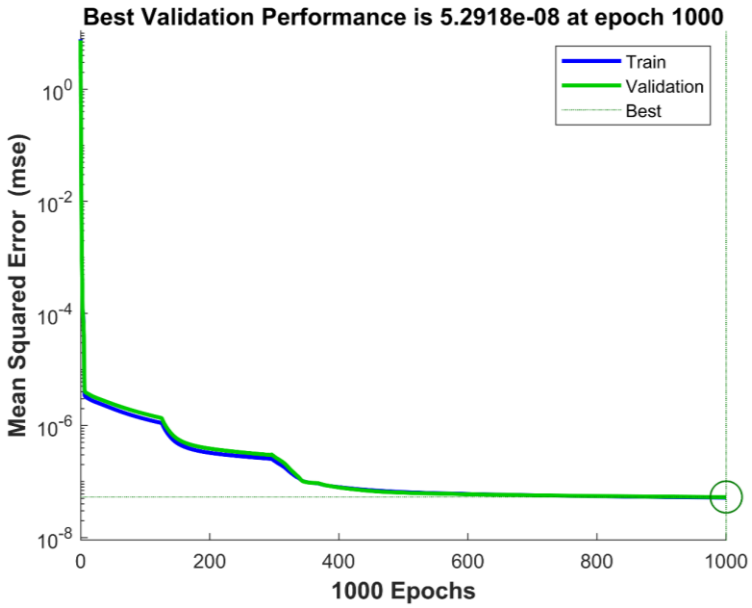
### 4.2.3 FFNN Model performance for Li-Ion battery datasets

The architecture of the model has a direct effect on its convergence. As shown in Table 4.2, the epochs used for the starter and optimized models differ, affecting the

convergence time and accuracy as well as illustrated in Figure 4.6, where the model training is evaluated based on number of epochs and MSE score for each iteration.




(a). Number of epochs for Network 1



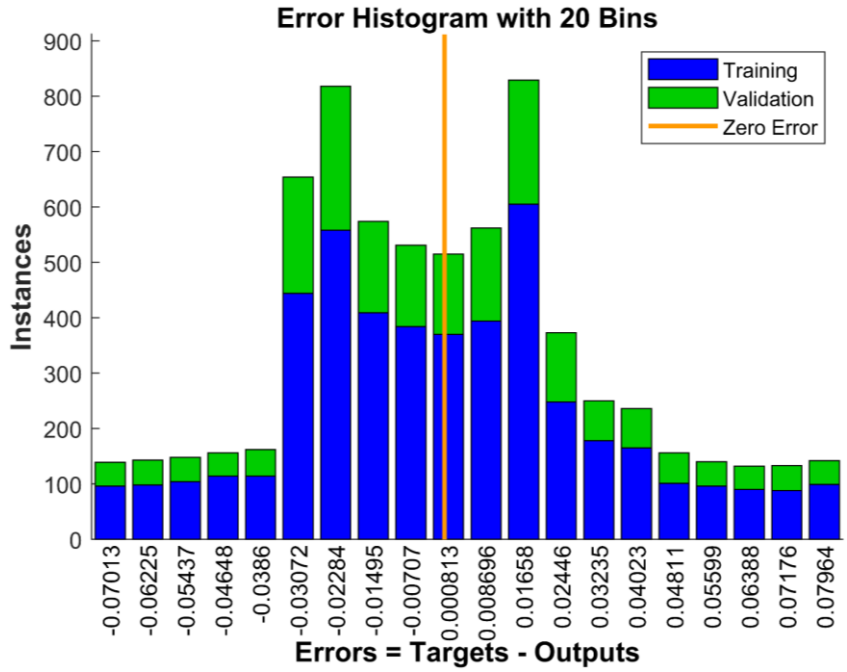

(b). Number of epochs for optimized Network.

Figure 4.6: Number of epochs and associated network accuracy for Li-Ion batteries.

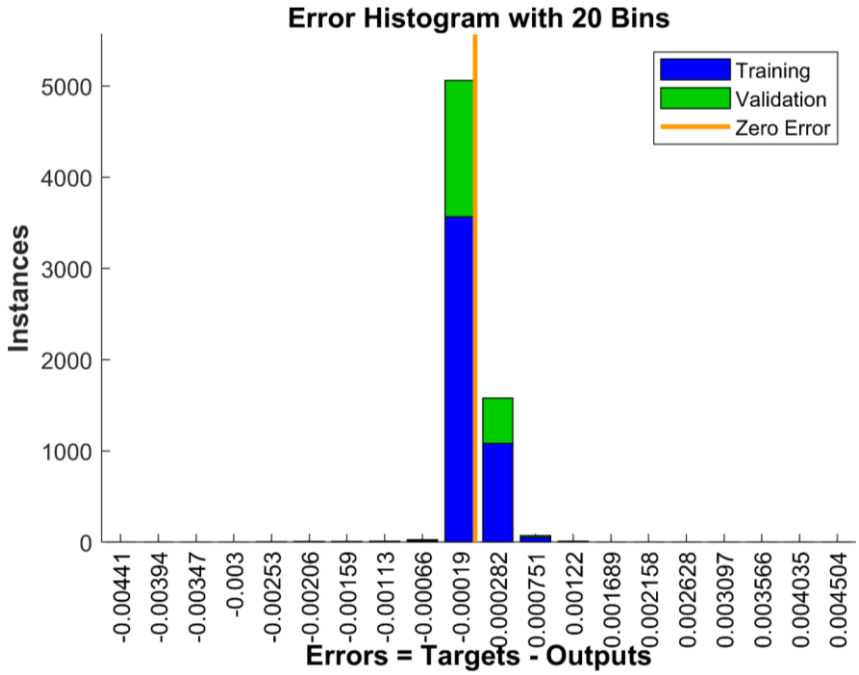
For a simpler network architecture with 1 neuron per layer, the network training progress was faster, as less epochs were passed to the algorithm for learning. This achieved a training time of 6.987 seconds as compared to the optimized network where it took 61.974 seconds, for training. The increase in network training time can be

associated with the increase in number of neurons that require additional learning in each iteration, therefore, requiring additional epochs. Another significant consideration is the reduction in MSE that clearly indicated the ability to optimize the network. However, further optimizing the network by adding additional hidden layers (as presented in Network 4) resulted in increased training time, but adversely affecting its performance. As illustrated in Figure 4.7, this can also be seen in the error histograms for both the training and validation stages of the models.

The ampere-hour method was used in this study to determine the battery's state of charge during the discharge cycle. However, because battery current and environmental conditions (such as temperature) have a greater effect on battery voltage, the OCV method was used to develop battery discharge curves (Murnane & Ghazel, 2017). Therefore, a precise mapping of voltage values is required to accurately estimate the soc values. The models were iteratively tuned in this regard to observe the effect of voltage variation errors on validation data. As illustrated in Figure 4.8, the optimized model's voltage estimation errors are significantly reduced, resulting in accurate mapping and thus accurate soc estimation.




(a). Error histogram illustrating residuals without optimization (Network 1)




(b). Error histogram illustrating residuals of optimized network

Figure 4.7:Error histogram of models prior, and post optimization

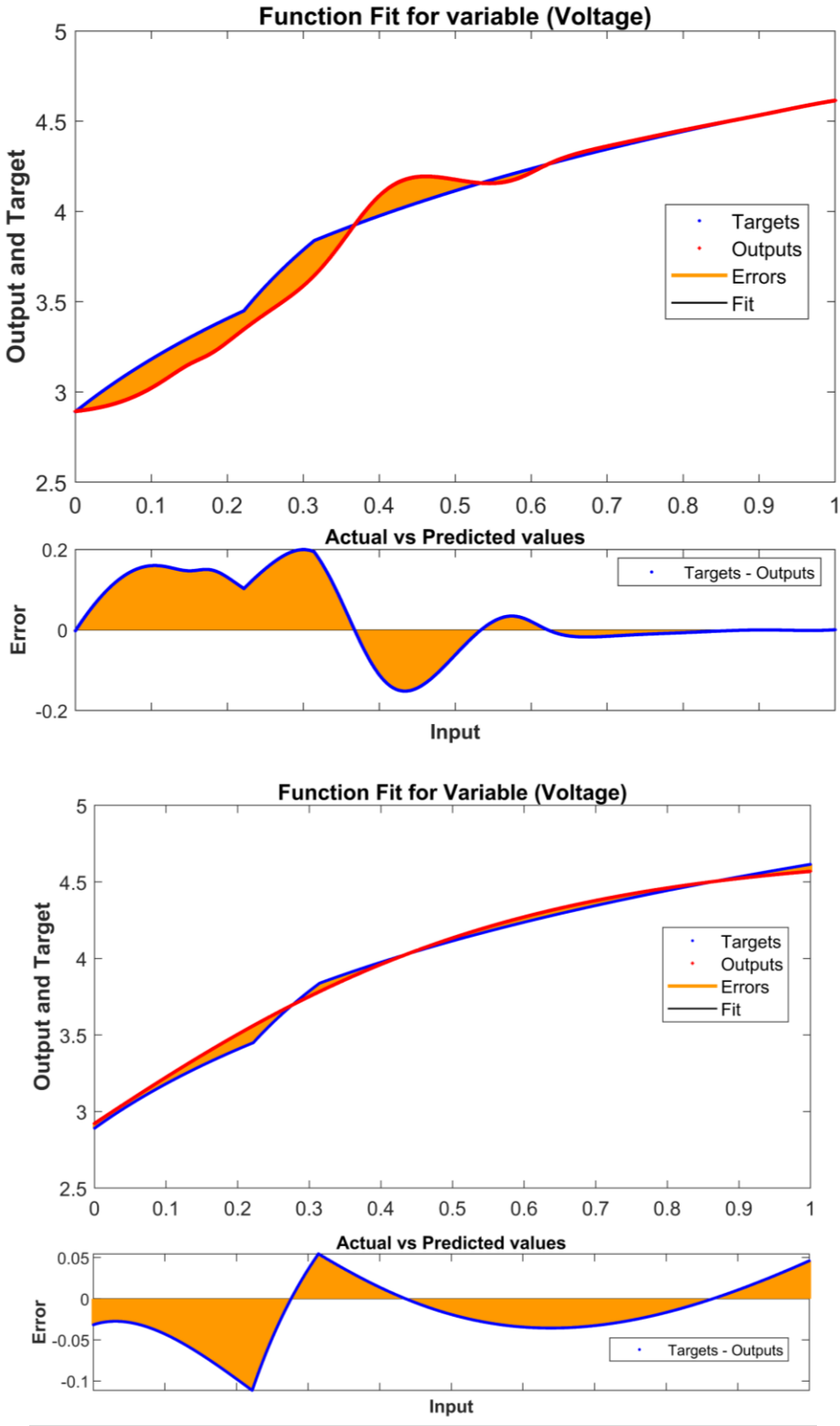



Figure 4.8: The function fitting mapping for predictor variable (voltage) for FFNN – Li-Ion batteries. Default configuration (top), optimized network (bottom).

The mapping between the predictor variable (voltage) and the target values is very close, resulting in very few and small errors. This is a critical step because it allows us to determine the degree of mapping accuracy for the critical predictor variables. Most neural network architectures are based on experimentation, which necessitates training hundreds of models in order to achieve the desired results. The most critical and decisive factor is the dataset dependency, which may require tuning a model for each independent application. Typically, when designing an optimal NN architecture, all moving parts, including the number of layers, the number of neurons per layer, and the activation functions must be investigated. Therefore, manually determining the optimal architecture is time-consuming and tedious work that can take hundreds of hours in some cases. Therefore, a computationally efficient algorithm was designed to allocate hidden layers and neurons per hidden layer that operates on a linear incremental step size. For each batch in which the model converges, the model performance metrics (represented by $R^2$ and RMSE scores) are calculated. The model with the lowest RMSE is then selected as the optimal model. Figure 4.9 illustrates the iterative process for Li-Ion battery datasets where optimum number of neurons are calculated.

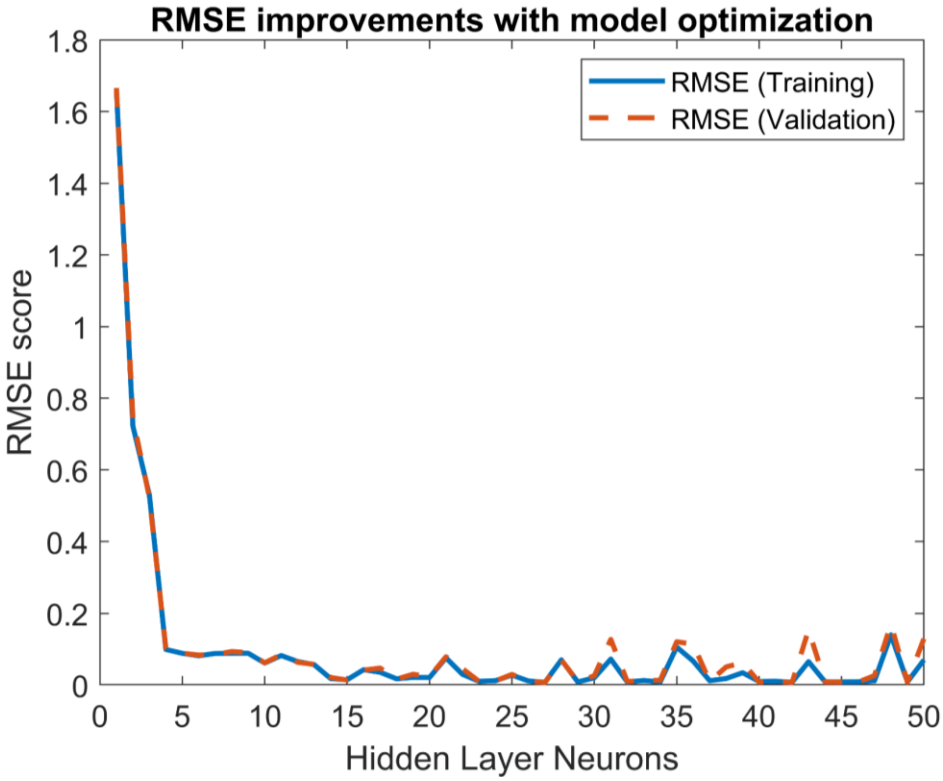


Figure 4.9: RMSE improvements by observing iterative increase in neurons in the hidden layers

As evident, the model achieves the lowest RMSE score with 44 neurons for both training, and validation data. Once the number of neurons is established, the algorithm advances to calculate the RMSE score by adding another hidden layer. Contrary to popular belief, increasing the number of neurons and layers does not always result in improved performance. This is demonstrated by the results in Table 4.2, where increasing the number of neurons (in Network 3) resulted in a decrease in the RMSE score. Additionally, the addition of another hidden layer degraded the model's performance. The model performed optimally with a single hidden layer architecture consisting of 44 neurons per layer. Figure 4.10 illustrates the optimized model's performance and estimated values.

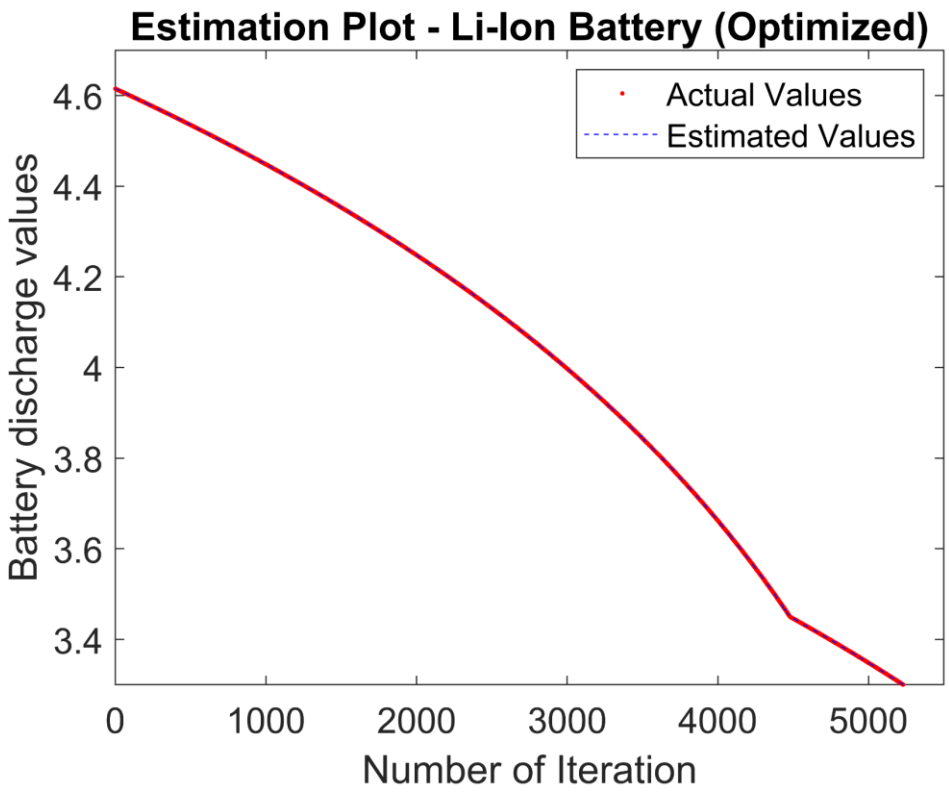


Figure 4.10: Model predicted voltage values mapping for Li-Ion battery datasets

### 4.2.4 FFNN Model performance for Li-Po battery datasets

Similarly, the FFNN model was investigated for Li-Po battery datasets keeping similar conditions. The model was iteratively tuned, and it was observed that increasing the number of layers increased the model accuracy in this case. Unlike Li-Ion battery dataset where a single layered architecture with 44 neurons outperformed other configurations; the dual layered model with nine neurons performed the best. (Tran, Lee, & Kim, 2020) reported a similar effect where network architectures are strongly dependent on the dataset and its features. It was reported that the network architecture varies by controlling its capacity based on layers and associated neurons per hidden layers. In this regard, the optimized model for Li-Po batteries converged around 13.45 seconds, with a few residuals for training and validation data. The error histogram of the model is presented in Figure 4.11.

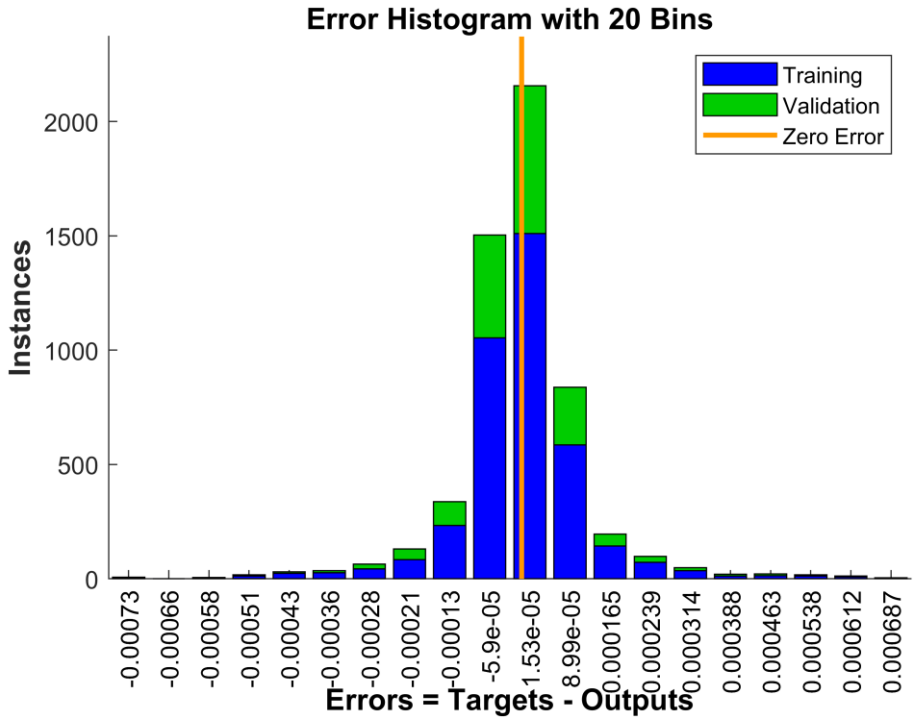



Figure 4.11: Error histogram for optimized 2-layer model for Li-Po batteries

As illustrated in Figure 4.10, the errors between the targets and output values for both training and validation data are marginal. Furthermore, the residuals follow a Gaussian distribution thus validating the data filtering and processing steps. The predictor variable (Voltage) mapping using 2-layered model was also found to be accurate as presented in Figure 4.12. For almost the entire discharge curve, the predicted voltage values are closely related to the actual values. However, a slight deviation in the flat region of the battery discharge curve was observed, which is most likely due to the flatter Li-Po profile. Even though the errors reported in this region are minor (0.1 – 0.18), this necessitated further investigation and network optimization. The network was fine-tuned further with up to four hidden layers and ten neurons per layer, yielding a slight improvement. This, however, increased network complexity, resulting in a transition from shallow to deep neural network (DNN) architecture, which may not be suitable for low-resource embedded devices. DNN has the potential to increase network capacity by utilising a fully converged network that provides improved accuracies.

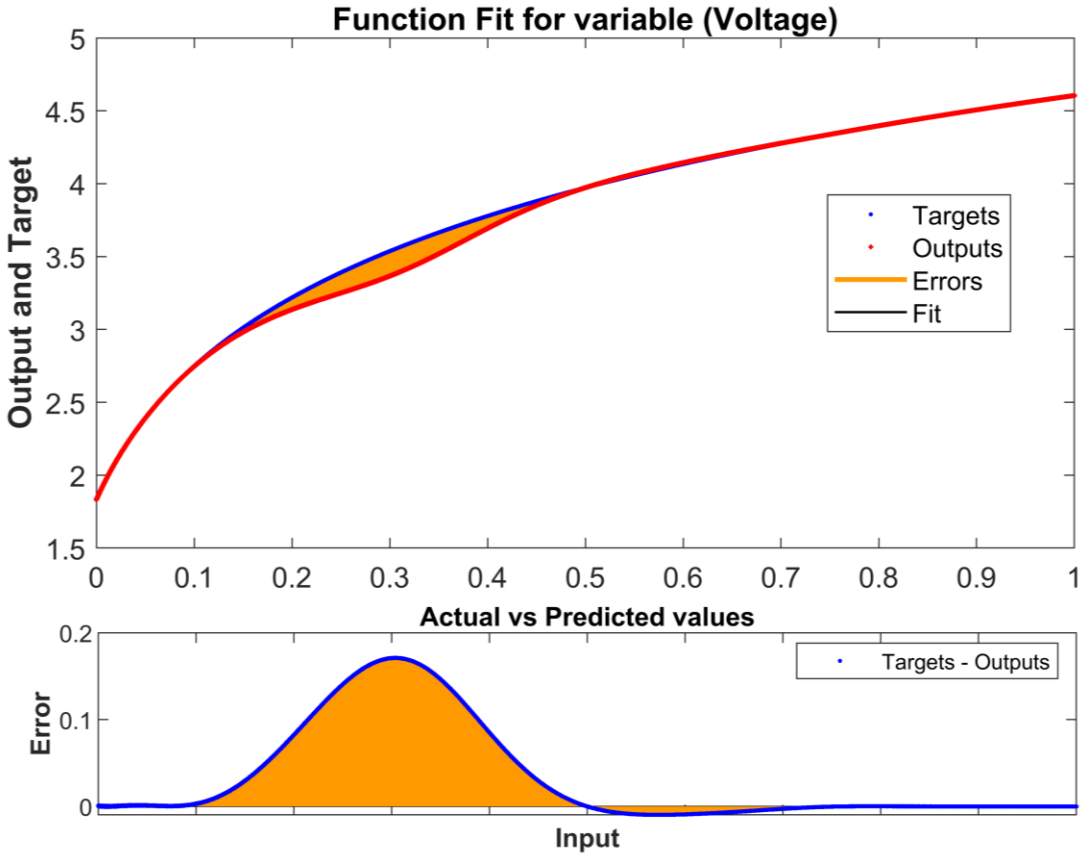



Figure 4.12: The function fit plot for predictor variable (Voltage) for 2-layer FFNN – LiPo batteries.

However, this results in increased complexity and training times, as well as longer on-device estimation times and the need for more memory. Therefore, the proposed architecture with two hidden layers was investigated further. The estimated values as presented in Figure 4.13 illustrate the accuracy in estimation predictor variables, thus a consistent and considerable model performance.

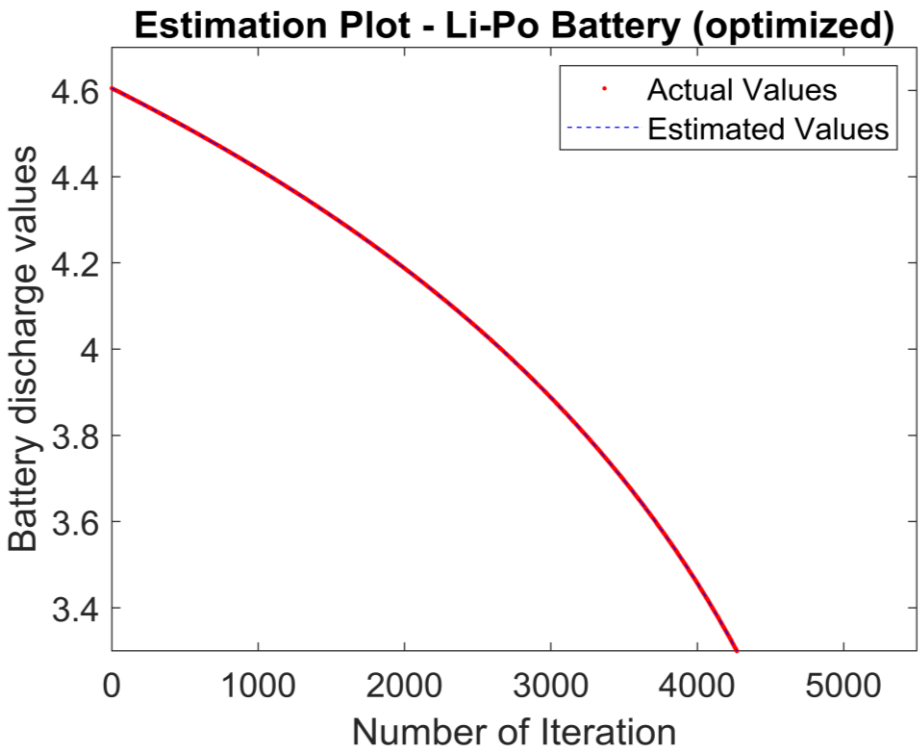


Figure 4.13: Model predicted voltage values mapping for Li-Po battery datasets

### 4.2.5 FFNN Model performance for Ni-MH battery datasets

The Ni-MH batteries are known to have a very distinct and flat discharge curves, resulting in discharge zones that have a very consistent voltage value. The discharge curves for Ni-MH batteries as reported in Section 3.3.4 (Figure 3.7), illustrate a flat voltage profile. This results in good electrochemical properties, but difficult for parametric estimation. In this regard, although it was evident that a depth was architecture model was preferable for these batteries, the design process still incorporated all the steps described in the previous subsection. The models were initial designed width-wise for simplicity, which were later optimized to achieve a model with 2 hidden layers having 10 neurons per layer. Figure 4.14 illustrates the residual error histogram for training and validation dataset for Ni-MH batteries. The errors are marginal, resulting in a very accurate estimation. Furthermore, the residuals follow a Gaussian distribution, which has no effect on model accuracy in the presence of noise or outliers. Although the battery discharge curves are non-linear in nature, the flat region

that accommodates 40-80 percent of the battery energy (and is the most difficult to estimate) appears to be flat or linear by definition. That implies that in the absence of noise or outliers, the residuals must adhere to the principles of normality. Therefore, the flat region is treated as linear, and the model inference (estimations or predictions) are also linear, making the difference (residuals) appear to have a normal distribution.

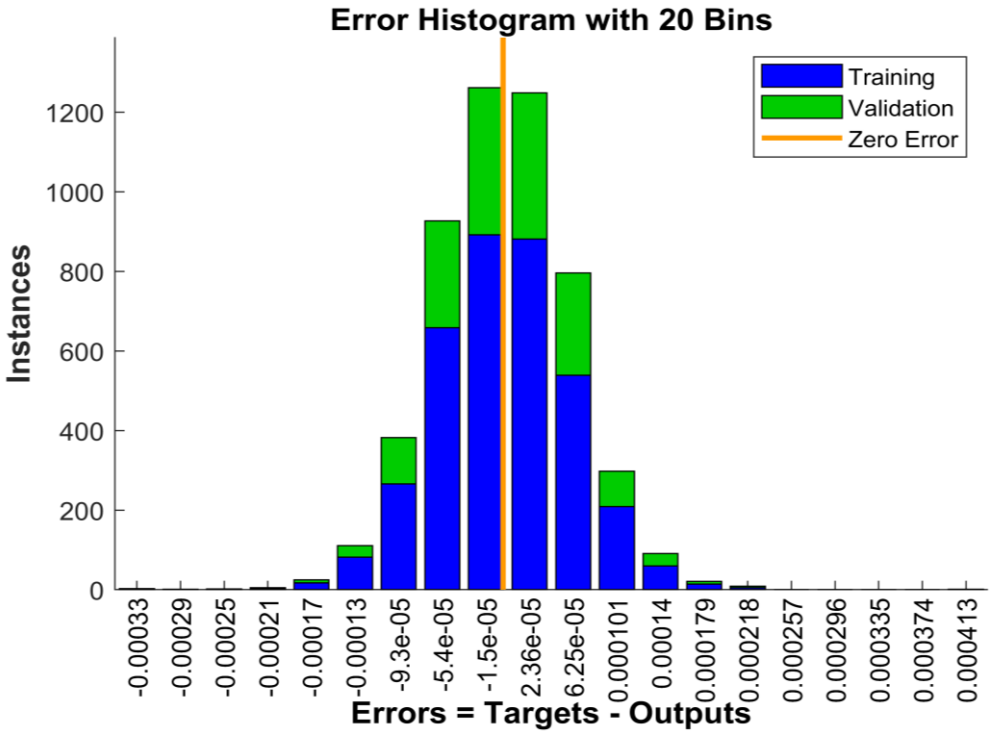



Figure 4.14: Error histogram for 2-layer FFNN model of Ni-MH batteries

The model fitness plots, as given in Figure 4.15 illustrate a close-fitting model that estimate accurately. The estimated voltage values are mapped closely to the target values, resulting in a few residuals in the flat discharge region, as discussed earlier. The plot reveals errors in the magnitude of $e^{-3}$ for this linear or flat region that are marginable and does not affect model accuracy. This also ensures that the predictor variables are accurately mapped thus resulting in precise SOC estimation. As previously reported in Table 4.2, the model training and validation trend shows fast convergence trend, yielding an average convergence time of 12.873 seconds.

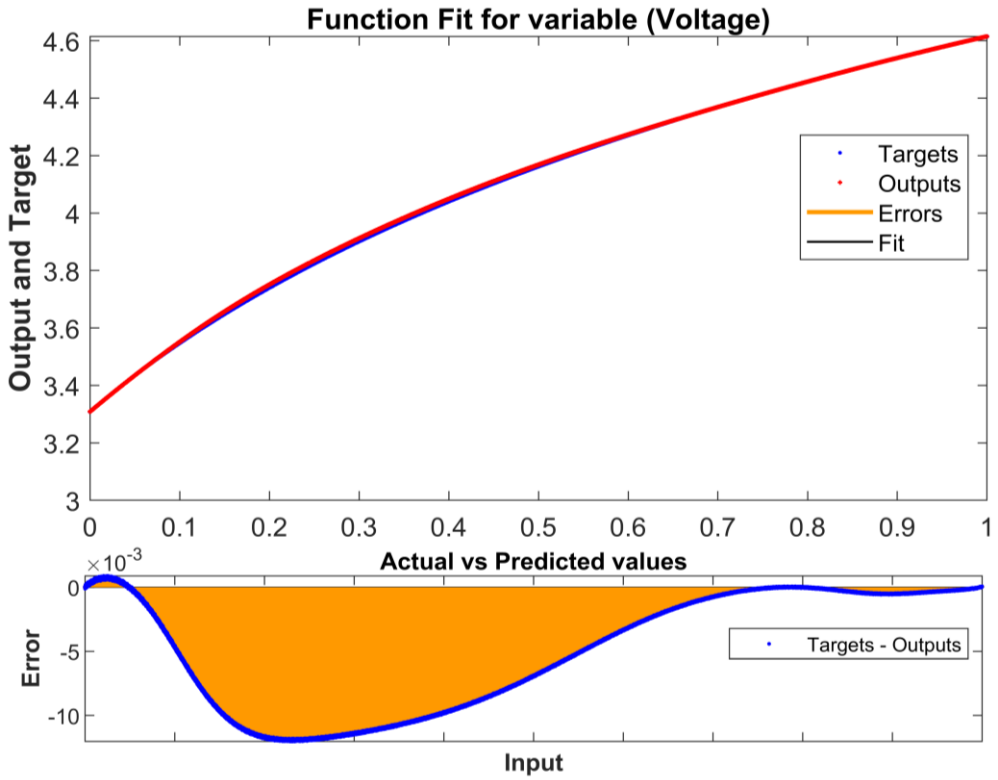



Figure 4.15: The function fit plot for predictor variable (Voltage) for 2-layer FFNN – Ni-MH batteries.

Finally, the predictor variable estimation accuracy can be verified from Figure 4.16.

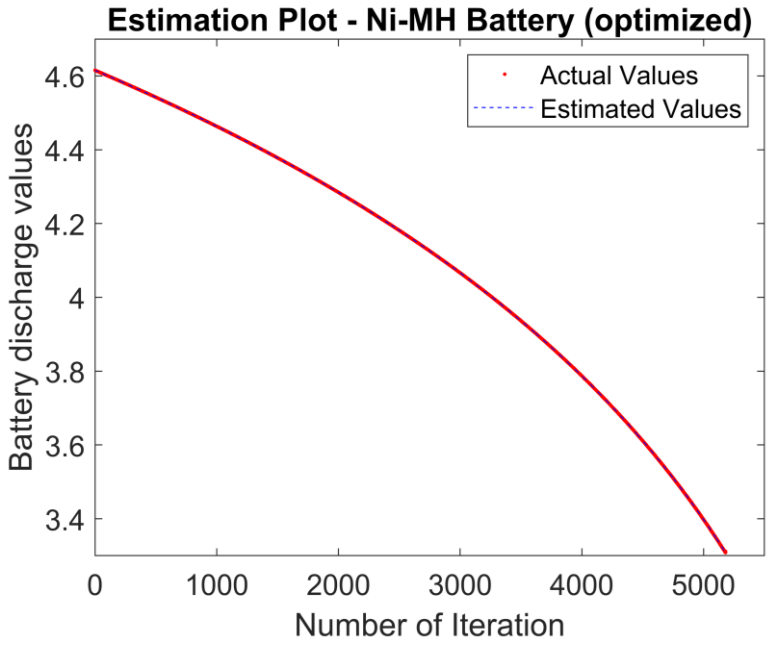



Figure 4.16: Model predicted voltage values mapping for Ni-MH battery datasets

### 4.2.6 Summary

It was observed that model architectures are highly dependent on the battery discharge dataset, such that a single feature can significantly affect the model architecture, necessitating a new design in some cases. Certain models performed optimally in terms of width, for example, the single-layer model designed for Li-Ion batteries. Other datasets necessitated a depth-wise multi-layer FFNN architecture. Additionally, the dataset can change Therefore of the effect of input parameters such as voltage, current, and temperature. Therefore, developing a single ANN model that can accommodate multiple battery chemistries, or in other words, datasets with varying operating conditions, is challenging. Although the error is marginable and does not directly affect the model's accuracy, the most significant challenge is the model's predictability uncertainty. That is, a model may work well for a particular data type, but its suitability for another source remains uncertain. Additionally, adding multiple trained networks to an embedded platform is not a viable option because it consumes additional storage, memory, and computational time.

In the case of Li-Ion, adding layers reduces model accuracy while increasing resource consumption and convergence time. For Li-Po batteries, it was observed that a two-layer FFNN outperformed a single layer architecture. Increasing the number of layers and neurons per layer, had no discernible effect on model accuracy. On the downside, it adds complexity to the model, necessitating the use of additional computational resources. NiMH behaved differently, with an increase in the number of neurons resulting in a decrease in RMSE and convergence time. However, as the number of layers increased, the model's accuracy increased. This means that when the input data set changes slightly, the model requires optimization that directly affects its architecture. It is highly improbable that a model can be tuned to perform well across

the entire dataset for all batteries. Due to the similarity of the battery discharge profiles, a close-enough model can be trained to accurately estimate SOC; however, model inaccuracies cannot be avoided. One approach to this problem is to investigate DNN, which is beyond the capabilities of the majority of embedded platforms. This inability prompted us to investigate alternative machine learning techniques that are more effective at adapting to new datasets.

## 4.3 Support Vector Machines (SVM)

The support vector machine (SVM) is a well-known and widely used computational intelligence technique in a variety of fields. The general concept underneath SVMs is using non-linear mapping for data analysis in certain areas and the linear algorithm in the others. An SVM is a binary linear classifier that is not probabilistic. Its main strength is its non-probabilistic nature. other words, an SVM divides data across a decision boundary (plane) established by a limited subset of the data (feature vectors). The data subset that represents the decision boundary is appropriately referred to as the support vectors. The remaining feature vectors in the dataset have no impact on the competitiveness of the decision boundary in the feature space. In contrast to SVMs, probabilistic classifiers present a system that best explains the data by considering all of the data rather than just a small subset. Therefore, probabilistic classifiers are likely to necessitate more computing resources. SVMs are amongst the most prominent and commonly used algorithms in machine learning for dealing with classification problems. This algorithm recognizes the presence of non-linearity in the data and generates an accurate prediction model. Support Vector Regression, or SVR, is another name for the SVM regression algorithm, which will be used interchangeably throughout this section.

The goal of an SVM is to identify a hyperplane in an n-dimensional feature space that classify data points clearly. Support Vectors are the data points on either side of the hyperplane that are closest to the hyperplane. These influence the hyperplane's position and orientation, and thus aid in the construction of the SVM. The SVR algorithm is a supervised learning algorithm for predicting discrete values. It operates on the same principle as SVMs, which is to find the best fit line. The best fit line in SVR is the hyperplane with the greatest number of points.

The SVR, unlike other regression models, tries to fit the best line within a threshold value, rather than minimizing the difference between the real and predicted value. The distance between the hyperplane and the boundary line is the threshold value. SVR's fit time complexity grows more than quadratically with the number of samples, making it difficult to scale to datasets with more than a few tens of thousands of samples. Linear SVR or Stochastic gradient descent (SGD) regressor are used for large datasets. Linear SVR is faster than SVR, but it only considers the linear kernel. Because the cost function disregards sample data whose prediction is close to their target, the model produced by SVR is only based on a subset of the training data. Therefore, in these experiments two SVR implementations with kernels (linear, polynomial, and Gaussian) were performed to observe the combined effect of model accuracy and convergence times.

Typically, a regression-based support vector solution can be described by Equation (12),

$$\Upsilon_i = \Sigma_i^N W.K(x_i, x_k) + B \tag{12}$$

Where,

$$\Upsilon_i = predicted\ output$$
$$W = weights$$
$$K = Kernel\ function\ parameters, given\ by\ \varepsilon\ and\ \mu$$
$$x_i, x_k\ are\ support\ vectors$$
$$B = Bias$$

The goal of most linear regression models is to reduce the sum of squared errors (SSE). In contrast to SSE, the goal of SVR is to minimize the coefficients, or more precisely, the coefficient vector, rather than the squared error. The value of epsilon can be adjusted to achieve the model's desired accuracy. This generates the new objective function by which the hyperplane and degree of margin are calculated for SVR, as given by Equations (13), and (14).

$$MIN\ \frac{1}{2}\ \|w^2\| \tag{13}$$

With the constraints,

$$\varepsilon \geq\ |\Upsilon_i - w_i x_i| \tag{14}$$

Another, factor influencing the kernel response is $\mu$, that is dependent on the number of predictors. Figure 4.17 illustrates a SVM hyperplane with epsilon values defining the margin of error. The epsilon provides flexibility in optimizing the model by providing a soft margin limit that increases the model accuracy. As the value of epsilon increases, more data points are included near the hyperplane resulting in learning more points. However, it must be noted that further increasing its value may include outliers or overlapping values that may affect model accuracy. The outliers were mostly removed, however the overlapping points near the soft margin boundary resulted in extended support vector classifications, thus increasing the convergence time as well as adversely affecting the model accuracy. SVM have a genius kernel trick that enables to compute the dot product in a higher order Z-dimensional space. A typical dot product may expand linearly, with a larger dataset resulting in an

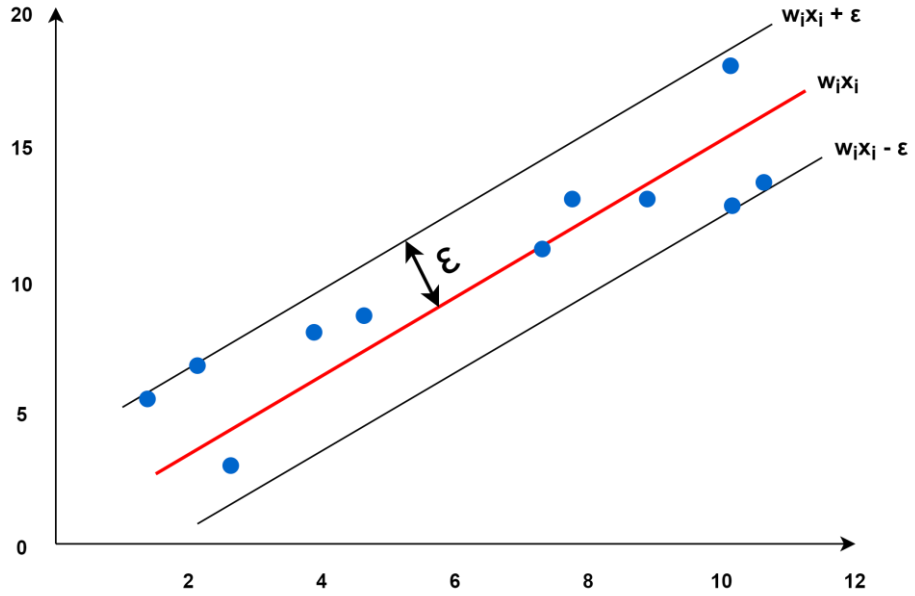



Figure 4.17: An illustration SVR hyperplane and error margin boundary.

exponential increase in computational time. SVMs utilize generalized dot product for simplicity, thus provides an efficient method to compute the support vectors, and the predicted variables. The kernel function computes Gram Matrix for n number of vectors, that are an inner product of predictor variables. In nonlinear SVM implementations, a Gram Matrix is created using the rows of predictor variables X. The function formula used to compute the Gram matrix are presented in Table 4.3.

Table 4.3: Kernel functions used to develop Gram Matrix for predictor variables

| Function Name | Description | Formula |
|---|---|---|
| Linear | Enter Linear kernel details here | $G(x_j, x_k) = x_j' x_k$ |
| Gaussian | Gaussian or Radial Basis Function (RBF), due to Gaussian-like data distribution. | $G(x_j, x_k) = \exp\left(-\|x_j - x_k\|^2\right)$ |
| Polynomial | Polynomial kernel of $n^{th}$ order. In these experiments, a polynomial of $3^{rd}$ order was used (based on section 0). | $G(x_j, x_k) = \left(1 + \acute{x}_j x_k\right)^n$ |

As mentioned earlier, three kernels (linear, polynomial, and gaussian or radial bias function RBF) were used to compute the hyper-planes and the support vectors. The models were further optimized using Bayesian optimization with Sequential Minimal

Optimization (SMO) solver (Fan, Chen, Lin, & Joachims, 2005; Platt, 1998). The problem of finding an input that results in the minimum or maximum cost of a given objective function is known as global optimization. The objective function's form is typically complex and intractable to analyse, and it is frequently nonlinear, high dimensional, noisy, and computationally intensive to assess. The optimization of functions is an important aspect of machine learning. In most machine learning algorithms, parameters (weights, coefficients, and so on) are optimised in reference to the training data. Optimization furthermore refers to the process of identifying the appropriate set of hyperparameters for a machine learning algorithm's training. Turning it a step further, selecting training data, preparing data, and developing machine learning algorithms are all problems of function optimization.

Bayesian Optimization is a pragmatic method based on Bayes Theorem for directing an efficient and reliable search of a global optimization problem. It works by creating a surrogate function, which is a probabilistic model of the objective function that is then efficiently searched with an acquisition function before candidate samples are chosen for evaluation on the real objective function. In this regard, the performance metrics for each battery dataset is presented in subsequent sections by contrasting the effect of parameter tuning on the output estimation accuracy.

### 4.3.1 SVM Model performance of Li-Ion battery dataset

The Li-Ion battery dataset (as given by Table 3.4) was used in an 80-20 split ratio for training and validating the SVM model. It was observed that the SVM model converges quickly by identifying the hyperplane and evaluating support vectors of the dataset. However, as most of the datapoints during the battery discharge cycle lies on the similar region, the overlapping results in a lower model accuracy. The performance metrics for Li-Ion SVM model is presented in Table 4.4.

Table 4.4: Performance metrics of various SVM Kernels for Li-Ion battery dataset

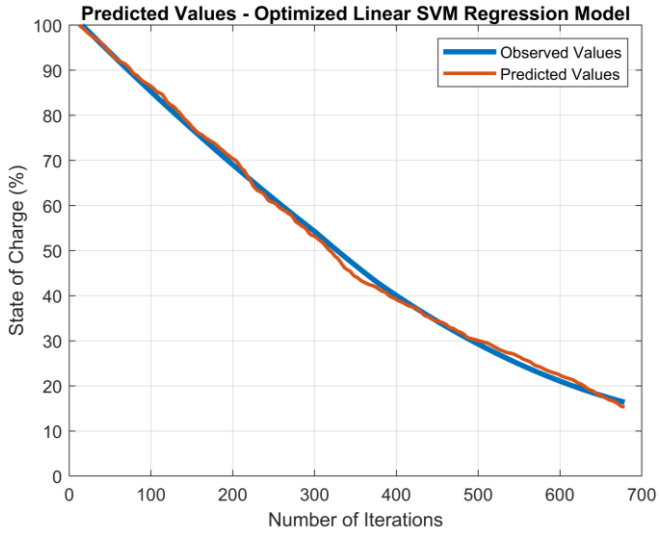
| **Performance Metrics** | **SVM Model Kernels** | | |
|---|---|---|---|
| | Linear | Gaussian | Polynomial |
| Elapsed Time (seconds) | 3.318 | 0.892 | 14.706 |
| MSE | 4.4035 | 4.6195 | 5.420 |
| RMSE | 2.0985 | 2.149 | 2.328 |
| Model Loss | 4.3380 | 4.726 | 5.449 |
| S-Index | 4.095 | 4.141 | 4.586 |
| MAE | 1.674 | 1.714 | 1.857 |
| **After Optimization** | | | |
| Elapsed Time (seconds) | 8193.67 | 2873.1002 | 8207 |
| MSE | 3.1423 | 0.0021 | 2.482 |
| RMSE | 1.7726 | 0.0465 | 1.5756 |
| Model Loss | 3.1086 | 0.0021 | 2.4083 |
| S-Index | 3.4037 | 0.0894 | 3.0409 |
| MAE | 1.4144 | 0.0371 | 1.2572 |

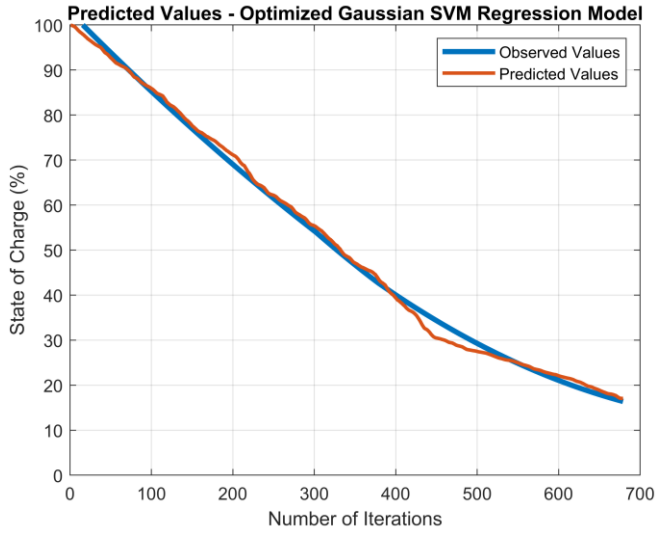
As presented, the SVM models with all three kernels converge quickly resulting in a fast-adapting technique that was implemented on the observed dataset. The fast convergence is associated with the uniform discharge profiles with little to no outliers, thus facilitating the algorithm to identify the hyperplane and respective support vectors quickly. However, it is evident that all three models underperformed with a very high MSE. As SVM model defines a boundary and then assigns support vectors to its margins, the model loss increases for both, the incorrect predictions and the ones close to the boundary. The incorrect predictions are mostly the case of outliers that can be reduced during data transformation and normalization. Whereas the overlapping sample values may result in an increase in model loss due to proximity to the hyperplane boundary.

In these experiments, the model loss was significantly high (further away from the value 1), thus resulting in an SVM with low degree of accuracy and predictability. Furthermore, the Scatter Index (S-Index) is very useful for observing SVM model performance. The S-index calculates the variance in the error spread across the evaluation function by calculating the percentage of RMS difference with respect to mean observations. In other words, the expected percentage of error per parameter can

be determined by measuring S-index, with a value close to 0 indicating an ideal model with accurate predictions and a large value indicating an increase in parameter prediction error. It was observed that underperforming models with higher RMSE values affect model accuracy and thus yielded a higher S-index score. Similarly, the Mean Absolute Error (MAE), that gives the mean error between set of observations, is also high resulting in variance between model predictions. The SOC estimation of Li-Ion batteries using SVM kernel is presented in Figure 4.18. Gaussian based kernel performs well and predicts accurately for resulting in a good degree of RMSE, SI, and MAE scores. Gaussian Kernel is a good starting point for parameter estimation because it computes the Euclidean Distance between two vectors (or data points). It also has a high degree of flexibility thanks to the sigma parameter, which helps to smooth the function. Sigma ($\sigma$) and mu ($\mu$) from Gaussian Kernel parameters were adjusted to find the balance between bias and variance to achieve good model performance and avoid overfitting.

The max deviation point (as given by Figure 4.19) is a good measure to observe the vector indices that suffer from improper estimation. The Gaussian based SVM kernel suffers performance drops at the highlighted data points. This is due to the concentration of very large overlapping values in the dataset. One way to improve this is achieved by normalizing the data in segments based on a moving average window. However, this may also result in a normalized data with lesser number of SoC values thus, fewer estimations in real-time application use. Therefore, the standard data normalization was used throughout the experiments and optimization was performed on Kernel and hyperparameters tuning alone.


Predicted Values - Optimized Linear SVM Regression Model
Observed Values
Predicted Values
State of Charge (%)
Number of Iterations
0
10
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500
600
700


Predicted Values - Optimized Gaussian SVM Regression Model
Observed Values
Predicted Values
State of Charge (%)
Number of Iterations
0
10
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500
600
700

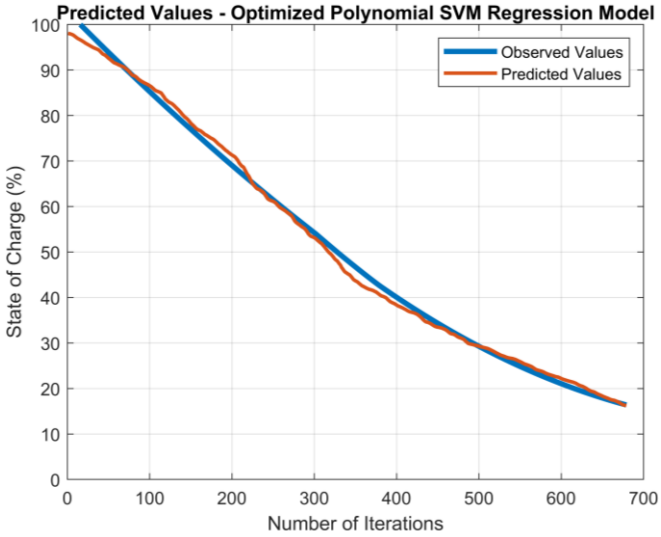

Predicted Values - Optimized Polynomial SVM Regression Model
Observed Values
Predicted Values
State of Charge (%)
Number of Iterations
0
10
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500
600
700

Figure 4.18: Predicted SOC values for Li-Ion batteries using three SVM Kernel models

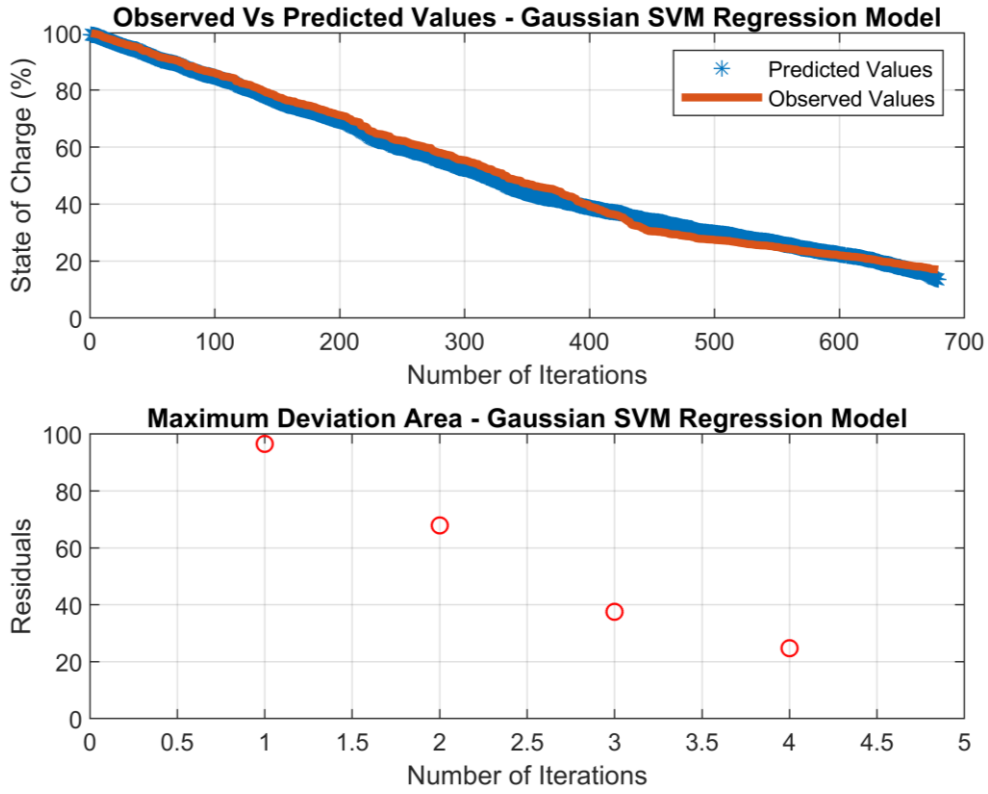



Figure 4.19: Maximum deviations resulting in residuals plotted at error indices for Li-Ion battery dataset

The maximum deviations are indicated by the ratio of the residuals plotted against the iteration intervals of the battery discharge curves. As evident, the most prominent deviations occur between iterations (100 and 200) and then (300 and 400). Additionally, the model starts converging and achieves a good fit. The highlighted region is mostly the flat region (as illustrated previously in Figure 3.7) where the values are mostly overlapped, thus resulting in boundary margin errors of the SVM kernel. Although the model training times increased a lot during optimization, the model accuracy and predictability score has improved significantly. The optimized SVM algorithm with Gaussian kernel is presented in Figure 4.20 with the iterative selection of hyper tuned parameters.

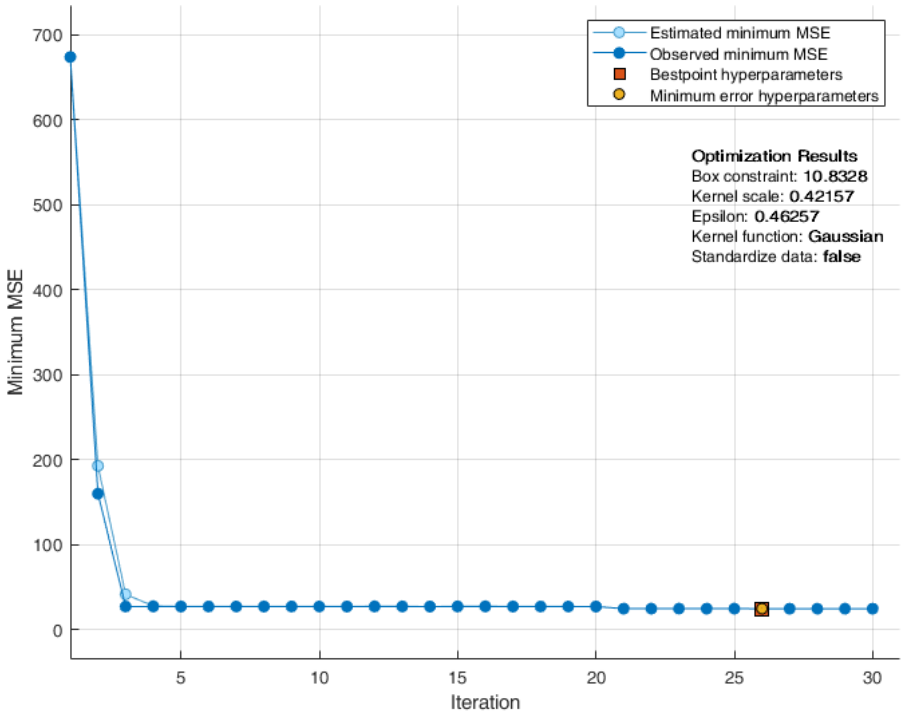



Figure 4.20: Gaussian Kernel based SVM optimization on Li-Ion battery dataset

### 4.3.2 SVM Model performance of Li-Po battery dataset

A similar approach was implemented on Li-Po battery datasets to observe the SVM model efficacy using multiple kernels. Initially, three independent models were trained using kernels (linear, gaussian, and polynomial). Later, these SVM algorithm, and kernels were optimized to achieve hyper tuned parameters that offer the best performance results. It was observed that like Li-Ion experiments, the SVM models for Li-Po battery datasets converged very quickly resulting in usable performance metrics. The performance scores are detailed in Table 4.5. The models were trained quickly resulting in almost negligible training times. The preliminary models with all three kernels converged quickly, similar to the model training on Li-Ion batteries. However, as evidenced by RMSE and MAE scores, the high degree of variance in estimated values indicates poor model performance. Furthermore, these models had a high S-index, resulting in higher model variance. Therefore, all three kernels were optimized for kernel and parameter optimizations using the same optimization scheme as previously discussed.

Table 4.5: Performance metrics of various SVM Kernels for Li-Po battery dataset

| **Performance Metrics** | **SVM Model Kernels** | | |
|---|---|---|---|
| | Linear | Gaussian | Polynomial |
| Elapsed Time (seconds) | 2.0377 | 0.221 | 0.238 |
| MSE | 5.5354 | 5.4716 | 6.6665 |
| RMSE | 2.3528 | 2.3392 | 2.5820 |
| Model Loss | 5.4419 | 5.4684 | 6.8962 |
| S-Index | 5.5228 | 5.5774 | 6.1177 |
| MAE | 1.8772 | 1.8664 | 2.0601 |
| **After Optimization** | | | |
| Elapsed Time (seconds) | 3105.2905 | 587.8419 | 5052.2289 |
| MSE | 0.000301 | 0.0025 | 0.0024 |
| RMSE | 0.0182 | 0.0501 | 0.0493 |
| Model Loss | 0.00324 | 0.0025 | 0.0026 |
| S-Index | 0.0431 | 0.1187 | 0.1169 |
| MAE | 0.0145 | 0.0399 | 0.0393 |

In this regard, the tuned models performed exceptionally well, with extremely low RMSE and MAE scores, resulting in highly efficient models. For Li-Po battery datasets, the linear kernel outperformed the Gaussian and polynomial-based kernels almost identically. Typically, it is known that not enough training data leads to a bad approximation. If a model is too constrained, it won't be able to fit the small amount of training data, and an under-constrained model, will probably overfit the training data, which leads to poor performance. Too few tests will give an overly optimistic and high variance estimate of how well the model will work.

However, in these experiments the ratio of training data (10%) was maintained throughout the experiments. Furthermore, the average performance over repeated testing was used to account for the variance in the neural network solution. Fitting the very same algorithm over the same dataset repeatedly required varying the train-validation data split ratio each time the algorithm was run. Each run, the model is evaluated on the same test set, and the score is recorded. The distribution of the scores is enumerated using a mean and standard deviation at the end of all repetitions.

The average performance of a model over repeated testing provides an estimate of the model's average performance on a given dataset. The standard deviation or spread of the scores indicates the amount of variance introduced by the supervised learning. Because it was discovered that the size of the training dataset influences model performance, the best performing 90-10 percent ratio was chosen. This also demonstrated that the performance of the linear SVM kernel for Li-Po batteries (as opposed to the Gaussian kernel for Li-Ion batteries) is entirely dependent on the dataset for that battery type. The model performance in terms of estimated SOC values is presented in Figure 4.21. From an initial investigation all three kernels perform significantly well and very close to each other. The model behavior for Li-Ion battery datasets illustrated a high degree of variance, which maybe associated to its battery discharge trend, and hence the shift of hyper plane and support vectors. To further investigate the performance of linear SVM kernel for Li-Po batteries, the residual plots with maximum deviations was computed to estimate the model performance as illustrated in Figure 4.22.

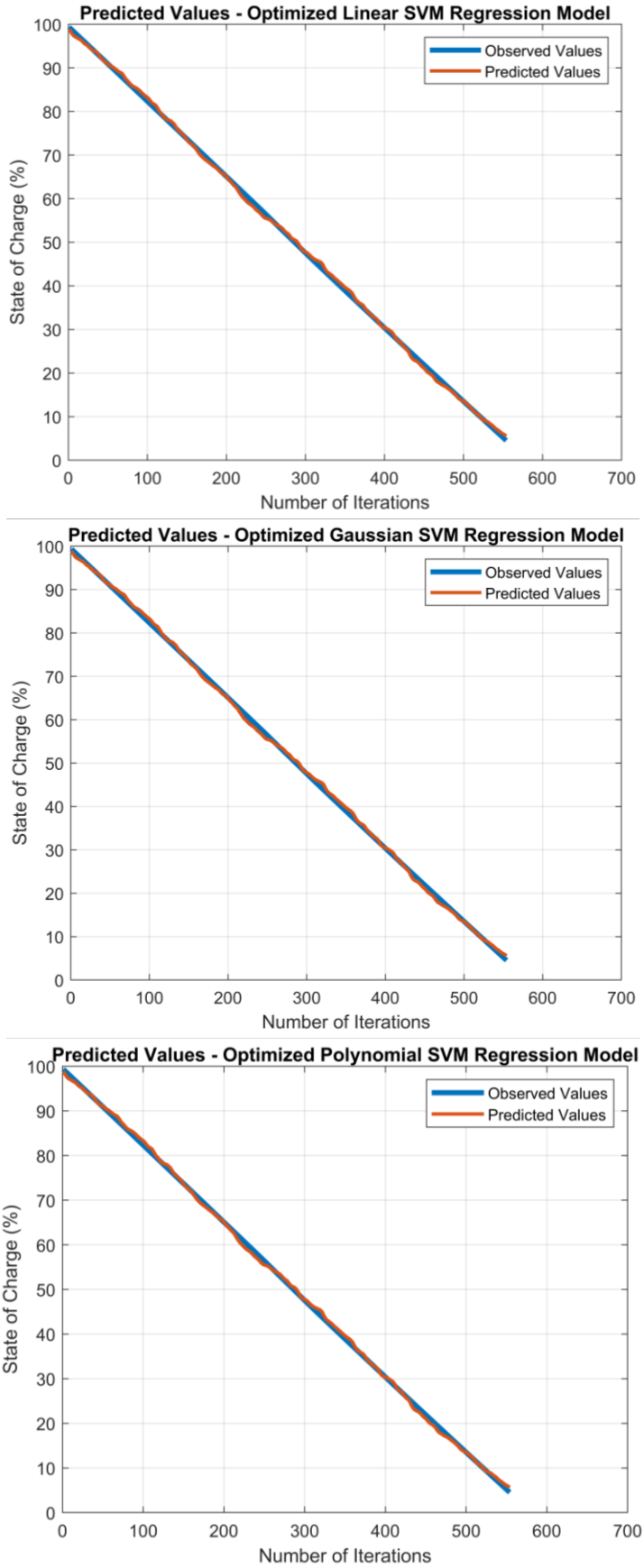



Figure 4.21: Predicted SOC values for Li-Po batteries using three SVM Kernel models

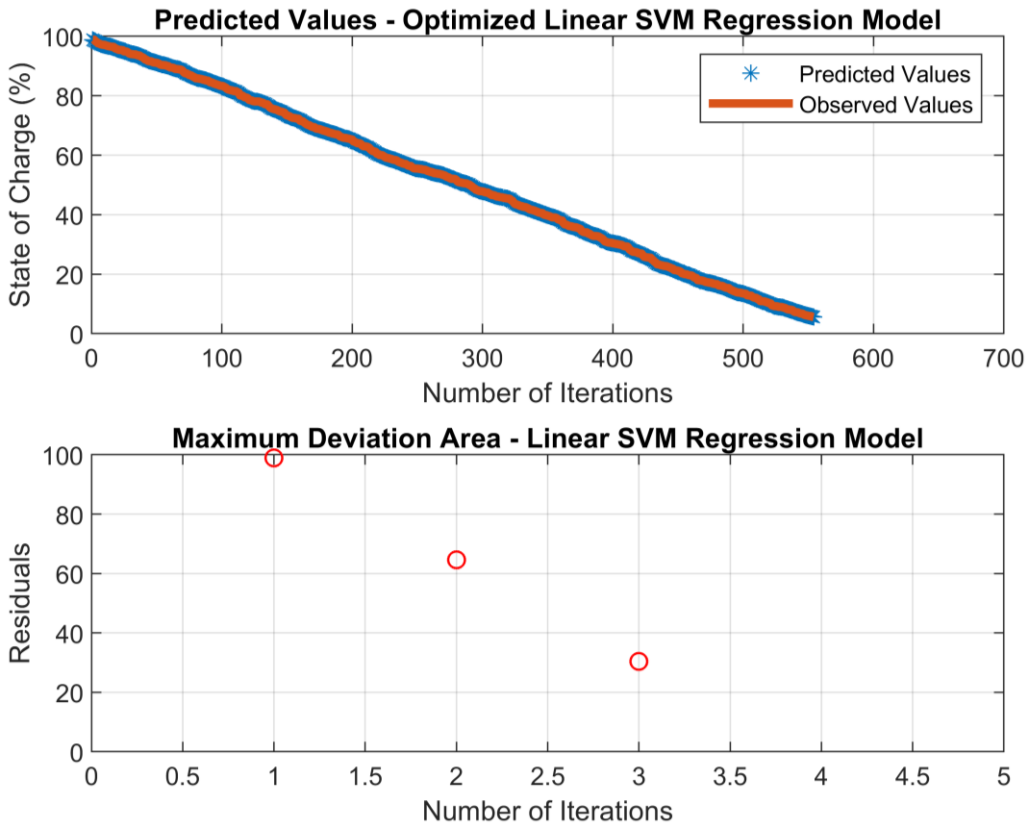



Figure 4.22: Maximum deviations resulting in residuals plotted at error indices for Li-Po battery dataset

It was observed that most of the residuals were reported around the flat battery discharge region. The overlapping sample of this region affects the hyper plane placement and thus the model accuracy. However, the number of residuals is negligible, thus illustrate the model efficacy. Finally, the SVM algorithm with linear kernel was further optimized to record the best performing model parameters for Li-Po based battery datasets.

### 4.3.3 SVM Model performance of Ni-MH battery dataset

Unlike Lithium-based batteries that exhibit somewhat similar discharge profiles, Ni-MH batteries have a very profound and distinguishable discharge profile (as given by Figure 3.7). Ni-MH batteries illustrated a lesser impact of temperature on battery performance and a flatter profile. As the discharge curves follows a near-perfect cubic equation, the initial estimation was to focus on the polynomial kernel for these

experiments. Keeping the training-validation data ratio consistent with previous experiments, the models were initialized, and the performance scores were recorded, that are given in Table 4.6.

Table 4.6: Performance metrics of various SVM Kernels for Ni-MH battery dataset

| **Performance Metrics** | **SVM Model Kernels** | | |
|---|---|---|---|
| | Linear | Gaussian | Polynomial |
| Elapsed Time (seconds) | 1.2578 | 0.2401 | 0.4517 |
| MSE | 2.9181 | 2.3773 | 3.5218 |
| RMSE | 1.7083 | 1.5419 | 1.8766 |
| Model Loss | 3.0743 | 2.2554 | 3.6415 |
| S-Index | 3.5123 | 3.1630 | 3.8633 |
| MAE | 1.3630 | 1.2302 | 1.4973 |
| **After Optimization** | | | |
| Elapsed Time (seconds) | 110.8846 | 925.5843 | 5199.6228 |
| MSE | 0.000808 | 0.0261 | 0.000675 |
| RMSE | 0.0283 | 0.1616 | 0.0260 |
| Model Loss | 0.000840 | 0.0257 | 0.00077 |
| S-Index | 0.0580 | 0.3313 | 0.0533 |
| MAE | 0.0226 | 0.1289 | 0.0207 |

The polynomial-based kernel provided by the most accurate SOC estimation for Ni-MH batteries. The cubic order polynomial kernel optimization is presented in Figure 4.23. As illustrated, the model converges quickly, however it is very sensitive to the polynomial order. Additionally, it was observed that the hyperplane boundary variance, as controlled by ($\varepsilon$) also had significant effect on model sensitivity and accuracy.

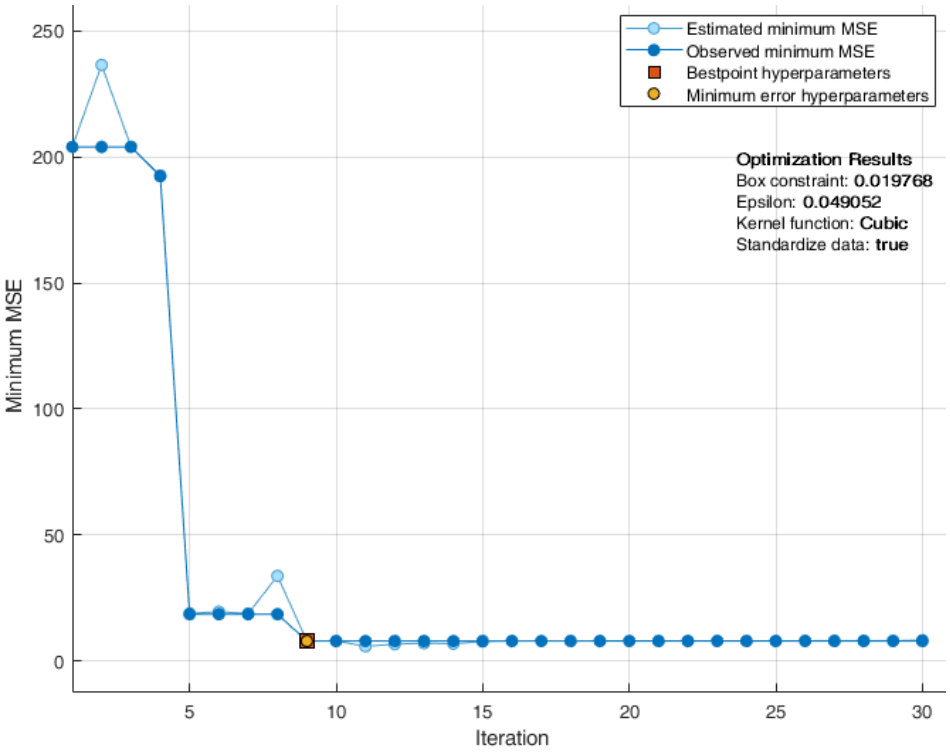
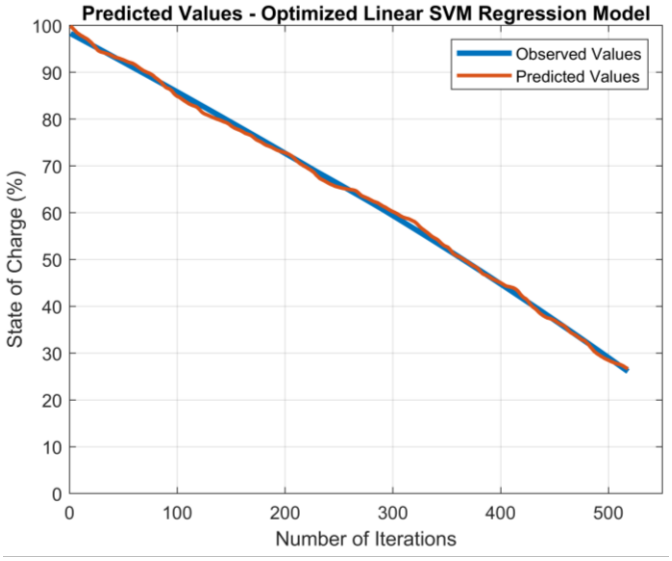
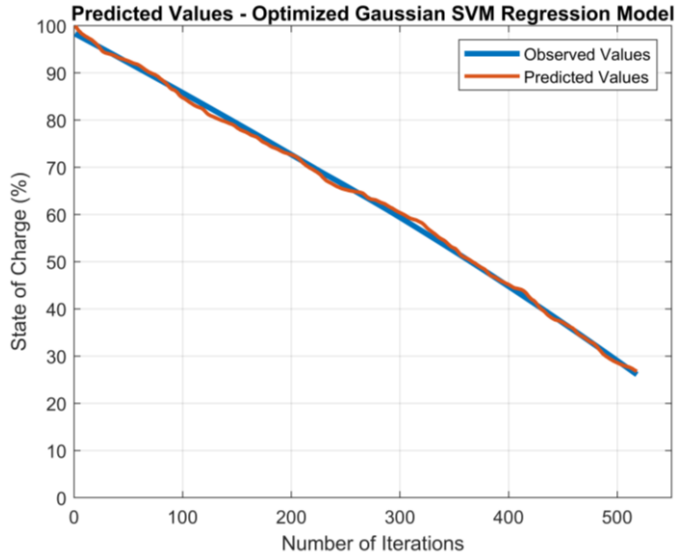


Figure 4.23: Polynomial Kernel based SVM optimization on Ni-MH battery dataset

The optimized polynomial outperforms the other kernels and produces accurate SOC estimation as given in Figure 4.24. The Ni-MH battery optimised variants function quite similarly to one another, with minimal differences. This is related to Ni-MH's discharge properties, which are very linear-like flat discharge curves with few outliers.

The performance benefits for each modified kernel were significant, but the cubic polynomial-based kernel performed the best. In this situation, the cubic order equation considerably improved the mapping of the hyperplane and support vectors. However, it should be highlighted that polynomial-based kernels necessitate a significant amount of training time. Because just the trained model is installed on the device, the training period may not appear to be an issue in real-world deployment. However, this also applies to model sensitivity, where a small change in the observed dataset (particularly the real-time values) might result in bigger offsets because on-device training is not always possible.


Predicted Values - Optimized Linear SVM Regression Model
Observed Values
Predicted Values
State of Charge (%)
Number of Iterations
0
10
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500


Predicted Values - Optimized Gaussian SVM Regression Model
Observed Values
Predicted Values
State of Charge (%)
Number of Iterations
0
10
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500

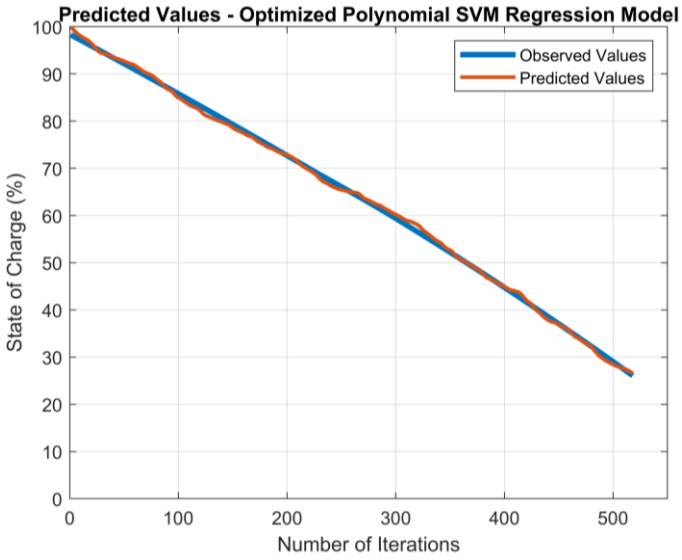

Predicted Values - Optimized Polynomial SVM Regression Model
Observed Values
Predicted Values
State of Charge (%)
Number of Iterations
0
10
20
30
40
50
60
70
80
90
100
0
100
200
300
400
500

Figure 4.24: Predicted SOC values for Ni-MH batteries using three SVM Kernel models

The model sensitivity can be compared by looking at the residual plots for each kernel, as detailed in Figure 4.25. The residuals for each kernel are extremely minimal, resulting in a model estimation accuracy of above 98 percent. The polynomial-based kernel was chosen for this dataset because it has the lowest degree of errors.

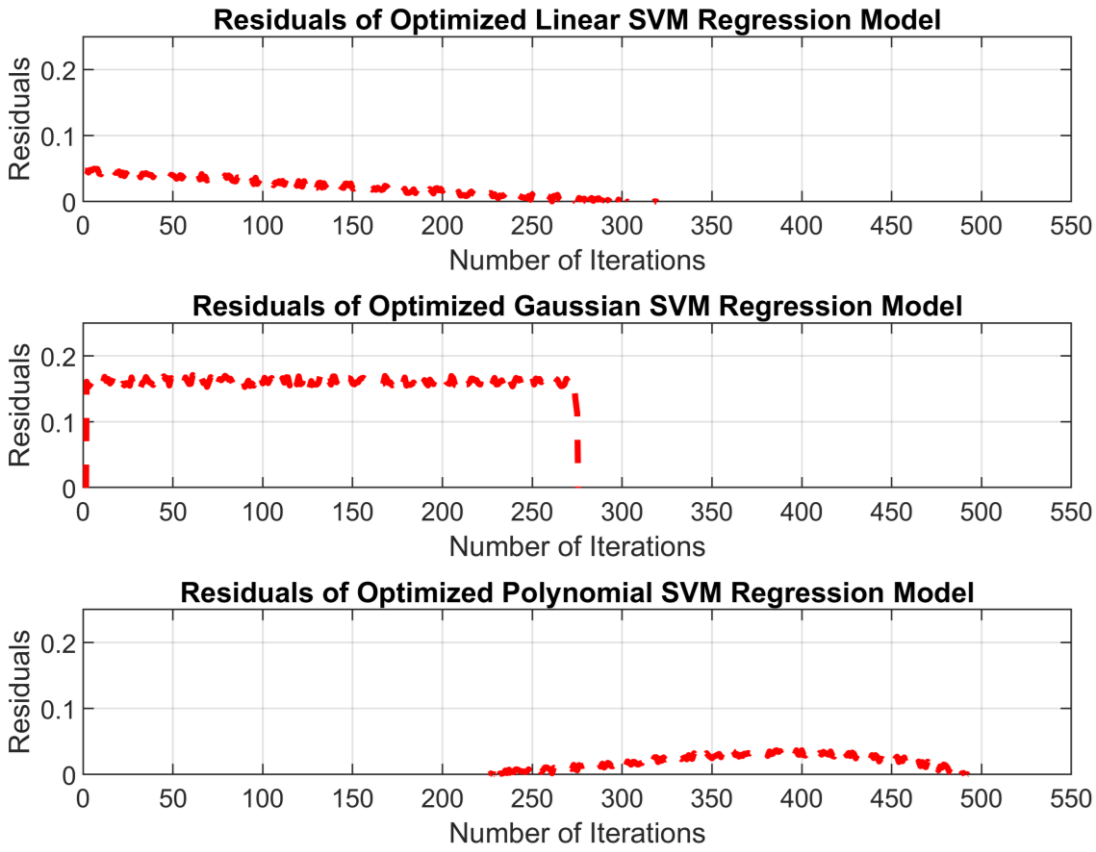



Figure 4.25: The residual plots for the batteries under various optimized SVM kernels.

### 4.3.4 Short Summary

Model architectures were shown to be extremely sensitive on the battery discharge dataset, to the point where a single feature may have a substantial impact on the model architecture, forcing a new design in some circumstances. Certain models, such as polynomial-based kernels, may perform well for each dataset if the discharge curve equation order is known. Other datasets require custom-tuned kernels that are

extremely sensitive to feature changes. The dataset can also change Therefore of the effect of input factors like voltage, current, and temperature. SVM kernels also attempt to detect hyperplanes and correlate support vectors with them for estimation. The selection of the optimal hyperplane is difficult since the battery discharge curves have a typical cubic order form with overlapping sections. Therefore, model sensitivity increases, potentially improving estimation accuracy. Finally, different SVM kernels may be necessary to handle all battery datasets, according to the findings. For Li-Ion batteries, for example, Gaussian kernels performed best, but Linear kernels performed best for Li-Po battery datasets. Whereas cubic order polynomial kernels were shown to be the most accurate for Ni-MH estimate. Therefore, it's difficult to create a single SVM model that can handle diverse battery chemistries, or, in other words, datasets with different operating conditions.

The major problem is the model's predictability uncertainty, which is marginable and does not directly affect the model's accuracy. That instance, a model might function well for one sort of data, but its applicability to another is unknown. Adding numerous trained networks to an embedded platform is also not a realistic option due to the additional storage, memory, and computation time required. Another important consideration is that unlike linear and logistic regression, SVM's cost function is convex, resulting in a guaranteed global minimum. This property may help to match any possible discharge curve, but may alter the output accuracy, which in this case leads to faulty SOC estimation. It is highly improbable that a model can be tuned to perform well across the entire dataset for all batteries. Due to the similarity of the battery discharge profiles, a close-enough model can be trained to accurately estimate SOC; however, model inaccuracies cannot be avoided. In summary, for many features, a linear SVM model may perform the best. Whereas, for smaller and flatter datasets,

Gaussian-based kernels normally performs the best. Finally, because of this sensitivity on the dataset, Gaussian Process Regression (GPR) based models were examined in detail.

## 4.4 Gaussian Process Regression (GPR)

Gaussian processing (GP) technique is normally deployed for non-parametric Bayesian modeling. In regression-based problems, the GP technique focuses on marginalization and conditioning of a Gaussian distribution function. Typically, a multivariate Gaussian distribution is analyzed by the relationship between mean vector data and its covariance. Unlike neural networks models where the network structure is formed on perceptron weights and bias at every layer, GP modeling directly maps the mean vector sector of the covariance matrix that satisfies the boundary conditions, resulting in fully connected network input matrix in Gaussian space, as illustrated by Figure 4.26.

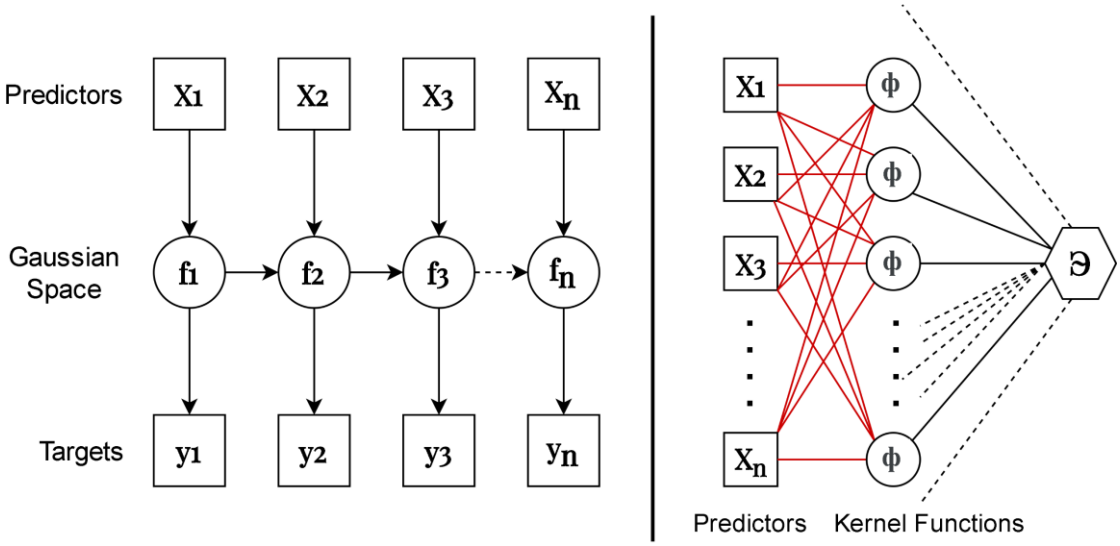



Figure 4.26: Gaussian Process Regression kernel parameters mapping

In every regression-based ML problem, the goal is to calculate the function that closely fits the input dataset. In addition, not only a close-fitting function is desirable, but the certainty of predictions must also be calculated. GP-based regression is widely

used to address these goals. In contrast to many popular supervised machine learning algorithms, which learn exact values for each parameter in a function, the Bayesian approach presumes a probability distribution over all distinct combinations. For a linear function that takes the form $y = wx+\in$, the Bayesian approach works by defining a prior distribution, p(w), on the component, w, and adjusting probabilities based on empirical evidence (observed data) using Bayes' rule as given in Equation (15).

$$posterior = \frac{prior\ x\ Likelihood}{marginal\ Likelihood}\ , p(w|y,X) = \frac{p(y|X,w)p(w)}{p(y|X)} \quad (15)$$

The posterior distribution *p(w|y,X),* contains information both from the dataset as well as the prior distribution. This enables to calculate predictive distributions at future intervals $x^*$, by weighing all posterior distributions, as given by Equation (16).

$$p(f^*|x^*,y,X) = \int_w p(f^*|x^*,w)p(w|y,X)dw \quad (16)$$

Because Gaussian process regression is nonparametric, it calculates the probability distribution over all probative functions that fit the data rather than the probability distribution of parameters of a specific function. Nonetheless, prior (on the function space) is specified to compute the posterior using the training data and calculate the predictive posterior distribution on the points of interest, as described above. Therefore, GPR functions assumes Gaussian distribution for input vector space. Typically, a function is said to have Gaussian distribution if the probability density function (PDF) for its random variable (x) is normally distributed with a mean ($\mu$) and variance ($\sigma^2$) and is given by Equation (17).

$$P_X(x) = \frac{1}{\sqrt{2\pi\sigma}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (17)$$

Where (X) represents a random variable, and ($x$) is the actual input vector. The Gaussian distribution can be estimated as Equation (18),

$$P(x) \sim \mathcal{N}(\mu, \sigma^2) \tag{18}$$

As evident, Gaussian PDF is characterized by ($\mu$) and ($\sigma$) that is obtained from (3). This holds for 1-D or univariate input vectors$D = [x^1, x^2, \ldots x^m] \Longrightarrow x^i \in R$. However, as the intended model requires two input vectors (battery voltage ($x1$), and capacity ($x2$)), a multi-variate Gaussian distribution in 2-D vector space was needed. In this regard, the PDF of input vectors are computed together by utilizing multi-variate Gaussian distribution as given by Equation (19).

$$P(X) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\,(x-\mu)^{T-1}\Sigma(x-\mu)\right\} \tag{19}$$

Where $\begin{cases} D \longrightarrow & no.of\ Dimensions \\ x \longrightarrow & input\ variable \\ \mu \longrightarrow & Mean\ vector \\ \Sigma \longrightarrow & Covariance\ Matrix \end{cases}$

The mean vector ($\mu$) is a 2-D vector ($\mu_1, \mu_2$) independent to input variables ($x_1$) and ($x_2$). The covariance matrix in 2-D Gaussian vector space can thus be computed as, $\begin{pmatrix} \sigma_{11}^2 & \sigma_{12} \\ \sigma_{21} & \sigma_{22}^2 \end{pmatrix}$, resulting in a 2-D Gaussian process as given by Equation (20).

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{pmatrix}\right) \sim \mathcal{N}\left(\mu, \Sigma_{i,j}\right) \tag{20}$$

Where, $\Sigma_{ij} = \Sigma(x_i, x_j)$ are covariance kernel parameters. Now, if variables ($x_1$) and ($x_2$) are highly correlated vectors that follow a normal distribution, the covariance function could be considered as the dot product that helps to smooth the samples at each $i^{th}$ iteration. In this context, an R-squared or Radial Bias Function (RBF) kernel was used to smooth the variance between input vectors, which is given by Equation (21).

$$cov\ K\left(x_i, x_j\right) = exp\left(-\frac{\left(x_i - x_j\right)^2}{2}\right) \tag{21}$$

For the problem set under discussion, the training dataset may take the form $\{(x_i, y_i) = For\ each\ i = 1,2, \ldots n\}$, and the predicted responses for test outputs can be given as $(x^*, y^*)$. The multi-variate Gaussian distribution for the input vector sample was calculated using Equations (22)-(25).

$$P(f|X) = \mathcal{N}\left(f_{ij}\middle|\mu_{ij}, K_{ij}\right) \tag{22}$$

Where

$$f = \left(f(x_i), \ldots f(x_n)\middle|f\left(x_j\right), \ldots f(x_n)\right) \tag{23}$$

$$\mu = \left(m(x_i), \ldots m(x_n)\middle|m\left(x_j\right), \ldots m(x_n)\right) \text{ and } K_{ij} = K\left(x_i, x_j\right) \tag{24}$$

In practice, access to noisy versions of function values is realizable, thereof, $y = f(x) + \in$. Given an additive independent and identically distributed Gaussian noise with variance $\sigma_n^2$, the prior on the noisy observation becomes, $cov(y) = K + \sigma_n^2 I$. This leads to the joint distribution as given below.

$$\begin{bmatrix} y \\ y^* \end{bmatrix} = \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma^2 I & K(X,x^*) \\ K(X^*,x) & K(x^*,x^*) \end{bmatrix}\right) \tag{25}$$

The prediction equations for Gaussian processes regression are obtained by deriving the conditional distribution. Finally, the conditional distribution and predicted variances are given by Equations (26), and (27) respectively.

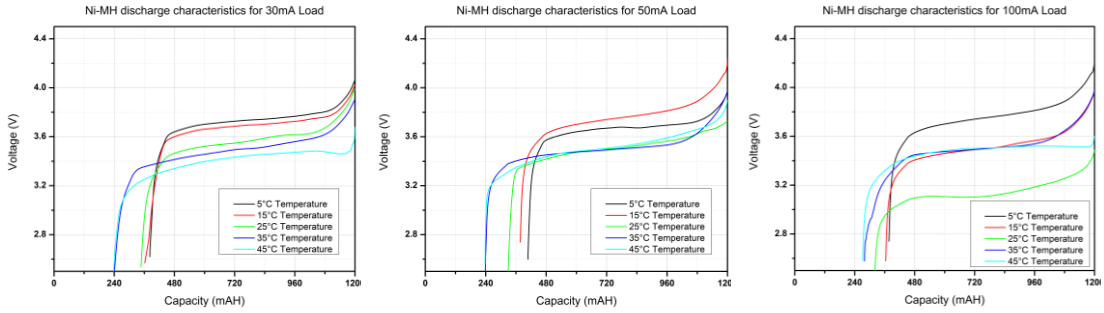
$$\vec{y}^* = K(x^*, X)(K(X,X) + \sigma^2 I)^{-1} Y \tag{26}$$

$$\Delta^2 = K(x^*, x^*) - K(x^*, X)(K(X,X) + \sigma^2 I)^{-1} K(X, x^*) \tag{27}$$

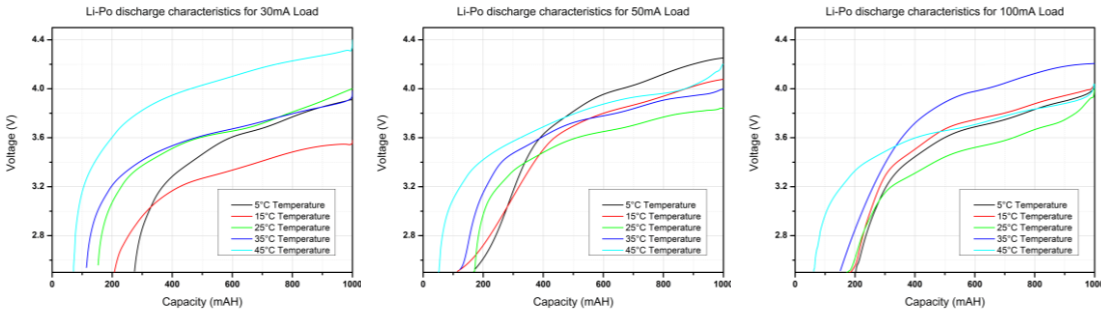
### 4.4.1 GPR Results

As previously mentioned, three different battery chemistries were examined at five temperature ranges in this study. Previously, the models discussed in preceding

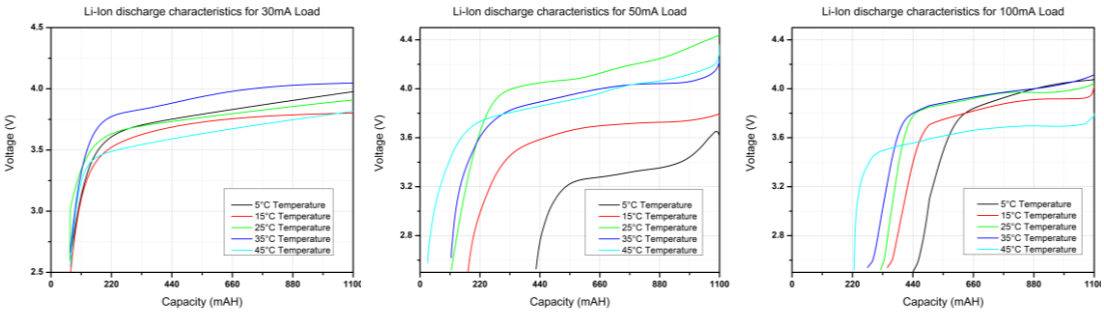
sections (polynomial regression, FFNN, and SVM networks) were developed using battery datasets at room temperature (25˚C). However, once their efficacy was reported, it was imperative to evaluate their performance with GPR models at all operating temperatures. The discharge characteristics of the batteries are discussed in Figure 4.27.




(a) – Ni-MH (Batt1) discharge characteristics over various temperature ranges and loads.




(b) – Li-Po (Batt2) discharge characteristics over various temperature ranges and loads.




(c) – Li-Ion (Batt3) discharge characteristics over various temperature ranges and loads.

Figure 4.27: Discharge characteristics of various batteries under drive cycles and temperatures.

In addition, the effect of data partitioning (data split ratio) was also observed during performance evaluation. In this regard, initially, all models utilized an 80-20% split, which was later compared to 70-30% split values. Furthermore, k-fold cross-validation (5-fold training data split) was performed to hyper-tune the model parameters to achieve minimum data leakage and optimum efficiency. These statistical indices are

discussed in Table 4.7. The correlation matrix enabled us to estimate the strong relationship between extracted features that affects the model precision and predictability. It was observed the polynomial and regression-based techniques established a strong correlation value compared to SVM. Although the correlation factor is statistically computed for the extracted features, the ML models use the correlation index to build and compute this value based on the amount of data used for training, and validation. Therefore, it is very crucial to optimize the model with various split ratios to estimate the best fit results.

As mentioned in Table 4.7, the FFNN model for (Batt1) can predict the battery SoC at a high degree of accuracy with very small RMSE and MAE values. However, at lower temperature datasets 5°C and 15°C, the model reported a small degree of errors (1.497 and 1.2515) respectively. Similarly, the SVM model due to its lower correlation between features performed significantly lower than the FFNN model. Although the model performs well for higher temperatures, the RMSE and MAE errors (4.0871, and 5.9127) at 5°C affect the model prediction accuracy. The GPR model established an accurate and predictable SoC estimation with average RMSE and MAE scores of 0.00294, and 0.002561 respectively. GPR model exhibited a slightly lower impact at lower temperature dataset which is due to its flexibility and a high degree of parameter tuning. However, as this objective aimed at building an adaptable ML model for various battery chemistries, the model estimation accuracy was further computed for other battery types at various operating temperatures.

In this regard, the FFNN model for (Batt2) presented a consistent estimation accuracy with average RMSE and MAE errors of less than 0.3, and 0.42 respectively. These values recorded a high degree of estimation accuracy with a slight increase in

RMSE errors at 5°C temperature. Whereas the SVM model provided an average accuracy score of RMSE and MAE values at 2.63, and 2.83 respectively.

Table 4.7: Model performance at various operating conditions

| Battery Type | ML Model | Correlation | Temp (C) | $R^2$ | RMSE | MAE |
|---|---|---|---|---|---|---|
| Ni-MH (Batt1) | FFNN | 0.8 | 5°C | 0.84 | 1.497 | 2.347 |
| | | | 15°C | 0.89 | 1.3515 | 2.0915 |
| | | | 25°C | 1 | 0.00296 | 0.0254 |
| | | | 35°C | 1 | 0.002153 | 0.0215 |
| | | | 45°C | 1 | 0.002031 | 0.0199 |
| | SVM | 0.7 | 5°C | 0.71 | 4.0871 | 5.9127 |
| | | | 15°C | 0.88 | 1.3245 | 2.02351 |
| | | | 25°C | 0.99 | 1.5253 | 1.3267 |
| | | | 35°C | 0.99 | 1.5404 | 1.3109 |
| | | | 45°C | 0.99 | 1.5399 | 1.3008 |
| | GPR | 0.9 | 5°C | 0.94 | 0.003012 | 0.02605 |
| | | | 15°C | 0.99 | 0.002981 | 0.02598 |
| | | | 25°C | 1 | 0.002947 | 0.02536 |
| | | | 35°C | 1 | 0.002959 | 0.02619 |
| | | | 45°C | 1 | 0.002837 | 0.02447 |
| Li-Po (Batt2) | FFNN | 0.8 | 5°C | 0.89 | 1.3491 | 2.0715 |
| | | | 15°C | 0.99 | 0.00308 | 0.0358 |
| | | | 25°C | 1 | 0.00295 | 0.0251 |
| | | | 35°C | 1 | 0.00231 | 0.02312 |
| | | | 45°C | 1 | 0.00297 | 0.0275 |
| | SVM | 0.7 | 5°C | 0.85 | 3.5971 | 4.9019 |
| | | | 15°C | 0.92 | 2.5313 | 2.3987 |
| | | | 25°C | 0.99 | 2.3039 | 2.2048 |
| | | | 35°C | 0.99 | 2.4010 | 2.3719 |
| | | | 45°C | 0.99 | 2.3121 | 2.2716 |
| | GPR | 0.9 | 5°C | 0.97 | 0.00414 | 0.0617 |
| | | | 15°C | 0.99 | 0.00301 | 0.0567 |
| | | | 25°C | 1 | 0.00292 | 0.0254 |
| | | | 35°C | 1 | 0.00267 | 0.0231 |
| | | | 45°C | 1 | 0.00300 | 0.0497 |
| Li-Ion (Batt3) | FFNN | 0.9 | 5°C | 0.97 | 1.8812 | 1.6871 |
| | | | 15°C | 1 | 1.7308 | 1.3991 |
| | | | 25°C | 1 | 1.7519 | 1.4793 |
| | | | 35°C | 1 | 1.7661 | 1.4971 |
| | | | 45°C | 1 | 1.7566 | 1.4841 |
| | SVM | 0.9 | 5°C | 0.81 | 4.5056 | 5.0918 |
| | | | 15°C | 0.88 | 3.0127 | 4.1240 |
| | | | 25°C | 1 | 1.8107 | 1.5142 |
| | | | 35°C | 1 | 1.7980 | 1.4974 |
| | | | 45°C | 1 | 1.7800 | 1.3636 |
| | GPR | 0.9 | 5°C | 0.92 | 0.00871 | 0.09908 |
| | | | 15°C | 0.99 | 0.00409 | 0.671 |
| | | | 25°C | 1 | 0.0035 | 0.020 |
| | | | 35°C | 1 | 0.0045 | 0.039 |
| | | | 45°C | 1 | 0.0067 | 0.089 |

Although the SVM model still provided a high degree of estimation accuracy, the degree of error in contrast to FFNN model is quite high. The degree of SVM

variation is likely due to the data filtering process, as the SVM model only weighs support vectors instead of features, standard normalization may affect its performance. Finally, GPR based model for (Batt3) outperformed the other two techniques with an average RMSE and MAE score of 0.0031, and 0.0043 respectively.

Finally, the performance metrics for (Batt3) revealed similar patterns, with the GPR-based technique outclassing the other techniques. Moreover, it was observed that the model accuracies are influenced by correlation values and the effect of temperature, especially at lower temperature datasets. In this study, battery voltage, capacity, and temperature were chosen as the predictors. Battery voltage and capacity presented a strong correlation whereas temperature readings had a low correlation on the other features. Another important consideration is the boundary conditions of the data-split ratio that affects the model performance. It was also observed that at a lower temperature due to the reduced electrochemical transport phenomenon, the battery discharge properties exhibit a great deal of variation. These variations can have an adverse effect on SoC and capacity calculations during the dataset extraction, which directly impacts the correlation between features, and thus the model accuracies. As evident, the models for all battery chemistries experienced a reduced performance at a lower temperature (5°C, and 15°C) when compared to the observations at room temperature (25°C).

Similarly, at instances with correlation values of less than 0.9, the MAE for all cases increased resulting in an average performance drop of approximately 6%. Therefore, a larger dataset with highly correlated features must be utilized for model training to account for feature loss during data split and especially validation. GPR model outperformed the other models and provided consistent accurate SoC estimation for all datasets.

### 4.4.2 GPR Model optimization

The model was further tuned using various kernels to estimate the accurate values of ($\sigma$) and ($\mu$) to optimize its performance. It was therefore very important to populate the $n^{th}$ degree polynomial coefficients that fit the voltage discharge curves of the training data. In this regard, a polynomial regression-based technique fits the predictive response to compute the order and coefficients of the polynomial equation, which were later used to optimize GPR kernel. Polynomial regression is ideal for the correlated feature set with non-linear behavior. However, this technique is very sensitive to outliers and can generate a large residual error resulting in a poorly fit model equation. Therefore, the normalized data was post-filtered to remove any overlapping and anomalous entries before the application of polynomial regression.

In its most simple form, the $k^{th}$ order polynomial model in one variable can be given by Equation (28).

$$y = \beta_o + \beta_1 x + \beta_2 x^2 + \varepsilon \tag{28}$$

$$y = \beta_o + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \varepsilon \tag{29}$$

Where, Equations (28), and (29) are second-order polynomials in one and two variables respectively. The $k^{th}$ order polynomial can thus be simplified using Equation (30).

$$y_n = \beta_o + \beta_1 x + \beta_2 x^2 + \cdots + \beta_m x^n + \varepsilon_i \tag{30}$$

Where,

$(m, n = 1,2, \ldots k)$

$Y = [y_1, y_2, \ldots y_n]^T$

$y_n$ is the $n^{\text{th}}$ degree polynomial equation for dependent variable Y.

$\beta_o$ is the Y-intercept of regression surface.

$\beta_m^n$ is the regression slope w.r.t $X_m^n$

$\varepsilon_i$ is the residual error at $i^{th}$ iteration.

The multi-variable quadratic polynomial equation for the dependent variable (SoC) and independent variables (voltage, capacity) can be solved as given by Equation (31).

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ 1 & x_3 & x_3^2 & \cdots & x_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{pmatrix} \begin{pmatrix} \beta_o \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{pmatrix} \quad (31)$$

Which can be simplified to, $\vec{y} = X\vec{\beta} + \vec{\varepsilon}$. The estimated polynomial regression coefficients were computed using the least square method resulting in Equation (32).

$$\vec{\beta} = (X^T X)^{-1} X^T \vec{y} \quad (32)$$

The polynomial equation revealed a 3rd order quadratic model was the best fit. This polynomial model equation provided efficient statistical scores ($R^2$ = 1, *RMSE* = 0.00293) and therefore, was used in the GPR kernel parameter optimization. The hyper-tuning process and optimal kernel parameter extraction are presented in Figure 4.28. In this regard, the quadratic function as described above was used as the basis function, to compute the sigma parameters for model optimization. The kernel approximation parameter ($\sigma$) was computed through iterative optimization for each GPR kernel. Moreover, four GPR kernels (namely Linear, Squared Exponential, Parametric, and $R^2$ exponential) were analyzed.

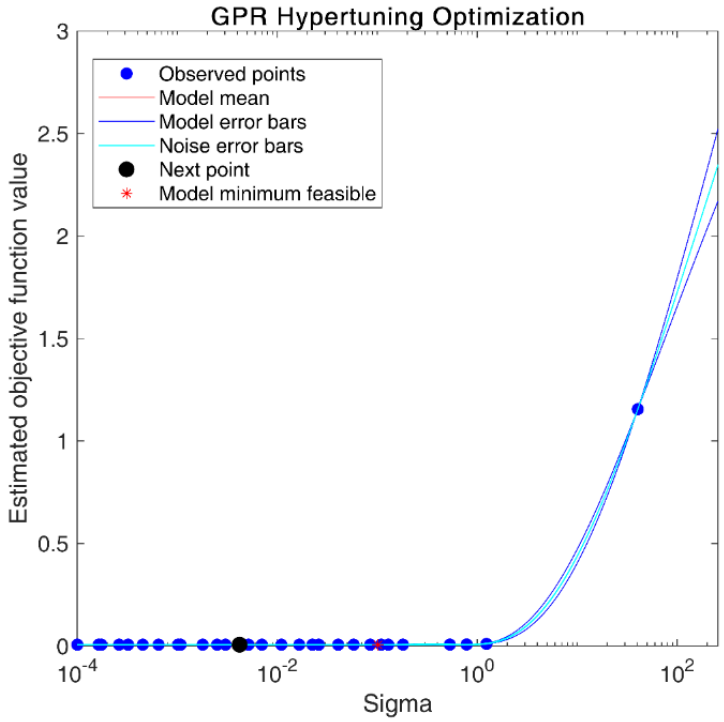



Figure 4.28: GPR model Hyper-Tuning for optimized Kernel Parameters estimations

The RBF kernel provided the best model fit and is therefore used for model extraction and deployment. The results of parameter optimization and SOC estimation for each battery is discussed in the following sections.

### 4.4.3 GPR SoC estimation for Li-Ion batteries

The optimized GPR kernel proved high degree of SOC estimation accuracy. The predicted values were found to be closely approximated with observed values for Li-Ion battery datasets for all the operational parameters, as previously described. The model response for hyper tuned kernel is presented in Figure 4.29.

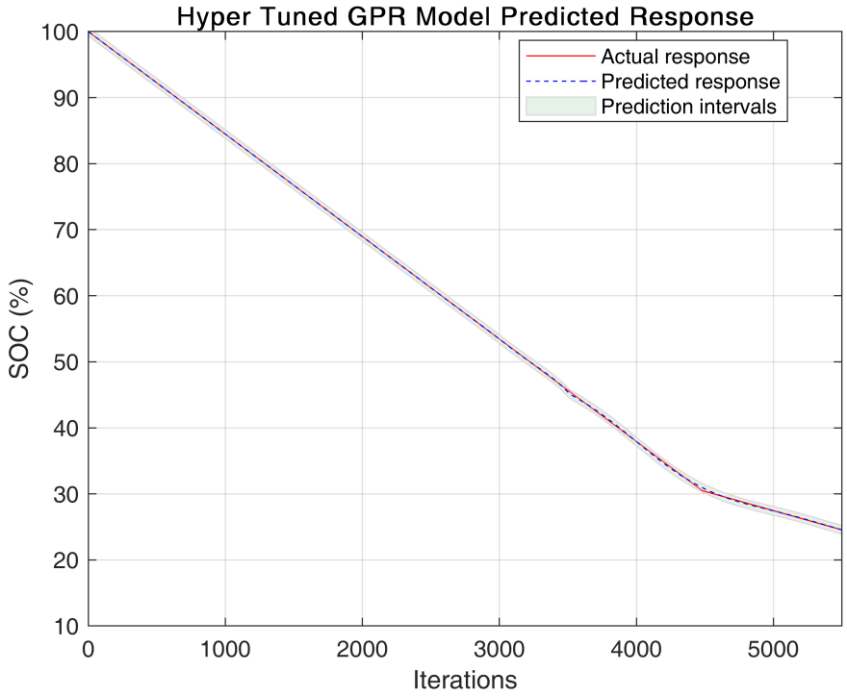


Figure 4.29: Hyper tuned GPR Kernel response for Li-Ion battery dataset

The estimated values appear to be mostly overlapping due to almost perfect estimation. Additionally, unlike FFNN and SVM models the optimized GPR kernel can accurately estimate SOC values even for the flatter region. Therefore, this results in an accurate and predictable model for all operating conditions of the battery. Additionally, the GPR model was compared with various kernels to observe the response of the best fitting model. The hyper-tuning enabled to identify the most accurate model for the available battery datasets. The performance evaluation of various kernels is presented in Figure 4.30.

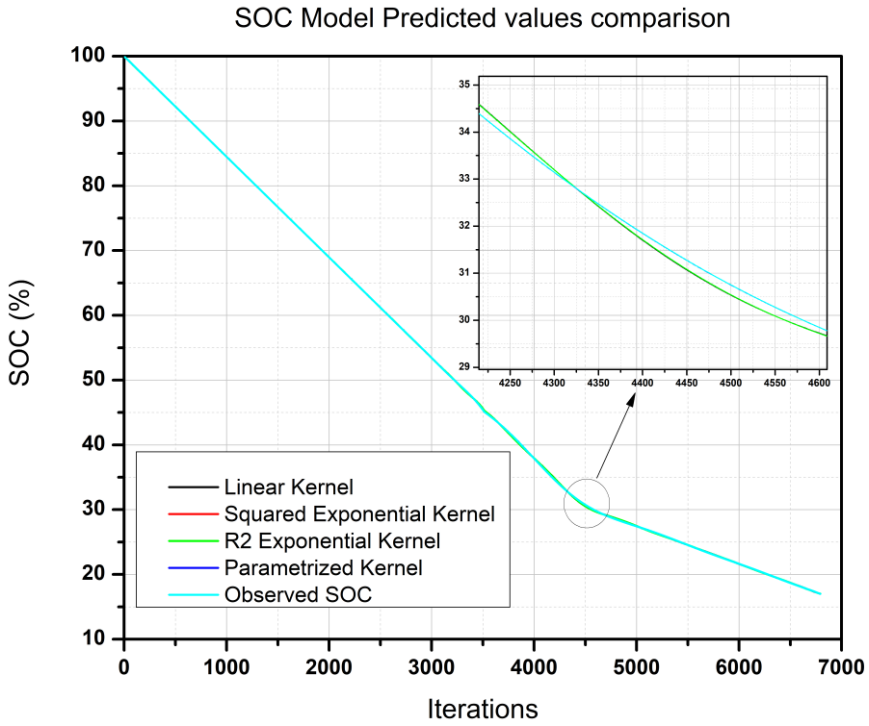


Figure 4.30: GPR model hyper-tuning kernel comparison for Li-Ion batteries

It can be noticed that all subjected kernels performed extremely well and yielded almost similar results in comparison to each other. A closer investigation further revealed a slight variation (less than 0.2%) around the end of flat discharge region, which was consistent among all kernels. Furthermore, it was observed that the Squared exponential kernel performed the best for all set of operational conditionals. This trend can be more closely observed by investigating the residual plots given in Figure 4.31.

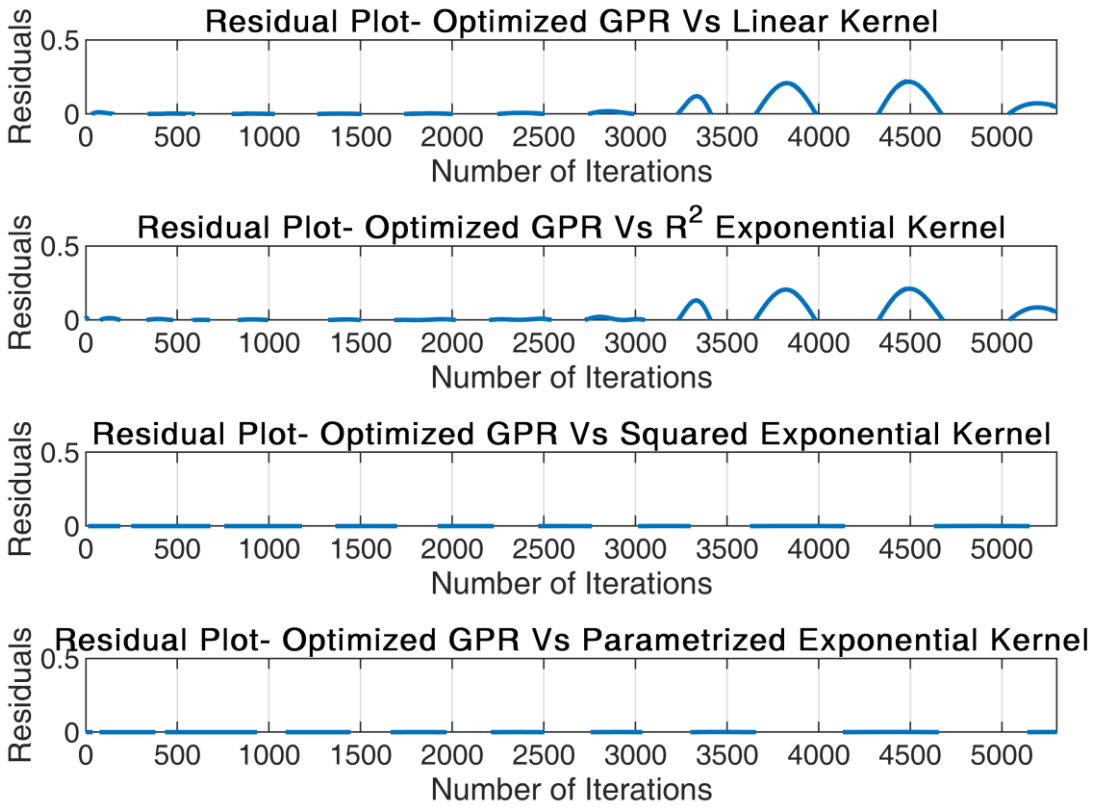



Figure 4.31: Residual plots comparing Kernel optimization for Li-Ion battery dataset

### 4.4.4 GPR SoC estimation for Li-Po batteries

A similar trend was observed for Li-Po batteries where the optimized GPR model exhibited almost near perfect SOC estimation. As illustrated in Figure 4.32, the predicted responses were almost identical to the observed values. This trend was also reported in Table 4.7 where the average RMSE score for Li-PO battery models were less than 0.0023, resulting in accurate and predictable SOC estimation. A more in-depth examination of the model performance comparison of various kernels revealed the model accuracy and marginal variation among all models. The residual plots also revealed nearly identical performance that may not be improved further. This implies that, for the Li-PO battery model, hyper-tuning produced comparable results and that the choice of a specific kernel has no additional impact on model accuracy. The highly correlated features caused by a flatter discharge profile aid in SOC estimation and model parameter hyper-tuning.

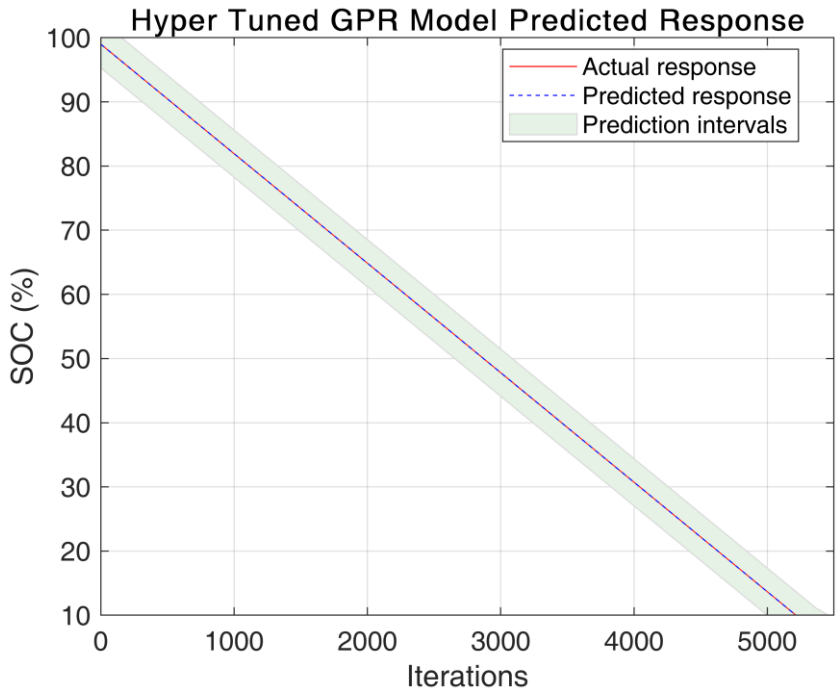



Figure 4.32: Hyper tuned GPR Kernel response for Li-Po battery dataset

These results are reported in Figure 4.33 and Figure 4.34 respectively.

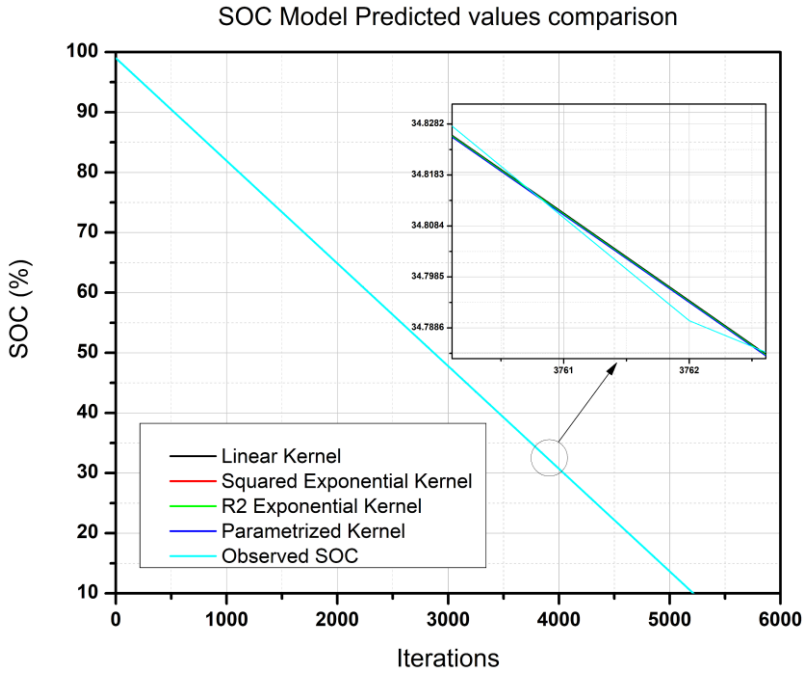


Figure 4.33: GPR model hyper-tuning kernel comparison for Li-Po batteries

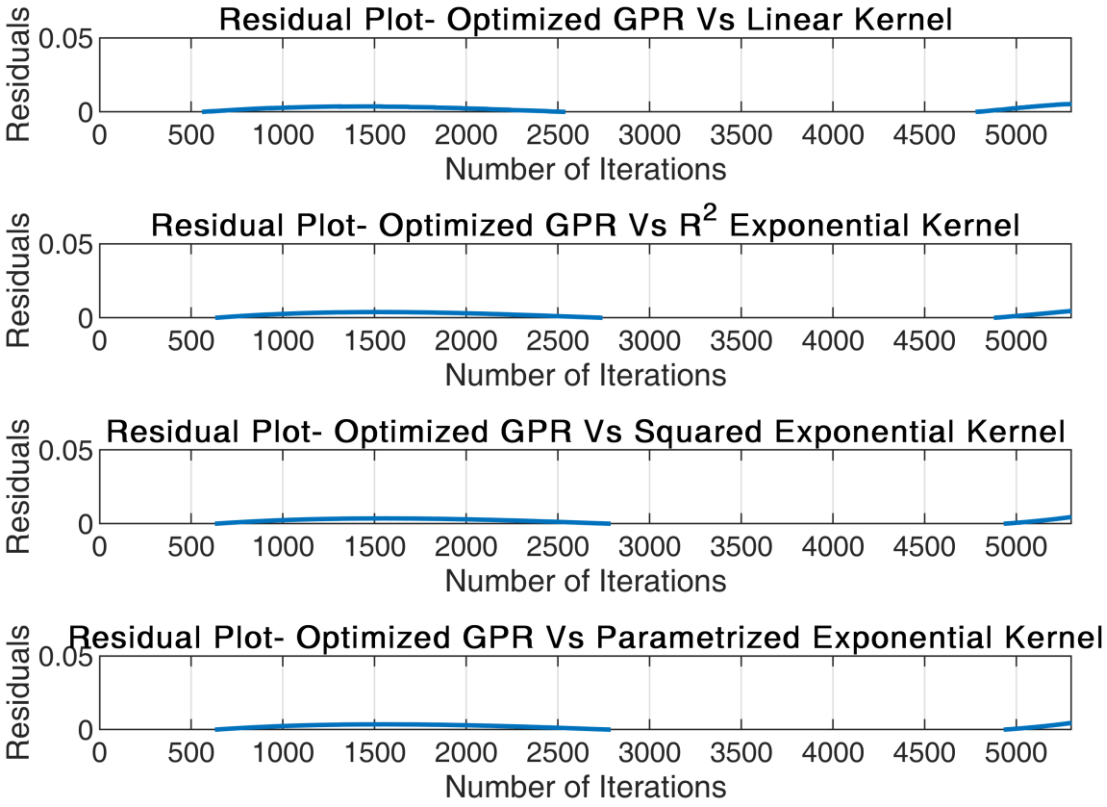



Figure 4.34:Residual plots comparing Kernel optimization for Li-Po battery dataset

### 4.4.5 GPR SoC estimation for Ni-MH batteries

Finally, the Ni-MH based model also performed significantly with accurate SOC estimation for almost entire length of the dataset. Due to the sheer flatter Ni-MH discharge profiles, there were very few outliers, resulting in a highly correlated feature-set and, Therefore, a more accurate model, as illustrated by Figure 4.35.

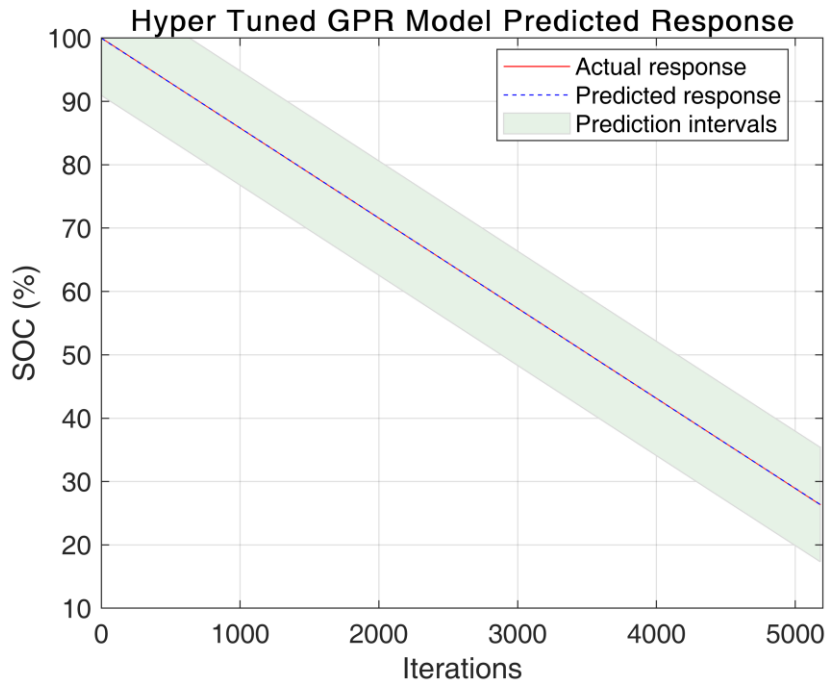


Figure 4.35: Hyper tuned GPR Kernel response for Ni-MH battery dataset

Furthermore, hyper-tuning all of the kernels yielded nearly identical results with minor variations. Although the estimated errors for Ni-MH were very small in comparison to the other two battery chemistries, the estimation vectors showed continuous variation. Because the model calculates the best fitting lines estimated using a cubic polynomial equation during the optimization stage, minor changes are frequently ignored in order to keep the overall variance to a minimum score. This was also evident in Table 4.7, where the MAE scores for Ni-MH were compared to other battery chemistries and found to be the lowest. A closer look at residual plots revealed that the RBF kernel performed slightly better for Ni-MH battery datasets. The results are illustrated in Figure 4.36 and Figure 4.37 respectively.

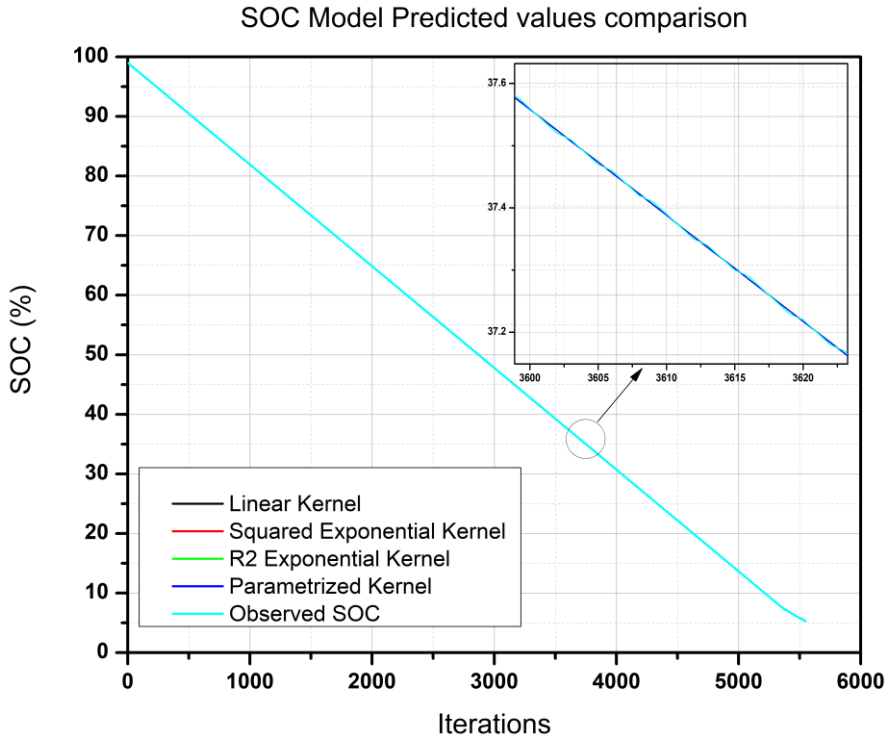



Figure 4.36: GPR model hyper-tuning kernel comparison for Ni-MH batteries

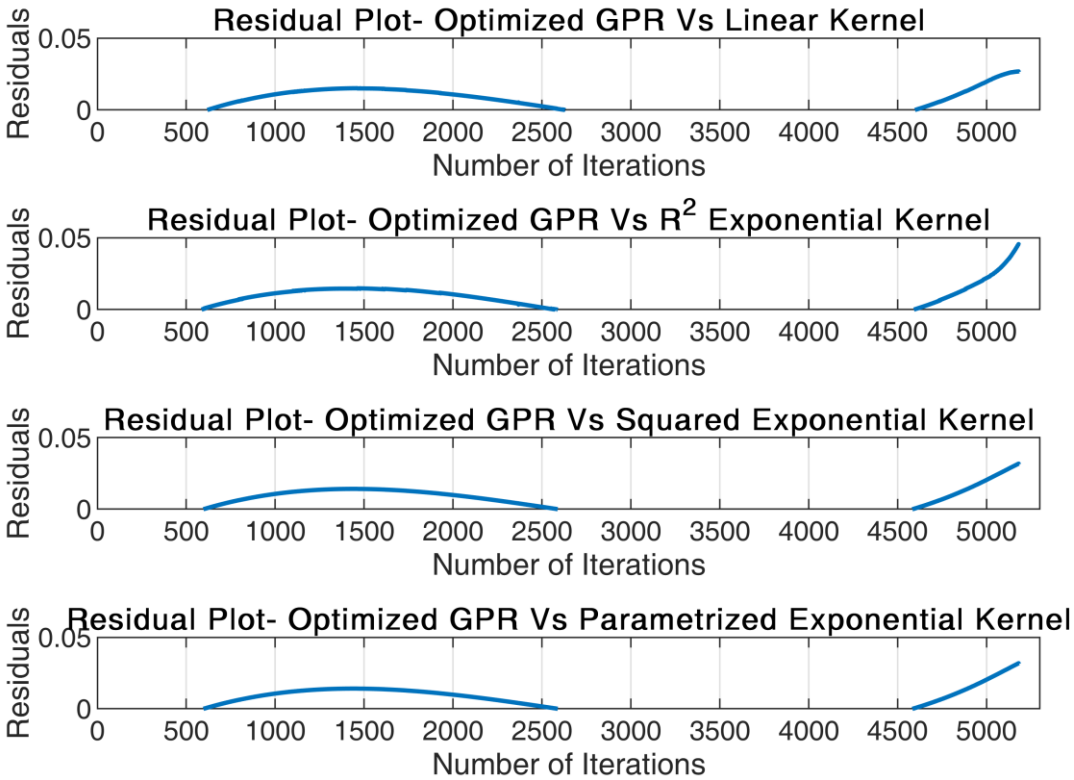



Figure 4.37: Residual plots comparing Kernel optimization for Ni-MH battery dataset

### 4.4.6 GPR Model Performance Summary

The GPR model provides a very high degree of SoC estimation accuracy for all tuned kernels with an average error rate of less than 2% in each scenario. It was also

observed that the proposed GPR model with (battery voltage and capacity) as predictors significantly improved the estimation accuracy, and adaptability on various operating conditions, as well as battery chemistries. The proposed scheme maps the intrinsic battery behavior at range of operating conditions (including drive cycles and temperatures). Furthermore, a single model can be extended to various battery chemistries with competitive performance, robustness, and scalability. The study also demonstrated that data normalization, partitioning, cross-validation can have a significant impact on model performance. The correlation values that exhibit the relationship between features can have a huge impact on model learning ability and may affect the model prediction. Finally, the results from the extensive validation tests illustrates that the trained GPR model with RBF kernel achieved the best performance metrics, with very low residuals.

The residual plots represent the difference between the model's predicted and actual SoC. If the residual has a normal distribution with a zero mean, the model is good, and the SOC estimation errors are entirely due to random noise. If the residual does not have a normal distribution with a zero mean, estimation errors are caused not only by random noises but also by unmodeled parts of the complex system. The residual values are positive and follow the zero mean normal distribution, confirming the model's comprehensiveness, predictability, and accuracy in SoC estimation. Although the results from all kernels were marginally close, the RBF kernel with hyper-tuned parameters performed the best across all battery drive cycles. Therefore, the GPR-RBF model was further optimized and converted for target ARM Cortex processors using MATLAB code generation for on-device online SoC estimation.

## 4.5 Summary

Four different techniques were investigated in this regard in order to determine the most flexible, scalable, adaptive, and predictable ML technique for battery SOC estimation. The preliminary tests on linear, nonlinear, and polynomial regression assisted in determining the order and equation of the battery behavior, resulting in hyper-tuning for each model. The number of neurons and hidden layers were then optimized for each battery chemistry to create ANN FFNN. The simplest of all the techniques investigated was found to be FFNN; however, these models are highly sensitive to feature set and may not adapt to real-time data. Furthermore, each battery chemistry required a separate model, which increased the memory and computational footprint, making them less suitable for embedded platforms.

In addition, SVM with hyper-tuned kernels for SOC estimation were investigated. For the recorded battery dataset, SVM proved to be the most complex and sensitive technique. The models were discovered to be extremely sensitive to outliers and struggled to maintain accuracy in flatter regions. Furthermore, no single model is adaptable to all battery chemistries and operational conditions, limiting its deployment on embedded platforms. Finally, a GPR model based on RBF kernels that was adaptive and least sensitive to battery data was proposed. For all battery chemistries and operating conditions, the GPR-based model performed admirably. GPR kernels can be programmed for almost all available battery chemistries and thus was chosen for further investigation in this research because battery discharge curves almost always follow a distinct and repetitive behavior.

# CHAPTER 5

# MODEL DEPLOYMENT AND ON-LINE SOC CALCULATIONS

The lightweight model generation for embedded platforms is discussed in this chapter. In this regard, lightweight MATLAB models for M4-Cortex platforms were developed. The proposed scheme used on-device real-time SOC to compute the actual voltage and thus the battery capacity. Later, this was compared to the simulated energy profiles generated by the Cooja simulator, where the native Contiki OS energy profile, KiBAM, and T-KiBAM techniques were compared to the proposed scheme.

## 5.1 The Contiki Operating System

Contiki is a memory-efficient OS for networked embedded systems and WSNs that is highly portable and multi-tasking. It runs on Atmel AVR, 8051, ARM Cortex and MSP430-based microcontroller architectures. Contiki includes a lightweight IP implementation that supports both IPv4 and IPv6 stacks. Additionally, with 6LoWPAN support, it is considered a favorable choice for IEEE 802.15.4 packets transmission using header compression. Lists some of the key architectural features and communication advantages of Contiki OS. Among all of the components and roles described in Table 5.1, small OS size (less than 30KB), ability to implement multiple architectures and platforms, native C code, ability to simulate platforms by emulating the firmware, low-power management ability, and complete TCP/IP stack support facilitating the implementation of various WSN and IoT network topologies were some of the most important features considered when Contiki OS was chosen.

Table 5.1: Contiki OS salient architectural features

| Components | Roles and Advantages |
|---|---|
| Memory and process management | Memory block allocation in Contiki OS is comparable to native C memory management functions. It introduced the concept of 'protothreads,' which reduce multi-threading overhead and allow for more efficient process management. |
| File system management | WSN and IoT devices have relatiely small memory for storage. Contiki OS provides Coffee filesystem that handles flash memory for external storage operations. |
| Dynamic modules | Contiki OS supports dynamic module loading for both the programming as well as for file storage using its coffee system. |
| Simulator/emulator | The Cooja network simulator is a network stimulator included in the Contiki OS. It stimulates networks on Contiki-supporting nodes (by emulating platform native firmware). |
| Full IP Networking | Contiki supports the entire IP network stack, including standard IP protocols like UDP, TCP, and HTTP, as well as new low-power protocols like 6lowpan, RPL, and CoAP. |
| Hardware Platforms | Contiki works on a variety of micro platforms, ranging from 8051-powered systems-on-a-chip to the MSP430 and AVR, as well as a variety of ARM devices. |
| Communication components | |
| RIME Stack | Contiki offers a customised wireless networking stack called Rime in cases when bandwidth is limited, or the entire IPv6 networking stack is overkill. |
| µIP | Because of their limited resources, low power WSN devices are unable to implement TCP/IP protocols. µIP is designed for 8- and 16-bit microcontrollers and has minimal components that are needed for the TCP/IP stack. |
| IPv4, IPv6, and low-power networks | Contiki OS supports IPv4, IPv6, 6LowPAN, CoAP, RPL and other lightweight implementations for low-power lossy networks. |

## 5.2 Lightweight SOC estimation models

The scarcity of computing, storage, and energy resources is an important consideration when working with ML-based techniques for embedded platforms. In enterprise environments, ML models are typically designed and tested on computationally efficient servers that support parallel processing and GPU utilization for improved performance. Furthermore, high-speed, and high-capacity storage, as well as an abundance of network bandwidth, ensure that these models can be deployed and integrated seamlessly in enterprise environments. Another important consideration is the ability to perform real-time model optimization, which can only be accomplished in enterprise settings due to the provision of underlying infrastructure. However, most embedded platforms lack all these resources to begin with. Furthermore, WSN nodes are mostly powered by microcontrollers rather than microprocessors, which limits the efficacy of ML models due to reduced parallel instruction handling. Therefore, due to limited resources, ML models for embedded platforms are mostly designed to be computationally efficient (to follow sequential instruction sets), lightweight, and pre-configured and optimized.

In order to account for the aforementioned challenges, a lightweight GPR SOC estimation model optimized for Cortex M-4-based platforms is presented in this study. To our knowledge, no previous literature has reported the use of GPR for SOC estimation on embedded platforms. The model deployment steps are illustrated by Figure 5.1. The GPR model was optimized and hyper-tuned (as described in section 4.4.2) using MATLAB R2020b environment. After that, the MATLAB code generation toolbox was used to create a code-routine that mapped the tuned GPR-model to the intended embedded platform. Converting the parametrized models into native functions for ARM-based microcontrollers was a significant challenge. In this regard, network

architecture parameters (such as weight and bias) and tuning parameters (such as sigma and kernel parameters) were mapped to independent libraries that can be used on any ARM Cortex-based platform. Additionally, a lightweight code routine (native C code) was written that was implemented on the device as well as within the Cooja environment.

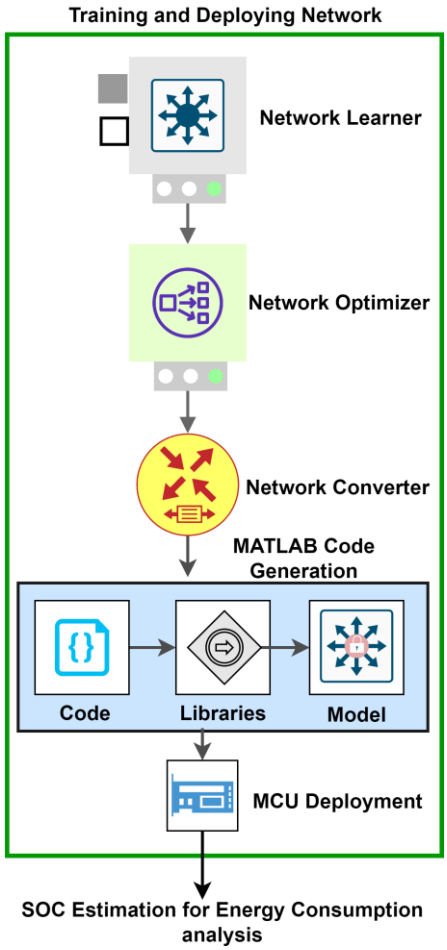



Figure 5.1: GPR Model generation flowchart for embedded platform.

### 5.2.1 On-device SOC estimation

The lifetime of WSN sensor nodes, which is limited due to stored energy restrictions, is a well-known constraint for their deployment. The key drawback is the sensor node's inadequate battery capacity, which limits its life of the equipment. In this setting, it is critical to estimate both the battery's SoC and its longevity based on the set of tasks performed by the nodes. For instance, when energy-aware algorithms must be implemented in sensor nodes, this information is quite significant. Forecasting the

battery lifetime in WSN nodes is a complex problem because various factors regulate their performance (as discussed in previous chapters).

This necessitates careful evaluation of SOC estimation strategies for WSN, which should ideally enable accurate SOC estimation without incurring significant operational and computational costs. Traditionally two major approaches are applied for battery behavior modelling, including:

i. **Hardware-based strategies** – that typically involves electronic hardware and smart ICs for battery voltage, current and other relevant parameters for SOC and capacity measurements.
ii. **Software-based strategies** – that require mathematical and analytical modelling for battery SOC estimation.

Typical hardware-based strategies include smart batteries, which utilise integrated circuits and electrochemical cells to provide battery operational data (voltage, temperature, current), as well as estimate the battery's state of charge and remaining life. However, employing hardware-based techniques increases the cost of battery production by approximately 25%. This technique may be economically unfeasible in WSNs, where a large number of nodes may be required (Y. Wang et al., 2016). Hardware-based techniques also make use of look-up tables to determine the characteristics of the cell under specified operating conditions. However, because the majority of WSN and embedded platforms are based on microcontrollers, look-up tables may introduce additional computational and storage overhead. While software-based techniques appear to be more fascinating from this vantage point, they also have several drawbacks.

As previously stated, software-based techniques are predicated on analytic battery models. Typically, these models employ a set of differential equations to approximate the battery's behaviour. Normally, these models are applied in WSN simulators to approximate the sensor nodes' operating behaviour prior to their

deployment. The current state of the battery is mathematically predicted in this context to facilitate the implementation of energy-aware techniques. This introduces inaccuracies, as discussed previously in Section 2.6.2.

Likewise, it is necessary to determine whether similar differential equation-based models can be implemented in real-world WSN nodes. Additionally, the computational impact of implementing such mathematical models on low-power embedded platforms would need to be evaluated. A relevant question in terms of computational cost is whether it is feasible to perform a battery model computation on a sensor node in order to enact an on-line SoC determination and the associated voltage level tracking functions. Therefore, the proposed scheme that eliminates the constraints of additional hardware, look-up tables, and complex mathematical operations is implemented on the hardware to perform SOC estimation. Additionally, the proposed technique is carefully assessed with several other estimation techniques (both hardware and software-based) under various topologies as illustrated by Figure 5.2. The following sub-sections first briefly introduces the energy estimation calculations performed by these methods. Next, the environment simulation and topology scenarios are discussed.

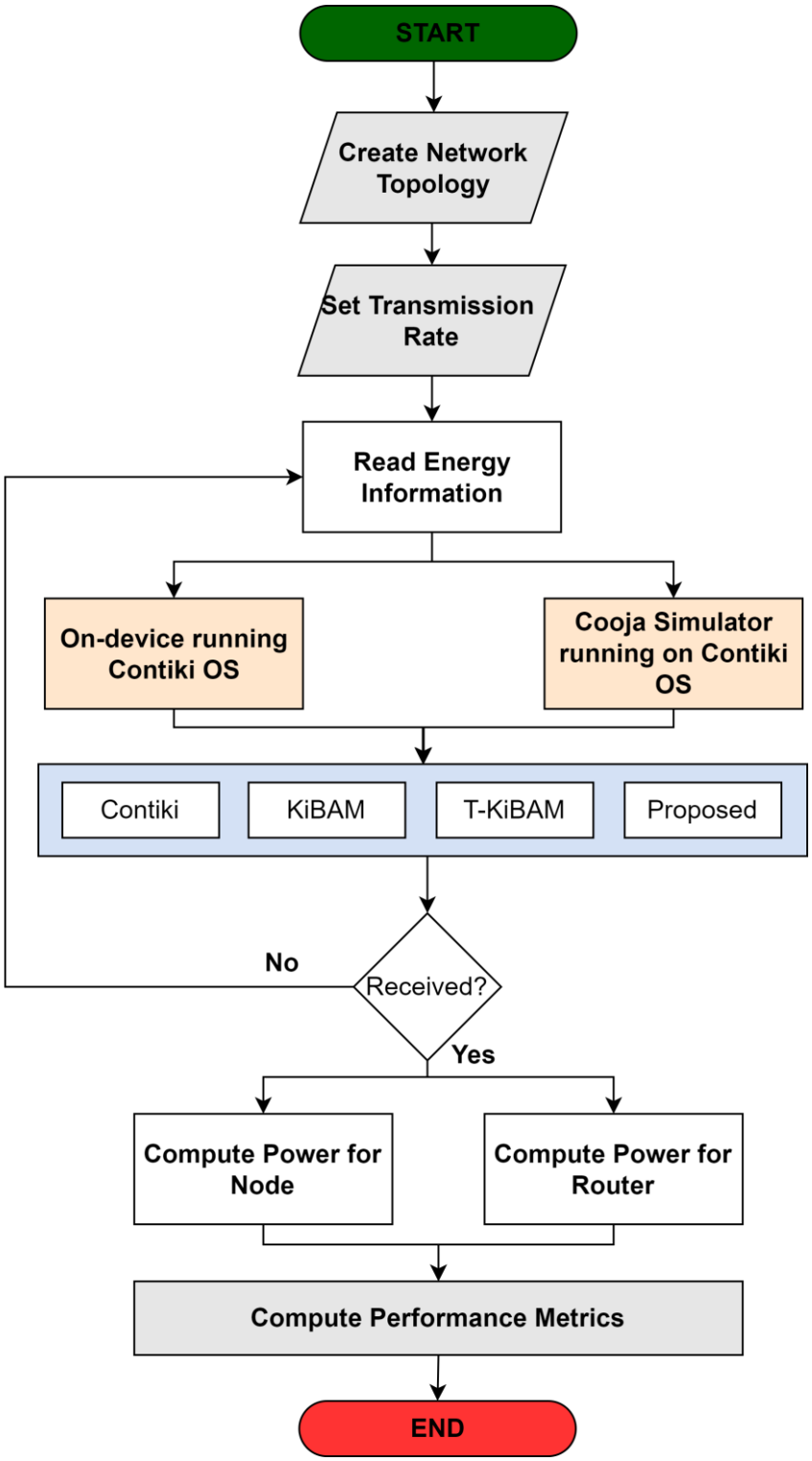



Figure 5.2: A flowchart illustrating the SOC estimation and comparison of various strategies

### 5.2.2 Contiki OS energy calculation

Contiki OS is a small, lightweight operating system optimized for wireless sensor network (WSN) applications. It enables reliable and energy-efficient radio transmission. The Contiki OS implements a scaled-down version of the TCP/IP stack in order to provide communication flexibility without adding complexity and implementation overheads associated with the full stack. The ContikiMAC protocol achieves energy efficiency through duty cycling ongoing communication, with the participating nodes typically programmed in low-power listening states. This is accomplished using the Low-power-Listening (LPL) mechanism, in which each node checks the channel for communication activity on a periodic basis. ContikiMAC enables effective radio management during these LPL and periodic wakeup stages by estimating the communication over the channel using pairs of clear channel assessment (CCA) symbols and a received signal strength indicator (RSSI). Given that radio transceivers consume the most power, it is the responsibility of the ContikiMAC operating system to keep the radios in sleep states to conserve energy while maintaining accurate and stable packet delivery rates (PDR).

Contiki OS uses the platform's on-board clock to timestamp the durations of these states in order to estimate the amount of time spent in each transition. Since vendors provide the radio power consumption figures for each state (which are also verified in laboratory experiments), the Contiki OS calculates the power consumption in each state by examining the battery voltage and current consumed. This is then carried out for each radio state, iteration by iteration (that are governed by a specific radio duty cycle). The Contiki OS uses this linear power consumption calculation (as defined by Equation (33) to estimate the power consumed during each iteration, which is then subtracted from previous cycles to determine the currently available energy.

$$P_{Total} = P_{CPU} + P_{LPM} + P_{RX} + P_{TX} \quad (33)$$

The remaining device lifetime is then estimated using the current available energy, the attached device load, and the duty cycle. Where, $P_{CPU}, P_{LPM}, P_{RX}, P_{TX}$ gives the power consumption in CPU mode, Low-power mode (sleep), receive, and transmit modes, respectively. The sum of the average power consumption in each individual state is the average power consumption of each node (P). The power consumption calculations are based on the battery voltage (VCC), which is commonly assumed to be 3V for Lithium-based batteries. The radio duty cycle is calculated by analyzing the CPU ticks of the respective embedded platform, which are then averaged over time to calculate the node's power consumption over time.

### 5.2.3 KiBAM energy calculations

Low-power designs for battery-powered systems necessitate careful design considerations as well as performance and longevity trade-offs. Many analytical models for improving battery estimation accuracy without adding computational overhead have been proposed. To improve the accuracy of battery capacity estimation, non-linear adaptive analytical models were proposed. Typically, these models rely on estimating energy consumption in various states, as well as consumption rates. Transitions between these states are also modelled in order to estimate overall energy consumption. However, because batteries are highly non-linear devices that are inherently dependent on rate of discharge and external environment effects such as temperature and humidity, such state-based transitions may not fully represent battery behaviour. Furthermore, the battery-recovery effect is critical in WSN applications, where nodes spend a lot of time in sleep states, allowing them to recover energy and potentially extend battery life. Because state variables cannot directly model these non-linear behaviours, a model

abstraction that focuses on important battery parameters (as defined by their battery chemistries) and their effects was required (dependencies).

In this regard a first order differential equations-based diffusion model, called Kinetic Battery Model (KiBAM) was proposed. As the name suggests, this model is based on battery kinetics (internal chemistry) or chemical processes. An arbitrary battery a certain point in time is represented by two charge wells, where the amount of charge flow regulates the battery capacity and amount of energy consumed. The two-charge-well illustration is presented in Figure 5.3.

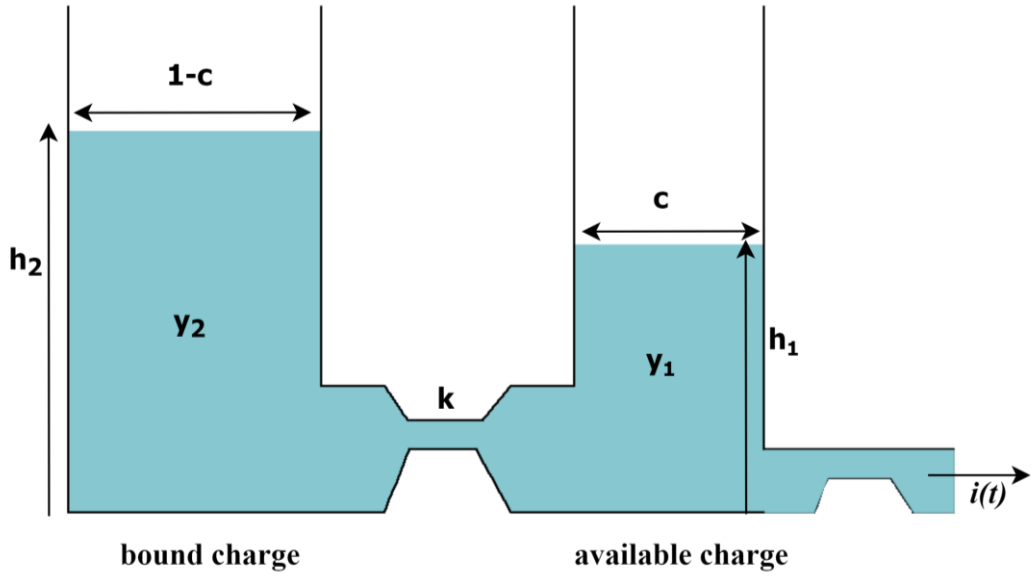



Figure 5.3: A two-charge-well model of the Kinetic Battery Model (KiBAM)

In the available charge well, a fraction c of the total battery capacity is made available, which is then directly transferred to the load (*i(t)*). A fixed conductance "k" that is defined by internal battery chemical reactions controls the amount of charge transferred between the wells. Another factor to consider is that the charge will continue to flow between the wells if there is a height or charge difference between them. The available charge well also keeps supplying charge to the load until it is completely depleted (SOC > 0%). This change or difference in charge-wells was modelled by (Manwell & McGowan, 1993) using differential equations as given by Equation (34),

$$\frac{dy_1}{dt} = -i(t) + k(h_2 - h_1),$$
$$\frac{dy_2}{dt} = -k(h_2 - h_1), \tag{34}$$

For initial conditions, where there is no-load attached, no charge flows from available-charge well and battery is considered empty. Assuming a total battery capacity of C, the initial conditions with net zero charge transfer can be given as $y1(0) = c.C$ and $y2(0) = (1-c).C$. However, with the attached load, the charge in available-well reduces, thus the height difference initiates a charge flow between two wells. This also suggests that during idle periods, no charge flow and the battery lasts longer, which can be used for battery recovery effect. Whereas, with a continuous discharge current of $i(t) = I$, the above equations can be simplified using Laplace transforms, resulting in battery discharge estimation equations, as given by Equation (35).

$$y_1(t) = y_{(1,0)}e^{-k't} + \frac{(y_0k'c - I)\left(1 - e^{-k't}\right)}{k'} - \frac{Ic\left(k't - 1 + e^{-k't}\right)}{k'}$$
$$y_2(t) = y_{(2,0)}e^{-k't} + y_0(1-c)\left(1 - e^{-k't}\right) - \frac{I(1-c)\left(k't - 1 + e^{-k't}\right)}{k'} \tag{35}$$

The KiBAM model could be implemented to track battery voltages for both idle and active states. The battery charge information together with electrical equivalent models are then used to calculate the internal battery voltage at a certain point in time $(t_0)$.

### 5.2.4 Temperature-dependent Kinetic Battery Voltage (T-KiBAM) Model

The KiBAM model, which was discussed in the previous section, can model a variety of battery transition states, including active and idle. It could also be used to investigate different battery chemistries by looking up internal charge transfer

coefficients on the manufacturer's datasheet. This model, however, was unable to predict the effect of temperature on battery discharge. Temperature, as previously mentioned, can affect the battery's charge capacity. At higher temperatures, batteries typically have higher effective capacities (M. Chen & Rincon-Mora, 2006), while at lower temperatures, they have lower effective capacities (Jaguemont, Boulon, Venet, Dubé, & Sari, 2015). The T-KiBAM model, which incorporates temperature effects on battery capacities, was introduced as an extension to the KiBAM model in this regard.

T-KiBAM investigated the thermal effect of batteries caused by internal chemical reactions, as well as their impact on the voltage and effective capacity of the batteries. A three-step process was proposed by (Rodrigues et al., 2017) to extend the KiBAM model in this regard. To begin, the Arrhenius equation was used to model the rate of chemical reactions as a function of temperature. Then, to account for the differences in chemical reaction rates between different battery chemistries, a correlation factor (CF) was calculated. The CF makes the T-KiBAM model adaptive and responsive to the most common battery types on the market. Finally, the missing battery parameters are needed to extract information on capacity, voltage, and SOC. Two techniques can be used to extract battery internal parameters in this case. On the one hand, parameter extraction experiments can be designed using battery discharge experiments at a controlled temperature and constant current. The Temblay-Dessaint voltage model (Tremblay, Dessaint, & Dekkiche, 2007; Tremblay & Dessaint, 2009), was used to create accurate battery discharge curves for various battery chemistries. This resulted in a differential equation to estimate the battery internal voltage at various temperature values, as given by Equation (36).

$$V_b = E_0 - R_b.i - K_b.\frac{Q}{Q - it.\tau_b}.(it.\tau_b + i^*) + Exp(t) \qquad (36)$$

Where $V_b$ is the battery voltage (V), $E_0$ is the constant battery reference voltage (V), $R_b$ is the battery internal resistance (Ω), $K_b$ is the polarization resistance (Ω), Q is the battery capacity (Ah), $i(t) = \int idt$ is the battery charge (Ah), $i$ is the battery current (A), and $i^*$ is the filtered current (A).

### 5.2.5 Limitations of Analytical battery models

The analytical models are highly dependent on the simulation environments and mathematical representation of internal battery parameters. These intrinsic coefficients can affect the overall accuracy of the analytical models. KiBAM model require suitable internal battery parameters and simulation configurations to model the behavior. Furthermore, the model is highly sensitive to frequency changes from one state to the other. A sudden change in frequency can result in a dramatic shift in battery's lifetime estimation. Another limitation for KiBAM model is the inability to model complex internal behaviors, such as hysteresis in the case of Nickel-based batteries. Therefore, KiBAM is more suited towards alkaline and lead-based batteries as compared to Nickel and Lithium-based batteries. T-KiBAM requires detailed information from battery manufacturers. Battery discharge rates and temperature-based calculations are quite complex and require extensive laboratory experiments for battery discharge profiling. This itself questions the efficacy of analytical models especially when detailed laboratory experiments are performed. Finally, the number of parameters and their interdependency may result in a complex algorithm, adding overhead to resource-constrained microcontrollers. However, despite their limitations, KiBAM and T-KiBAM accurately maps the behavior of various battery chemistries and are therefore chosen to compare against default Contiki OS energy estimation, as well as with the proposed estimation scheme.

### 5.2.6 GPR based energy calculation

The proposed energy calculations using Contiki OS is illustrated by Figure 5.4.

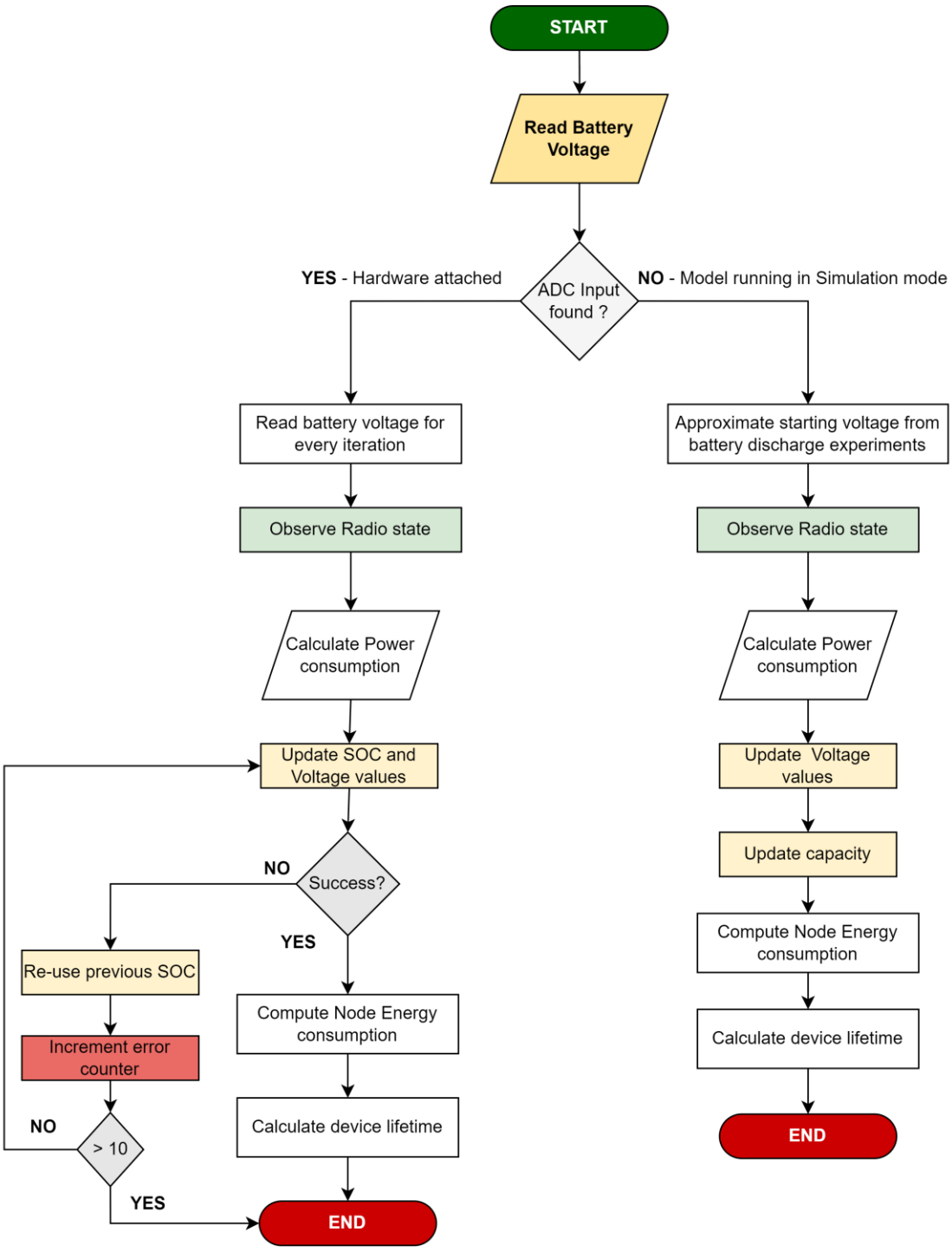



Figure 5.4: Proposed GPR-based scheme for device lifetime calculation.

The proposed scheme utilizes the extracted GPR model and native C code that was generated for M4 Cortex platform. This model code was then directly implemented

on the hardware as well as attached as a library file to emulate Sky mote in Cooja simulator. The model initially checks for the battery voltage values received on ADC specified pin. In case of no input response, the model estimates that no real hardware is attached and therefore resumes a simulation role, where initial battery voltage parameters from discharge and relaxation tests is chosen for the desired battery type. The model then iterates and records/estimates the voltage values which is then used for power consumption estimation for each cycle. The respective voltage and SOC values are then updated and are used to update the battery capacity values. As previously mentioned in Figure 5.2, once the required models are programmed, a network topology and simulation parameters are then defined to estimate the battery discharge behavior and its impact on node lifetime. These configuration settings are detailed in the following sections of this chapter.

## 5.3 Simulation Environment for Node's energy estimation

Contiki OS version 3.0 was used in these energy estimation experiments. Contiki, as previously stated, is an open-source operating system for constrained devices that supports multi-tasking, networking with a full TCP/IP stack, low-power routing, and MAC protocols, and can compile the code for multiple hardware architecture types such as ARM, PIC, MSP430, and the ability to port on 8051-based systems. Initially, a simulated network response (based on all four strategies) was observed using the Cooja simulator. The effect of packet rates on radio duty cycle and power consumption was later calculated. These calculations were carried out with the help of Contiki OS's default energy estimation, KiBAM and T-KiBAM models (attached as libraries to emulated nodes), and the proposed GPR-based SOC estimation (as added library for energy estimation).

In these simulations, Cooja's CollectView and Energest modules were used to examine various simulation parameters for node power consumption. Finally, because only T-KiBAM and the proposed scheme can monitor the effects of temperature on battery life, real-world on-device experiments were carried out on CC1352R2 platforms. The accuracy of both techniques was then compared using these hardware-based calculations.

### 5.3.1 Node requirements

Texas Instrument CC1352R2 platform was used in this study which is designed on Texas Instruments TI MSP430 platform. The on-board crystal oscillator on MSP430 based sky mote runs at 32 KHz frequency, which can be programmed in the increasing power of 2 ($2^0$, $2^1$, $2^2$, $2^3$, $2^4$, $2^5$) (Nasseri et al., 2015). This platform uses TI CC2652 low-power chip as the transceiver that operates at 2.4 GHz. The MSP430 microcontroller controls the CC2652 transceiver through a Serial Peripheral Interface (SPI) (Instruments, 2007). Based on the Contiki OS driver for T-mote sky platform, the transceiver chip can be programmed to adjust for CCA as well as RSSI, where:

- CC2652 records RSSI thresholds for a minimum of 8 symbols period (128 μs) (Uwase, Bezunartea, Tiberghien, Dricot, & Steenhaut, 2017)
- CCA utilizes the average RSSI value and validates it with the threshold, which is programmable during compile time.

The Energest module uses CPU ticks to convert the CPU seconds into the resolution of the platform which is usually in milliseconds. The programmable Energest software-module estimates the total power available on a node which is based on the following:

- CPU Time (with radios turned off)
- Low-Power-Mode LPM (node sleep mode)

- Transmit Tx and Receive Rx mode

The overall power estimation for a node at a given time, is calculated based on the energy available on a node as given in Equation (33). However, it is important to consider the energy profile values and the initial battery voltage calculations that regulates the accuracy of these simulations. Although, the nodes consume the same amount of current (as described in Table 3.3), the initial battery voltage estimation may change the node power consumption. In this regard, Table 5.2 presents the energy profiles of the nodes (based on Lithium batteries at room temperature), for various energy estimation models.

Table 5.2: CC1352R2 platform-based energy profiles (Lithium-based batteries) for each model.

| **Parameters** | **Node State** | **Power consumption values** | | | |
|---|---|---|---|---|---|
| | | Contiki | KiBAM | T-KiBAM | GPR |
| VCC | Node Supply Voltage | 3V | varies per vendor datasheet. Average value of **3.51V** observed in laboratory | varies per vendor datasheet. Average value of **3.51V** observed in laboratory | Average value of **3.51V** was chosen as observed in laboratory |
| $P_{CPU}$ | MCU on, Radio off | 0.33mW | 0.386mW | 0.386mW | 0.386mW |
| $P_{LPM}$ | MCU in sleep, radio off | 0.15mW | 0.175mW | 0.175mW | 0.175mW |
| $P_{TX}$ | MCU on, Tx mode | 24.3mW | 28.43mW | 28.43mW | 28.43mW |
| $P_{RX}$ | MCU on, Rx mode | 47.4mW | 55.46mW | 55.46mW | 55.46mW |

The algorithm determines each node's current state based on the difference in energy levels from its previous corresponding state. The Energest code routine is then

used throughout the simulations to obtain real-time node power consumption statistics. These system parameters are then compared to each subsequent recorded node value using the sequence timer, and the current residual node energy is calculated using the algorithm described below.

**Algorithm 1:** Algorithm to compute real-time node energy

```
Result: Node Residual Energy Calculation
initialization;
while energest_init(); do
    prev_cpu = energest_type_time(ENERGEST_TYPE_CPU);
    prev_lpm = energest_type_time(ENERGEST_TYPE_LPM);
    prev_transmit =
     energest_type_time(ENERGEST_TYPE_TRANSMIT);
    prev_listen = energest_type_time(ENERGEST_TYPE_LISTEN);
    if sequence_no ≥ prev_sequence then
        e_cpu = all_cpu - prev_cpu;
        e_lpm = all_lpm - prev_lpm;
        e_transmit = all_transmit - prev_transmit;
        e_listen = all_listen - prev_listen;
    else
        energest_flush();
    end
end
```

Figure 5.5: Contiki OS Energest Module algorithm for energy estimation.

### 5.3.2 Network Requirements

The network simulations in Cooja utilized 20 nodes simulating a simple yet fully converged network scenario, where nodes 1-18 were configured to be *sender* nodes, node 19 as *aggregator* node, and finally node 20 as the *sink* or *router* node. This network topology helped to configure required network stacks on every node consistently, thus, making it easier for node's energy consumption. Whereas, for real-time on-device energy estimations, a simple network topology emulating the roles of source, aggregator, and sink nodes is maintained by three independent CC1352R2 MCU devices. As evident, aggregator and sink nodes require a faster duty cycle to collect and process all the incoming data; the network simulations were carried out at several

network packet rates to observe the effect of energy consumption at node level. The network topology for energy estimation simulations is presented in Figure 5.6.

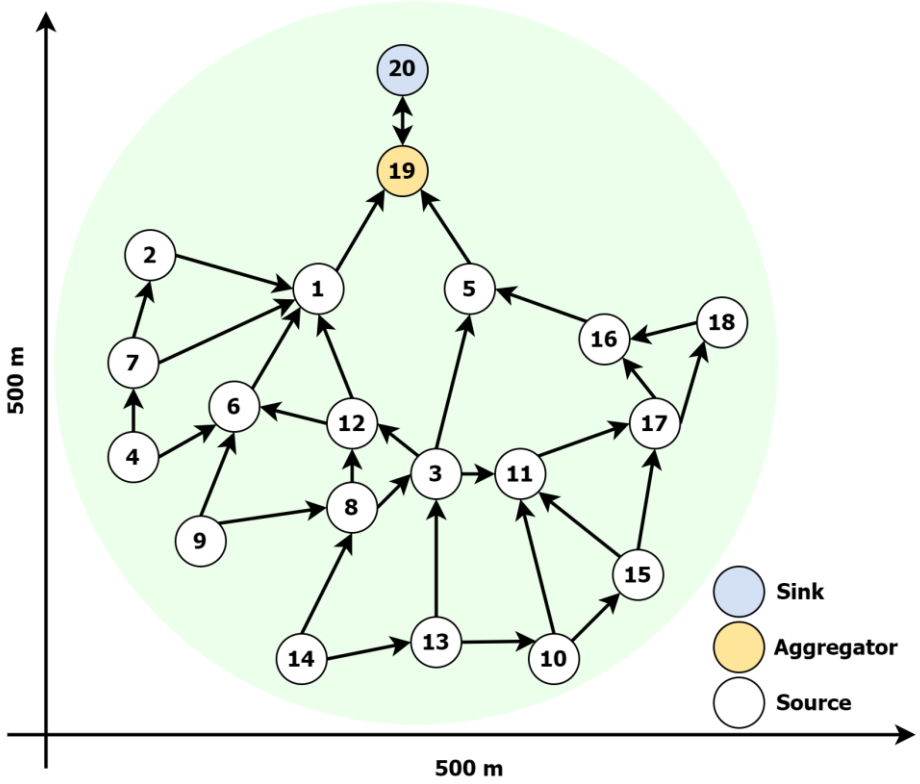



Figure 5.6: Network topology for CC1352R nodes in Cooja Simulator.

In Contiki OS implementation, the nodes frequently listen to the channel for the incoming data packets, which is controlled by Channel Check Rate (CCR), with a default value of 8 Hz, but can be extended in increasing exponential power of 2 ($2^0$, $2^1$, $2^2$, … $2^5$ Hz). The effect of CCR is directly related to the network's PDR (accuracy) and duty cycle (energy consumption). Nodes with lower CCR values may check for activity on the channel less frequently, resulting in interference and re-transmission. Although it may appear that a lower CCR equates to lower energy consumption, the need for re-transmission can affect both the PDR and energy consumption. Whereas, in most scenarios, a high CCR may not be required, resulting in an increase in radio duty cycle and overall network overhead (power consumption). Therefore, estimating the effect of CCR on PDR for the proposed network topology was critical.

### 5.3.2(a) Estimating effect of CCR on PDR values

ContikiMAC's behaviour is investigated for variable packet rates (low, high, and non-uniform) with varying CCRs. The channel suffers from a lot of radio interference as well as timing issues in a multi-hop mesh network topology with non-uniform packet rate. Therefore, a mesh network topology based on 20 network nodes is established, with each node having a maximum transmission range of 125 meters, as per the manufacturer data sheet. The radio duty cycling (RDC) layer of the ContikiMAC protocol is configured for multiple CCR ranging from 4 Hz to 32 Hz. For all three scenarios (low, high, and non-uniform), network performance is measured in terms of the packet delivery ratio (PDR) for various packet rates. As previously stated, the network is fully meshed, meaning that nodes relay packets to the sink node, resulting in a fully converged topology. The power consumption estimates are made on each node, while the sink node provides the successful packet delivery ratio (PDR). Table 5.3 shows the network, CCR, and packet rate configurations.

Table 5.3: Network configurations for energy estimation simulations using Cooja simulator.

| **Parameters** | **Configurations** |
|---|---|
| WSN Platform | CC1352R2 |
| Coverage Area | 500 $m^2$ |
| Node Range | 50 m |
| Protocol | ContikiMAC with RDC configured |
| Channel Check Rate | **Low CCR** (4 Hz)<br>**High CCR** (32 Hz)<br>**Non-uniform CCR**<br>Nodes 1-6 – 4 Hz<br>Nodes 7-11 – 8 Hz<br>Nodes 12-16 – 16 Hz<br>Nodes 17-19 – 32 Hz |
| Packet Rates | (1/10, 1/5, 1, 5, 10) packets per second |
| Source Nodes | Nodes 1 – 19 |
| Sink Nodes | Node 20 |

It is noteworthy to mention that ContikiMAC protocol assigns a default CCR value of 4HZ to every node. In these simulations the effect of network traffic (packet

rates) was investigated, therefore various CCR rates were configured to observe the network traffic patterns. In this regard, five different packet rates were chosen to simulate the network traffic and evaluate its impact on CCR values and the duty cycle. The trends for all three scenarios are divided into three categories: low packet rates, medium packet rates, and high packet rates. In Low-CCR based simulations, it is observed that as packet rates are increased, the network struggles to maintain the desired CCA frequency, resulting in packet loss and lowering the network's overall performance. Due to their high CCA rates and ability to transfer packets, high-CCR scenarios perform better even when packet rates are increased. When the packet rate is increased close to the packet interval boundary, however, the PDR performance suffers significantly. Non-uniform CCR-based distribution trends follow a similar pattern, performing better when the packet rate is close to the interval period. Figure 5.7 summarizes the findings.

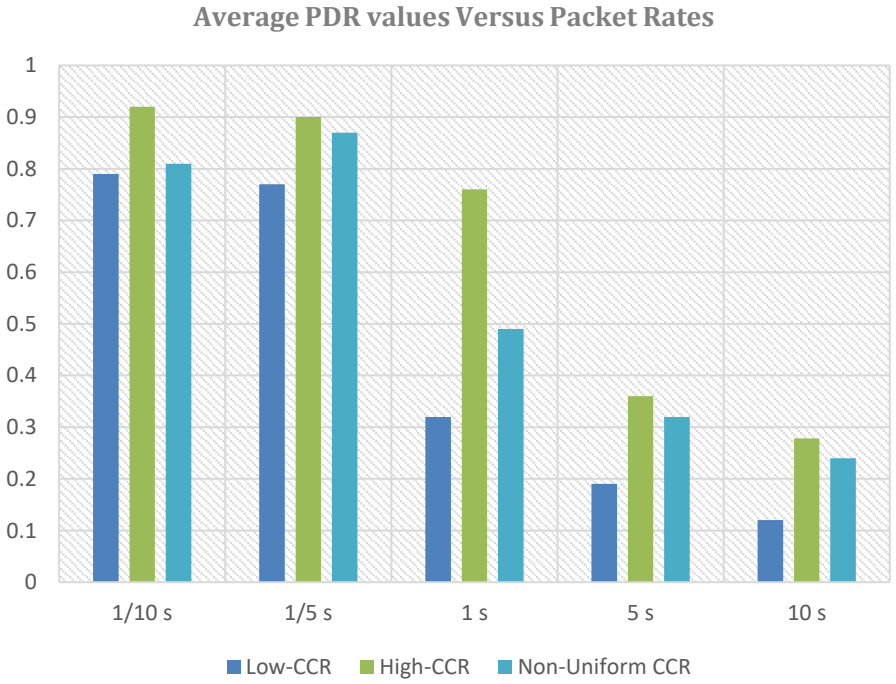



Figure 5.7: Average PDR values versus Packet rates.

The uniform CCR rate scenarios adapt well to the higher packet rate while maintaining the desired PDR rates. Low-traffic areas have higher average PDR and,

Therefore, lower CCR values. However, as network traffic grows, the average PDR suffers significantly, falling below 50% for packet rates greater than 5 packets/second. Similarly, the non-uniform CCR yields similar results, as shown in Figure 5.8.

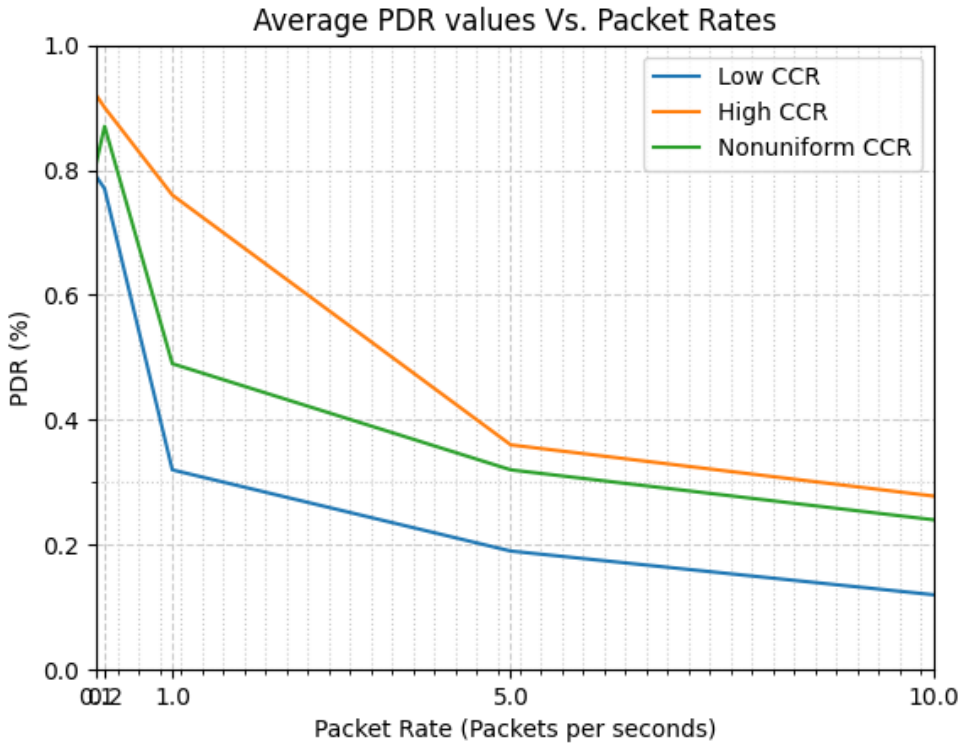



Figure 5.8: Network Performance evaluation: PDR rates with CCR patterns

The detailed examination of the trend for lower traffics, however, reveals that it is variable and unpredictable. As average PDR values struggle to maintain consistency, it's critical to comprehend the underlying behaviour in terms of duty cycle adaptation for their varying CCR rates. Figure 5.9 also depicts the performance evaluation in terms of duty cycle. As illustrated, an increase in packet rates causes the duty cycle to increase suddenly in all three CCR scenarios. With increased traffic (in terms of packet rates), network collisions and back-off intervals increase, causing the MAC protocol to increase its duty cycle significantly in order to maintain the average PDR. Increased energy consumption of these nodes in various power states accompanies the increase in duty cycle to match network performance.

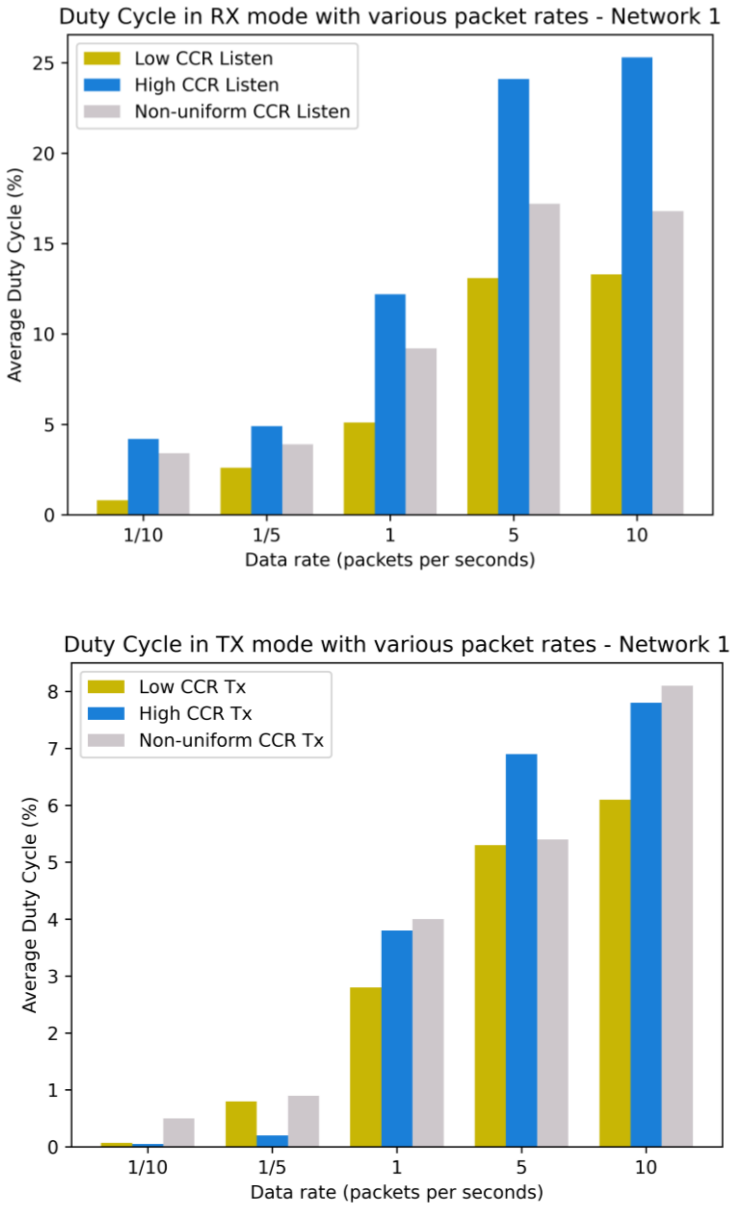



Figure 5.9: Average duty cycle of the nodes in RX and TX states.

Due to the proximity of the nodes, the nodes in Rx mode suffer greatly from increased CCR rates and thus increased interference. The variance and uncertainty in the behaviour of non-uniform CCR rate networks is another interesting finding from these experiments. Nodes with non-uniform CCR rates in both Tx and Rx modes vary their duty cycles for low and high packet rates, indicating ContikiMAC's inability to adapt in non-uniform data rate environments. It's also been observed that as network traffic grows, network throughput suffers significantly. The ContikiMAC protocol is

designed for low-traffic event-driven networks, where such rates of traffic are unlikely to be observed in the real world.

As ContikiMAC is originally designed for low-traffic networks, the next set of simulations utilized fixed (uniform) CCR of 4Hz, with four different packet rates (IDLE or 0, 1, 16, 32 packet/s) to investigate a low-traffic network scenario for energy estimation. At this stage, this step is critical because the aim of this objective was to accurately estimate the energy consumption of nodes running similar configurations, rather than to optimize the ContikiMAC protocol itself. These experiments served as the foundation for further MAC layer energy conservation optimizations, which are discussed in the following chapter by introducing the proposed Adaptive Clear Channel Assessment (A-CCA).

#### 5.3.2(b) Energy estimation in Cooja Simulator

Although the network can be programmed to have a CCR of up to 32 Hz, a medium CCR of 4 Hz was used in these experiments because these rates are close to symbol intervals and are susceptible to performance variations. Nodes 1 through 18 are source nodes, node 19 is an aggregator node, and node 20 is a sink and RPL-border node. NULL RDC is enabled on all nodes, allowing them to remain powered on throughout the simulation period without changing their duty cycle. The resource and power utilizations on each node are monitored in the case of channel activity. To complete the transaction, the successful packets are relayed to node 20, which acts as a border gateway sink. The node placement in the network topologies may vary, but the protocol configuration remained consistent throughout the investigations. Table 5.4 shows the protocol configuration for each layer.

Table 5.4: Network protocol configurations across multiple layers.

| Layer | Protocol | Configuration |
|---|---|---|
| Application | Null | • CollectView<br>• EnergyTrace<br>• Custom Energest Algorithm (implement energy estimation models) |
| Transport | UDP | Standard |
| Network | RPL/IPV6 | Standard |
| Adaptation | 6LoWPAN | Border router |
| Data Link | CSMA | Standard |
| Radio Duty Cycling | ContikiMAC | NULL_RDC |

The RPL protocol automatically forms a network topology by exchanging ICMP packets when the network is fully converged, whereas the 6LoWPAN layer favors the exchange of IPV6 packets from the source to the sink when the network is fully converged with nodes in proximity. The ContikiMAC will carry out its standard MAC layer functions by continuously monitoring the channel for activity and as previously stated, following the CCR rate. The ContikiMAC protocol then records CPU transition times between these states by flagging different events. In this case, it records the activity during the time between CCA CHECK TIME and CCA SLEEP TIME, whereas MAX NONACTIVITY PERIODS estimates the time between radio activities while idly listening to the channel. This time interval is used to calculate false WakeUPs between activity periods, later on. The CCA CHECK TIME is usually determined by the hardware used to perform CCA checks. Because the hardware clock in this example is 32 KHz, Contiki OS divides the timer by 4 (based on the distinct states) to get ticks per second, which for the MSP430 platform is 8192 ticks. The equation (37) is then used to calculate the total time (seconds) spent by a node in each state.

$$T_{node_{state}} = T_{idle-listening} + \left(T_{false-WakeUps}.T_{non-active\ period}\right) \qquad (37)$$

The timing information is then used to compute the energy dispensed in each state, which later combines as given by Equation (33) to give the overall energy consumption estimation. Together, the node power level information (voltage), transition time ($T_{node\_state}$), and total overall time ($CPU_{time}$) are used to mathematically compute the energy consumption for each iteration, as given by Equations (38)-(41), however, to record battery parameters such as Voltage and SoC, the mathematical models are used that are detailed in *Appendix D*.

$$E_{CPU} = \frac{CPU_{Time} \times VCC \times E_{CPU}}{CPU_{Time} + LPM_{Time}} \tag{38}$$

$$E_{LPM} = \frac{LPM_{Time} \times VCC \times E_{LPM}}{CPU_{Time} + LPM_{Time}} \tag{39}$$

$$E_{TX} = \frac{Tx_{Time} \times VCC \times E_{Tx}}{CPU_{Time} + LPM_{Time}} \tag{40}$$

$$E_{Rx} = \frac{Rx_{Time} \times VCC \times E_{Rx}}{CPU_{Time} + LPM_{Time}} \tag{41}$$

## 5.4 Results – Energy Estimation

A series of simulations were conducted, each using an independently configured energy estimation technique, as previously discussed. Throughout these experiments, the network topology, configurations, and simulation parameters were maintained. The simulations used a 1000 mAh battery (reported close to 960 mAh on average, in the laboratory). The reported results examined energy estimations (for both the *end-node* and *router* node) using all four techniques at a variety of packet rates and highlighted the following:

- Performance based on average power consumption

- Performance based on radio duty cycle
- Simulated Lifetime estimation
- Real-time, on-device lifetime estimation comparing T-KiBAM and the proposed scheme

### 5.4.1 Performance based on average power consumption

As illustrated by (Figure 5.6), the network can be thought of two distinct components, namely the *router* (sink node) and *end-nodes* (source). The source nodes are responsible for packet generation, whereas the sink node aggregates and process the entire communication within the network. In these simulations, although the router was not connected to external gateway device, however, in typical IoT applications, the router is connected to internet-facing gateway, thus requires additional work as compared to the other nodes in the network. Therefore, it is only logical to investigate the average energy consumption of these distinct nodes separately. Additionally, the effect of various schemes for energy calculation (as discussed earlier), is also investigated separately for these nodes.

### 5.4.2 Average energy consumption of Source Nodes

The simulations were performed (as discussed in, 5.3.2(b)) and the average node energy consumption (from fully charged battery, to a completely depleted node) were calculated for all four schemes. The Contiki OS's default energy calculation scheme (Energest) was used for benchmarking other implemented schemes. This provided a clear insight on added resource utilization due to advance mathematical calculations performed by other models. Furthermore, it also enabled to observe the average energy consumption of all schemes with varying data rates under similar network conditions and topologies. As illustrated in Figure 5.10, it is evident that Contiki OS with its Energest module observed a linear increment in energy consumption. This is because the power consumed in each CPU state is calculated after estimating the battery's initial starting voltage, which is then computed for each iteration. Therefore, the process

advances linearly, adding similar CPU states for the next set of iterations while lowering the voltage levels (residuals left from the previous iteration). Although the energy calculation in Contiki OS appears to be simple and straightforward in its default configuration, the linear relationship results in approximation errors because battery voltages are highly non-linear and thus cannot be used as a linear input. This leads to estimation errors in each iteration, resulting in a node energy estimate that is inaccurate overall.

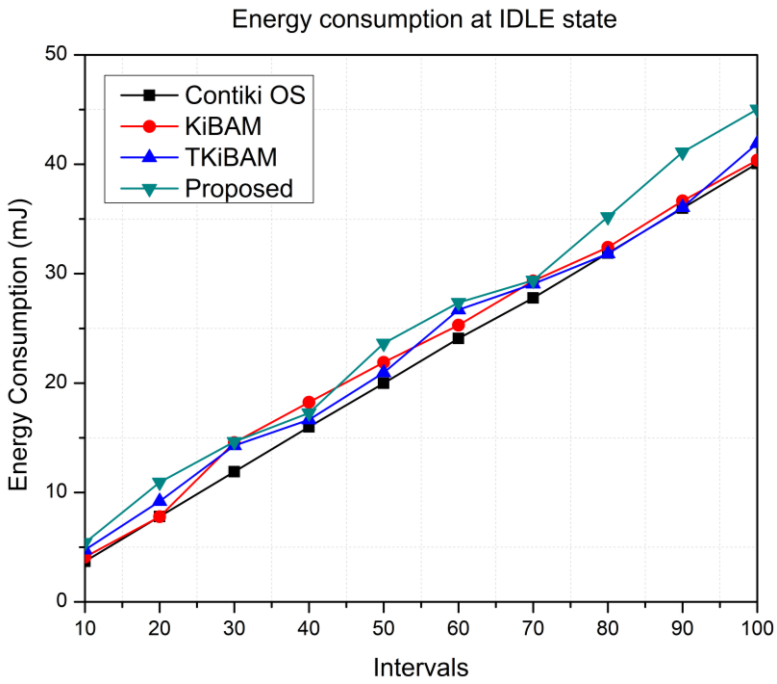



Figure 5.10: Average energy consumption for source nodes at IDLE state (0pkt/s)

The KiBAM and TKiBAM schemes provided a close match. Due to the additional computation required, the energy consumption of these schemes is slightly higher. Similarly, due to a larger memory footprint and additional computations required, the proposed scheme used slightly more energy. Another important factor is the proposed scheme's increased average power consumption when the remaining battery SOC falls below 30% (as represented by the interval 70–100). The algorithm performs additional calculations to accurately calculate the remaining energy during this region due to a sudden drop in battery voltage, incurring additional cost for energy

estimation. Finally, similar results are provided in (Figure 5.11- Figure 5.13) that illustrates the energy consumption of source nodes at higher data rates.

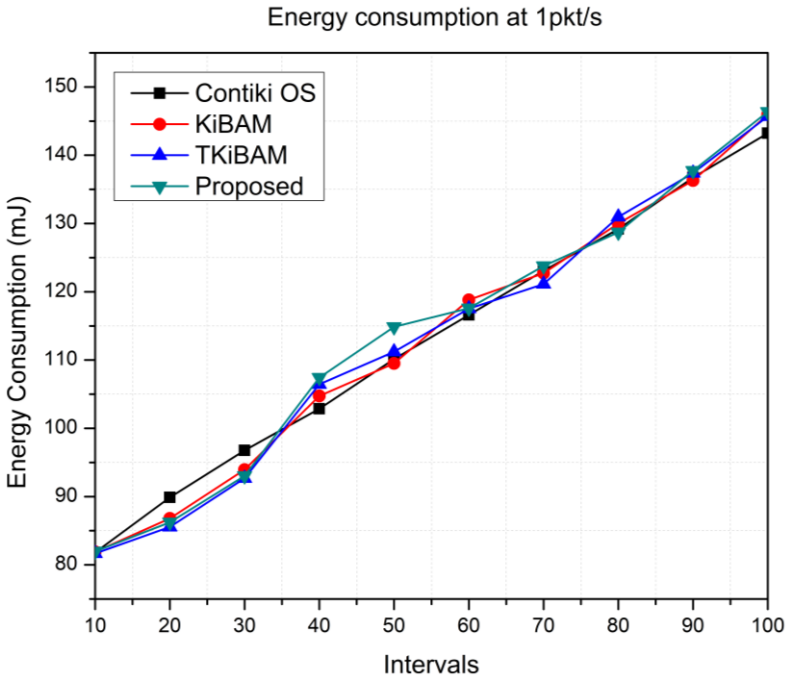



Figure 5.11: Average energy consumption for source nodes at 1 pkt/s

As shown, increasing the data rate (1 pkt/s) causes the nodes to communicate by exchanging packets. Therefore, of the increased computation cost, more energy is required. The linear increment in Contiki OS calculation is an interesting find, as other schemes rely on estimated battery voltage and power consumption in each CPU state. Therefore, estimations are more accurate as compared to the Energest module.

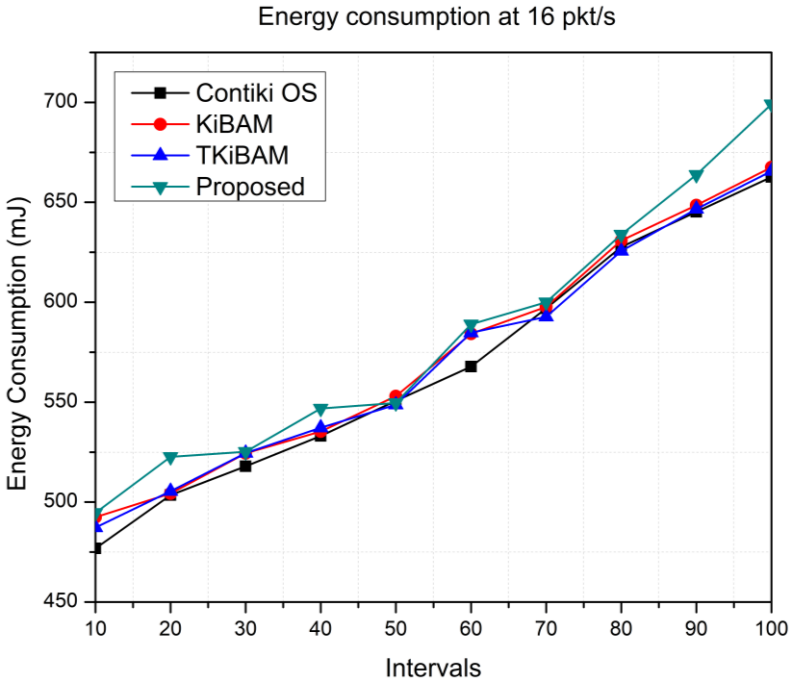



Figure 5.12: Average energy consumption for source nodes at 16 pkt/s

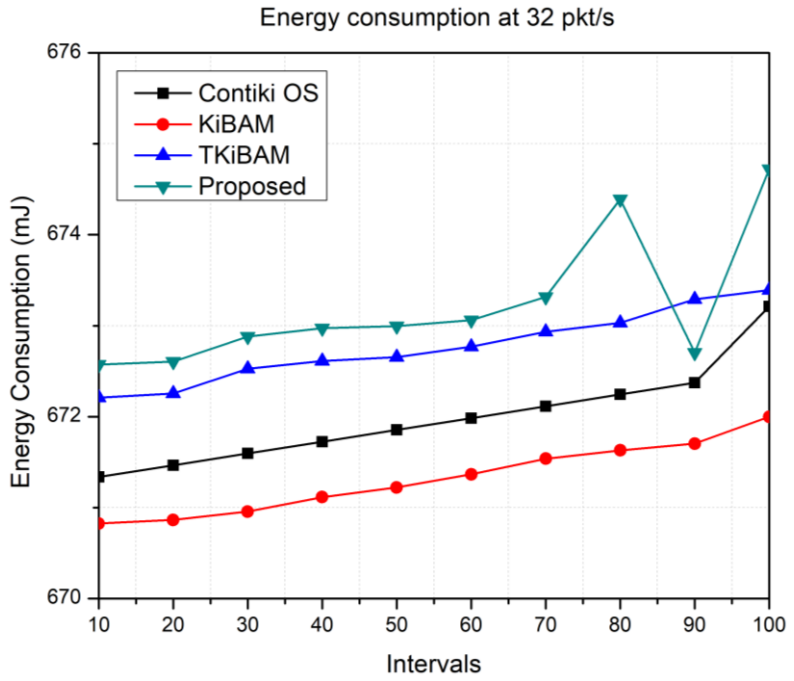



Figure 5.13: Average energy consumption for source nodes at 32 pkt/s

At higher data rates (16 and 32 pkt/s), a similar trend can be seen, with Contiki OS's Energest module maintaining a more linear-like behaviour than the other schemes. Other schemes resulted in increased computational overhead, as evidenced by increased network traffic. Additionally, two additional observations were discovered, particularly at higher data-rate network traffic (32 pkt/s):

- KiBAM scheme performed a little better than other schemes. In direct comparison to Contiki's energy consumption calculation, KiBAM incurred less cost, which may be because the initial battery transients were settled, the KiBAM module only computed the average effect in each iteration.
- Whereas the proposed scheme struggled as the network traffic increased, due to additional and complex calculations it performs. As the network traffic increases, it incurs additional overhead (both in terms of available CPU memory and computational power) where the process slows down, thus resulting in added CPU activity time. This additional CPU activity time consumes more energy and is considered as a trade-off for accuracy.

### 5.4.3 Average energy consumption of Sink Node

The average sink node consumption was calculated in a similar way, with a fully charged node being investigated using all schemes until it was completely depleted. One of the most important functions of a sink node is to aggregate and process all of the network's exchanged packets, which necessitates more CPU cycles and thus consumes more average power in the CPU activity state. Furthermore, because the nodes' hardware is similar, with no additional buffers for sink nodes, it's more likely that this has an impact on the computational process, which is reflected throughout the simulation.

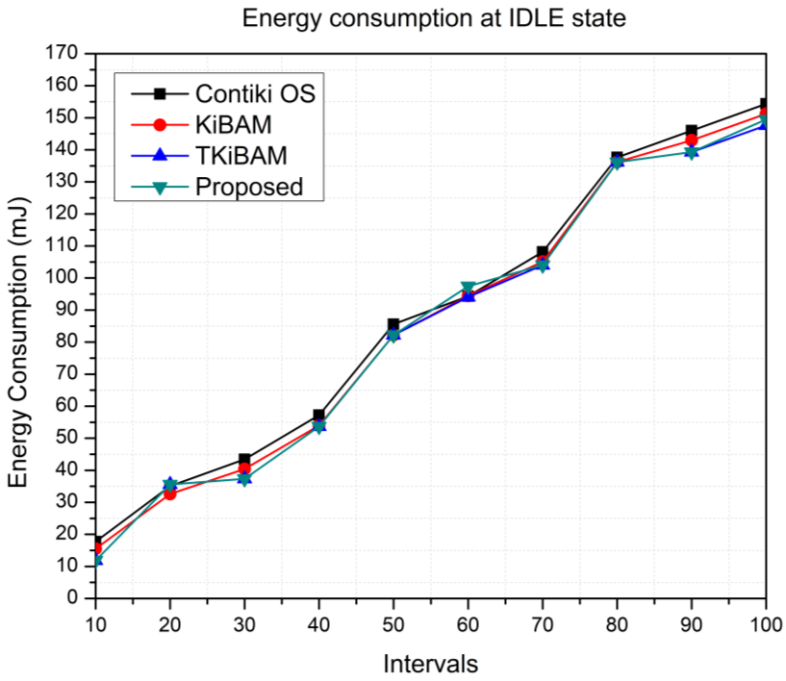


Figure 5.14: Average energy consumption for sink node at IDLE state (0 pkt/s)

It was discovered that, in contrast to the Energest module's behavior for source nodes, which was almost linear, the sink nodes had to go through a lot more CPU states, with little to no IDLE or other (TX and RX) states. This resulted in a non-linear pattern, where various CPU states added to node activity, resulting in a different energy consumption pattern, despite the voltage being chosen arbitrarily. Similarly, other schemes also followed the similar pattern when the sink nodes were in IDLE state.

However, as network traffic increased, the nodes engaged in more activities in the CPU state, resulting in increased power consumption. When network traffic increased, all three schemes consumed more energy than Contiki OS's Energest module at first. However, as network traffic increased (to 32 packets per second), a different pattern emerged. The Energest module had to calculate the energy consumption in each state iteratively, even though the linear voltage decay showed a higher value (thus resulting in more energy consumption). All other schemes relied on accurately modelled battery voltages that were iteratively used to compute node energy consumption. Furthermore, it was discovered that the proposed scheme outperformed all other

schemes, and thus can be used in networks with high data-rate aggregation nodes. These results are illustrated in (Figure 5.15- Figure 5.17).

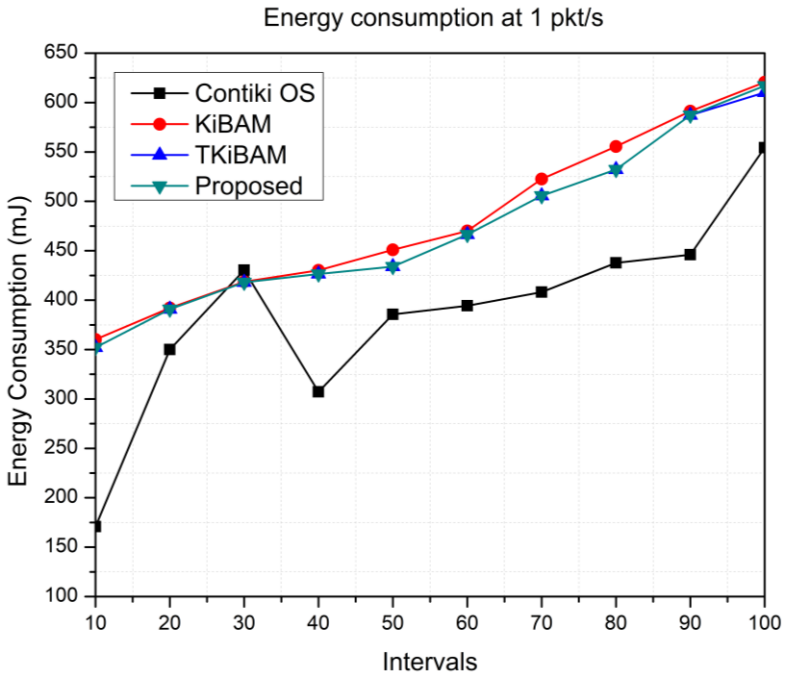



Figure 5.15: Average energy consumption for sink node at 1 pkt/s

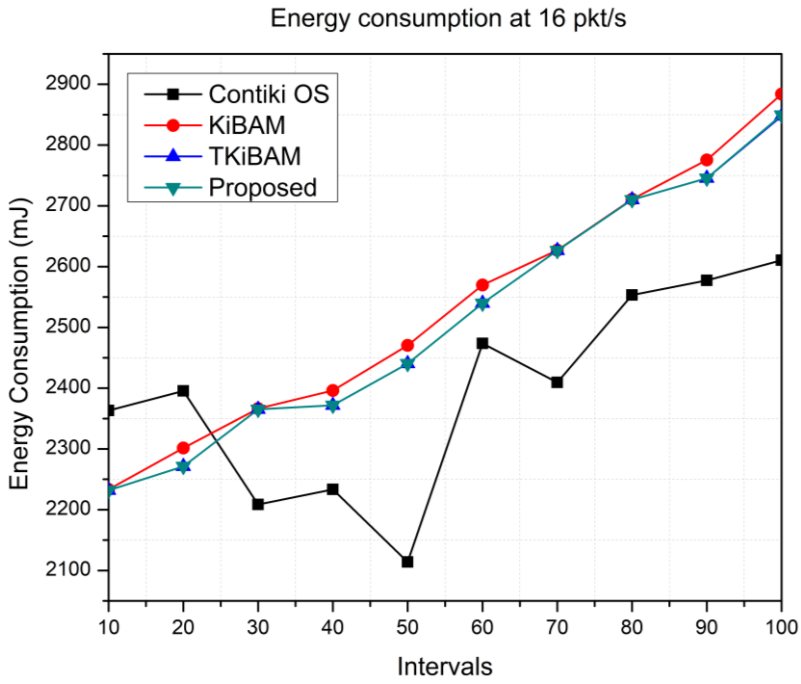


Figure 5.16: Average energy consumption for sink node at 16 pkt/s

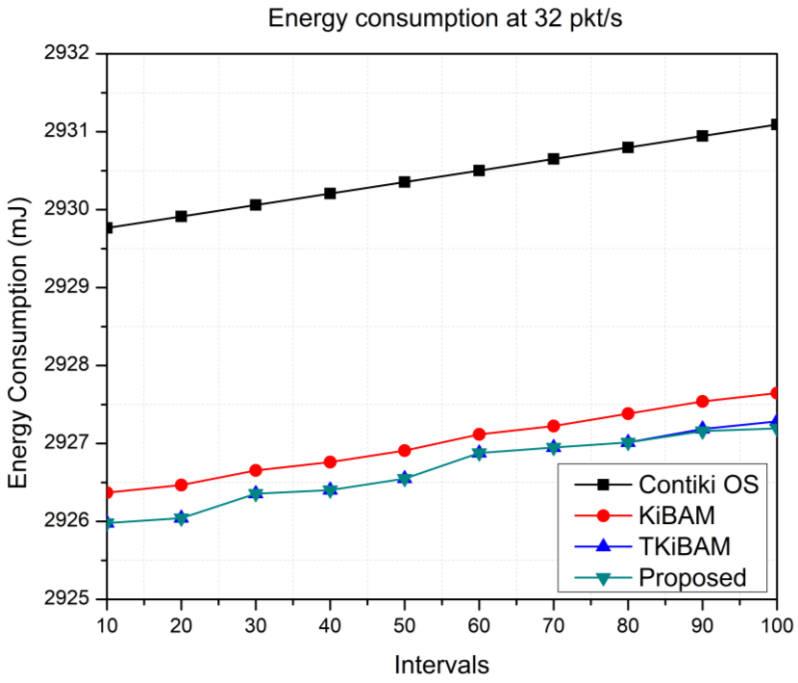



Figure 5.17: Average energy consumption for sink node at 32 pkt/s

Finally, the average change in the energy consumption for both, source and sink nodes are presented in Table 5.5, by contrasting against the average energy consumption of Contiki OS's Energest module.

Table 5.5: Average change in energy consumption for the schemes under investigation

| Observed Average change in energy consumption for Source Nodes (%) | | | |
|---|---|---|---|
| **Data Rate** | KiBAM Vs Contiki | TKiBAM Vs Contiki | Proposed Vs Contiki |
| IDLE | 5.15 ↑ | 5.55 ↑ | 14.04 ↑ |
| 1 pkt/s | 0.03 ↑ | 0.84 ↑ | 0.65 ↑ |
| 16 pkt/s | 0.99 ↑ | 0.64 ↑ | 2.50 ↑ |
| 32 pkt/s | -0.09 ↓ | 0.11 ↑ | 0.18 ↑ |
| Observed Average change in energy consumption for Sink Node (%) | | | |
| **Data Rate** | KiBAM Vs Contiki | TKiBAM Vs Contiki | Proposed Vs Contiki |
| IDLE | -2.82 ↓ | -4.28 ↓ | -3.69 ↓ |
| 1 pkt/s | 23.87 ↑ | 21.58 ↑ | 21.76 ↑ |
| 16 pkt/s | 5.83 ↑ | 5.06 ↑ | 5.06 ↑ |
| 32 pkt/s | -0.11 ↓ | -0.12 ↓ | -0.12 ↓ |

### 5.4.4 Performance based on Node Duty Cycle

Energy is conserved in typical WSN applications with duty-cycled MAC protocols by throttling duty cycles. If the overall Quality of Service (QoS) and any other

performance metrics are met, the lowest duty cycle should be used to maintain a reliable transmission between nodes (such as PDR). However, as mentioned in, the ContikiMAC protocol and its NULL RDC configurations were used in these simulations to estimate the energy consumption of the nodes under consideration. Where duty-cycles are not fixed or explicitly defined, ContikiMAC's NULL RDC configuration tries to maintain the best available PDR. In addition, the MAC protocol changes the duty cycle to maintain effective PDR rates, which is used as a diagnostics mechanism in normal WSN settings to investigate the impact of network traffic. In this regard, a closer examination of the duty cycle of the sink and source nodes revealed the overall change incurred Therefore of the proposed scheme's additional computations. It's also worth noting that Radio Duty Cycle (RDC) has a significant impact on WSN nodes, as devices tend to transmit (TX) and receive (RX) for longer periods of time, resulting in higher power consumption. However, *Appendix E* contains a detailed analysis of the respective RDC states, whereas this section focuses on the average overall change in duty cycle observed during these simulations.

#### 5.4.4(a) Effect on Duty Cycles for Source nodes

Because no packets are exchanged during the IDLE state, the nodes spend most of their time in IDLE or receiving mode, where the channel is checked for activity using (CCR rates). Therefore, as explained in the previous section, the nodes are mostly in an IDLE state with minimal energy dispensed. The minor variations in duty cycle (as observed in Figure 5.18) corresponds to the fact that requires additional computation, thus more CPU activity. This trend is evident for KiBAM, TKiBAM and the proposed scheme in contrast to Contiki OS’s default duty cycle.

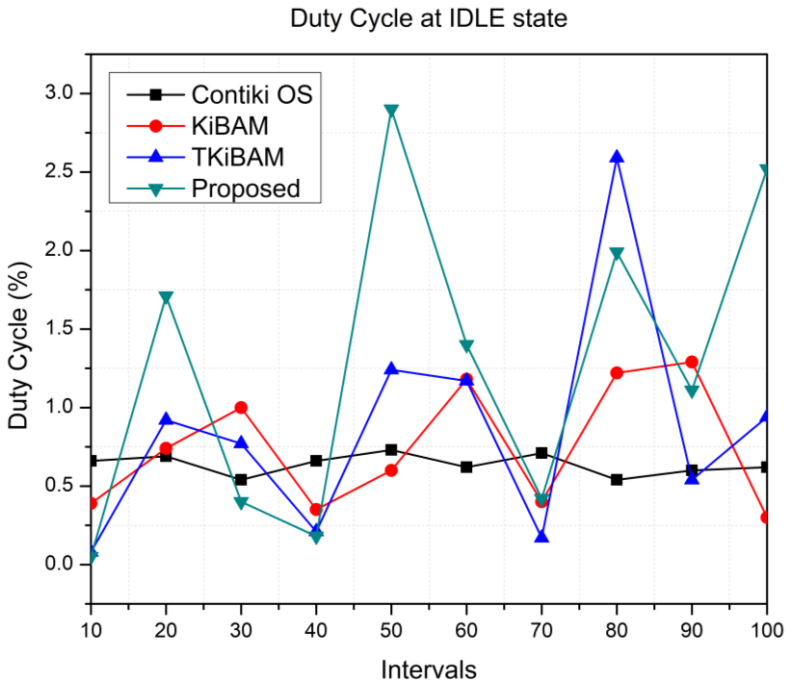


Figure 5.18: Average Duty Cycle for source node at IDLE State (0 pkt/s)

Similarly, the duty cycle performance of source nodes at higher data rates reveals that ContikiMAC protocol in its default configuration incurred lesser overhead. Whereas other schemes had to increase their duty cycles to maintain the PDR. This increase in duty cycle corresponds to the additional computational requirements in CPU state activities as illustrated by (Figure 5.19- Figure 5.21). However, this does not imply an additional overhead in RDC (particularly in RX and TX) states, that could dramatically increase node energy consumption (as illustrated in *Appendix F*).

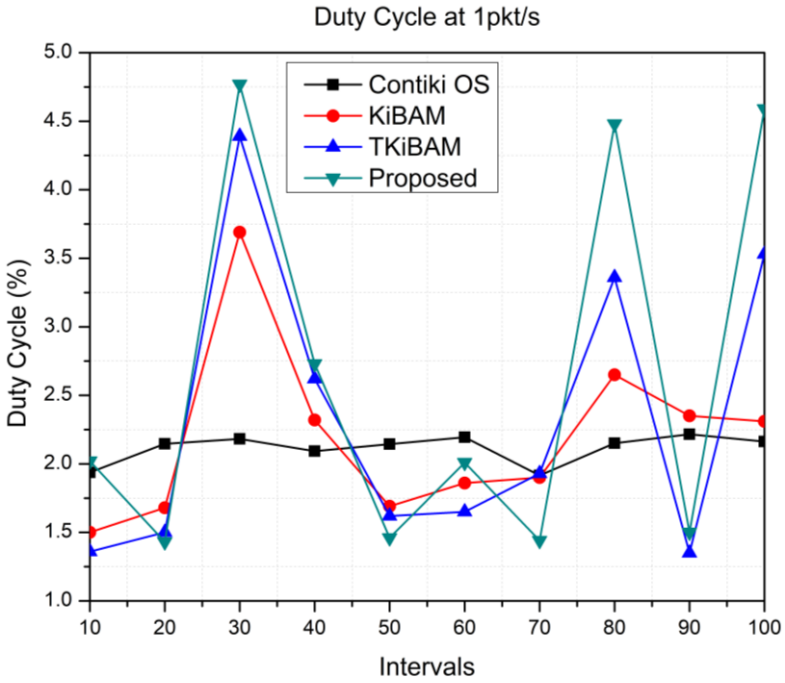



Figure 5.19: Average Duty Cycle for source node at 1 pkt/s

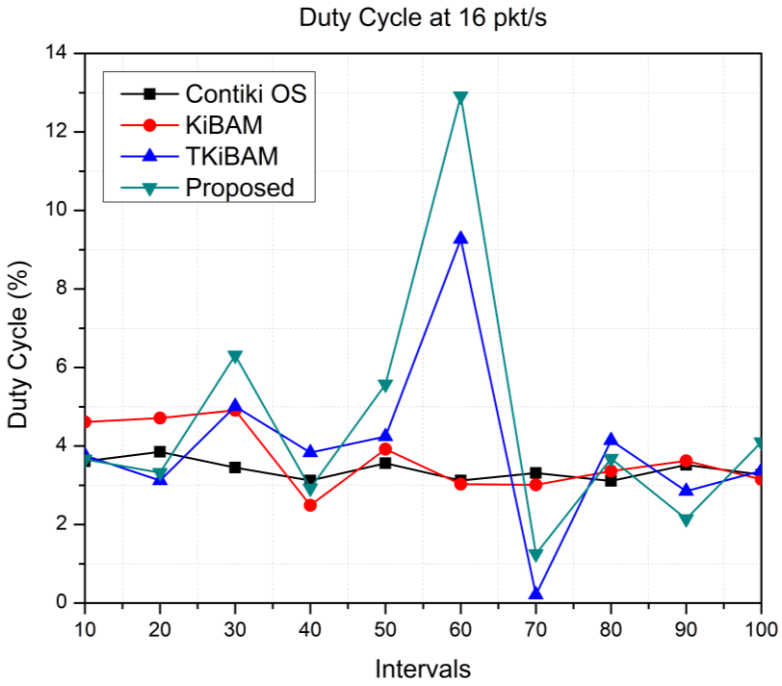


Figure 5.20: Average Duty Cycle for source node at 16 pkt/s

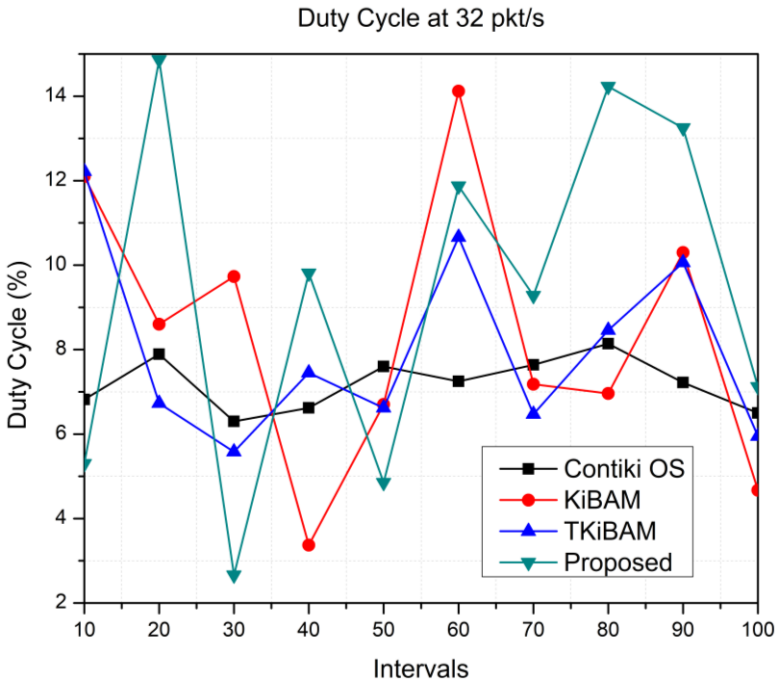



Figure 5.21: Average Duty Cycle for source node at 32 pkt/s

The proposed scheme struggles to maintain the duty cycle as the network traffic is increased. However, as the radios are mostly switched off, the additionally cost incurred in CPU state will not dramatically affect the overall lifetime of the node.

### 5.4.4(b) Effect on Duty Cycles for Sink node

Similarly, the change in duty cycle of sink (router) nodes were also investigated. In this regard, a similar pattern was observed where, the three additional schemes achieved a higher duty cycle as the network traffic was increased.

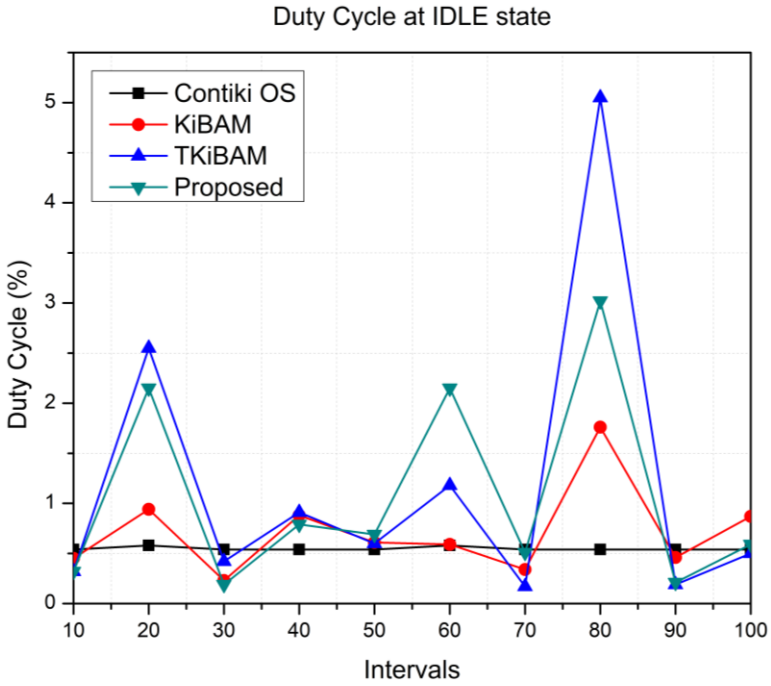


Figure 5.22:Average Duty Cycle for sink node at IDLE state (0 pkt/s)

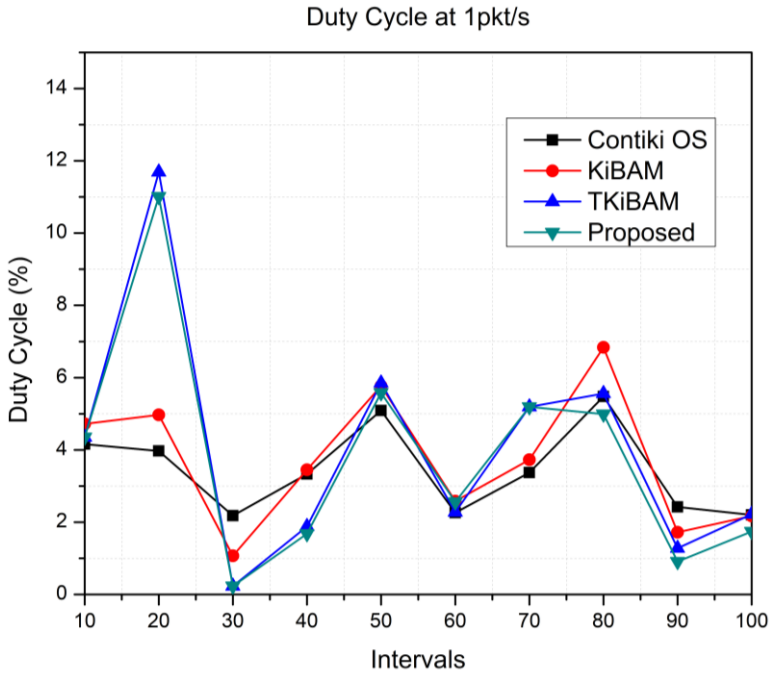


Figure 5.23:Average Duty Cycle for sink node at 1 pkt/s

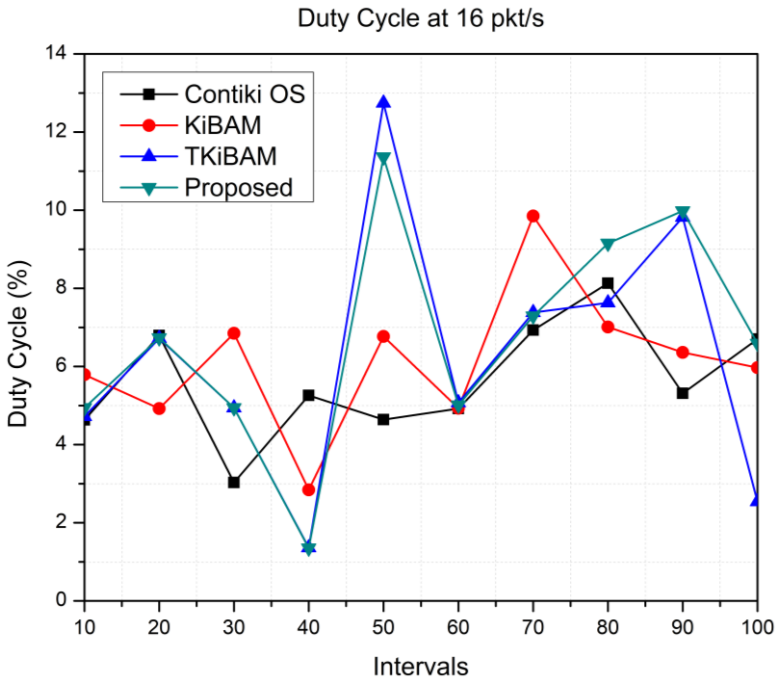



Figure 5.24:Average Duty Cycle for sink node at 16 pkt/s

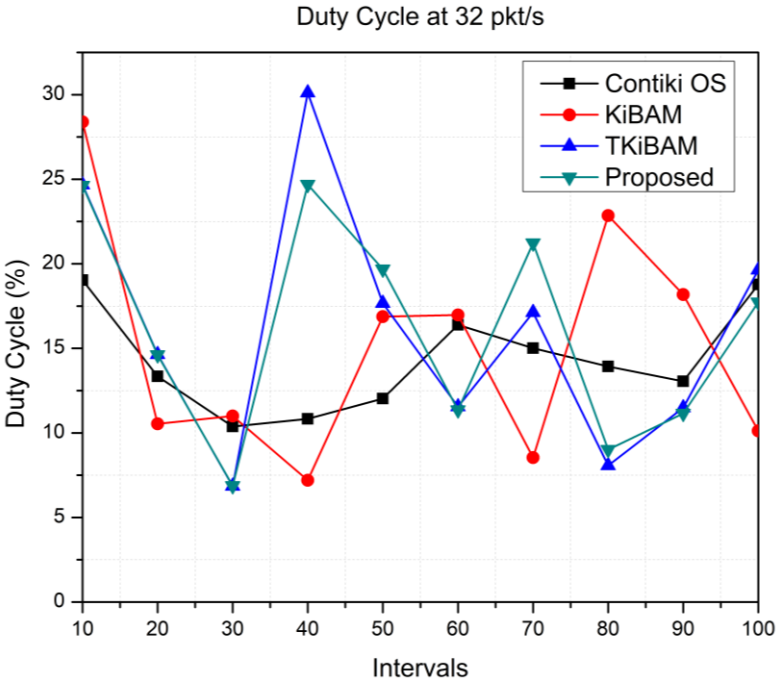



Figure 5.25:Average Duty Cycle for sink node at 32 pkt/s

As illustrated, the duty cycle increases from an average of 4 percent to 25 % with the increased in network traffic. This change is particularly evident for TKiBAM and the proposed scheme, where the algorithms need to account for additional parameters, including the effect of temperatures. The overall increase in duty cycle may

result in added computational cost that comes with increased accuracy. Table 5.6 lists down the overall impact of the increase in duty cycle in contrast to default ContikiMAC calculations.

Table 5.6: Average change in duty cycle for the schemes under investigation

| Observed Average change in duty cycle for Source Nodes (%) | | | |
|---|---|---|---|
| **Data Rate** | KiBAM Vs Contiki | TKiBAM Vs Contiki | Proposed Vs Contiki |
| IDLE | 7.26 ↑ | 5.47 ↑ | 9.05 ↑ |
| 1 pkt/s | 3.82 ↑ | 10.25 ↑ | 25.01 ↑ |
| 16 pkt/s | 8.52 ↑ | 17.33 ↑ | 35.20 ↑ |
| 32 pkt/s | 16.32 ↑ | 11.41 ↑ | 29.52 ↑ |
| Observed Average change in duty cycle for Sink Node (%) | | | |
| **Data Rate** | KiBAM Vs Contiki | TKiBAM Vs Contiki | Proposed Vs Contiki |
| IDLE | 3.10 ↑ | 11.97 ↑ | 13.79 ↑ |
| 1 pkt/s | 7.42 ↑ | 17.61 ↑ | 10.96 ↑ |
| 16 pkt/s | 8.78 ↑ | 24.10 ↑ | 19.52 ↑ |
| 32 pkt/s | 5.53 ↑ | 13.33 ↑ | 12.73 ↑ |

### 5.4.5 Node Lifetime estimation comparison

In this case, similar network topology was used, with a smaller battery capacity to speed up the process, to include:

- Comparative simulations of TKiBAM and proposed scheme to estimate the end of device lifetime
- Implementation of TKiBAM and proposed scheme on actual hardware to compare the degree of estimation accuracy

A series of simulations were conducted at various packet rate to estimate the device lifetime for the implementation on actual device. As the devices run for approximately months, it was not feasible for the research outcome. Some of the approximated device lifetimes as observed by Cooja simulations for TKiBAM and the proposed schemes are given in Table 5.7 .

Table 5.7: Simulated device lifetime estimation using TKiBAM and proposed scheme.

| Device lifetime (in months) | | | | |
|---|---|---|---|---|
| | **TKiBAM** | | **Proposed scheme** | |
| **Data rate** | **End-Node** | **Router** | **End-Node** | **Router** |
| IDLE | 39.94369019 | 43.5572884 | 39.9685 | 45.7603 |
| 1 pkt/s | 13.77238198 | 9.11597691 | 13.7678 | 9.4674 |
| 16 pkt/s | 8.235222894 | 5.7413904 | 8.2671 | 5.6938 |
| 32 pkt/s | 3.934619805 | 2.38639225 | 3.8917 | 2.3831 |

As the number of packets increases, the GPR-based scheme must do more work because of the algorithm, which takes more power than the TKiBAM scheme. However, the extra CPU time it takes to figure out voltage after each CPU state (CPU, LPM, RX, and TX) draws a little more power. This is clear from the calculations, which show that the proposed scheme uses more power than the other schemes do. As each packet rate scenario was chosen, the CPU cycles stayed the same. However, the extra CPU cycles to figure out which voltages were available were added to the overall CPU cycle for the simulation time. This causes the CPU to run more often, which means that the device will last less time. However, because the proposed scheme uses an analytical model for offline calculations (that doesn't read the true voltages at the battery), it needs to be investigated more.

In that case, the online SOC estimation method was used on a real device. There is a TI board called the CC1352R that looks for input voltage interrupts to make calculations. The GPR SOC model was put on this board. In order to do that, a simple circuit was made to divide the battery voltages that are sent to the microcontroller. The machine learning model used this analogue voltage input, as well as information about charge quanta, to figure out the SOC and battery capacity of the battery. The input voltage and battery capacity set the ML model's weights, and the corresponding SOC is found. The information from the SOC is then used to make sure that every battery

voltage and battery capacity is correct. This information is used to figure out how much power a device will use.

There is a lot of variation in the voltage of the battery, which means that the battery voltage can't be directly used to power the device. Voltage changes in the non-linear area, but it stays the same in the flat area, which makes it hard to figure out how much energy is in the battery. So, the proposed model used battery SOC and remaining capacity to figure out how much power was being used. A Panasonic Li-Ion CR-2032 coin-cell battery (230 mAh) was used to run the device. This battery is made by Panasonic and can be found at most stores. Ideally, this should keep the device running for about two months. Two TI CC1352R boards with the proposed scheme and TKiBAM models were used to run simulations of a UDP packet broadcast code to get an idea of how long the devices would last, as well. The length of a UDP packet is 127 bytes. To speed up the simulation, a data rate of 32 packets per second was used. When these packets were sent to a workstation, the CC1352R board that was connected to it was a sink for them. In this case, a sink is needed to show that the packets have been received and not send them again. As there were only two nodes running, there was very little chance that packets would be sent back and forth. Therefore, no extra power is used by external sensors and passive components like LEDs on the board.

The workstation was running the TI Code Composer studio – PowerTrace application, which kept track of how many packets were received. This made it possible to keep an eye on the device. Both nodes were read by a simple Python-based script that was run. The python-based logger started the first-time stamp as soon as the first packet came in. It kept track of the CPU time until the last packet came in. The difference between these timestamps helped to keep track of how long the nodes were running and, thus, how long each one had been alive. These results are detailed in Table 5.8.

Table 5.8: Experimental evaluation of proposed schemes for device lifetime estimation.

| Lifetime (months) | | | | |
|---|---|---|---|---|
| **TKiBAM** | | | | |
| | **Empirical Estimation** | | **Real Device Implementation** | |
| **Data-rate** | **End-node** | **Router** | **End-Node** | **Router** |
| **32 Packet/s** | 3.94011 | 2.3896 | 2.1981 | 2.015 |
| **Proposed scheme (GPR SOC Estimation)** | | | | |
| | **Empirical Estimation** | | **Real Device Implementation** | |
| **Data-rate** | **End-node** | **Router** | **End-Node** | **Router** |
| **32 Packet/s** | 2.8917 | 2.1639 | 2.2051 | 2.003 |

The device lifetime of both source and sink nodes was initially estimated using TKiBAM and revealed a longer device lifetime. However, the experimental results revealed a great deal of variation, casting doubt on the TKiBAM model's accuracy. Although the model is non-linear and accounts for actual battery voltages, the mathematical implementation of computing the effect of temperature and load current can further complicate the process, leading to inaccuracies, as reported in Figure 5.26.

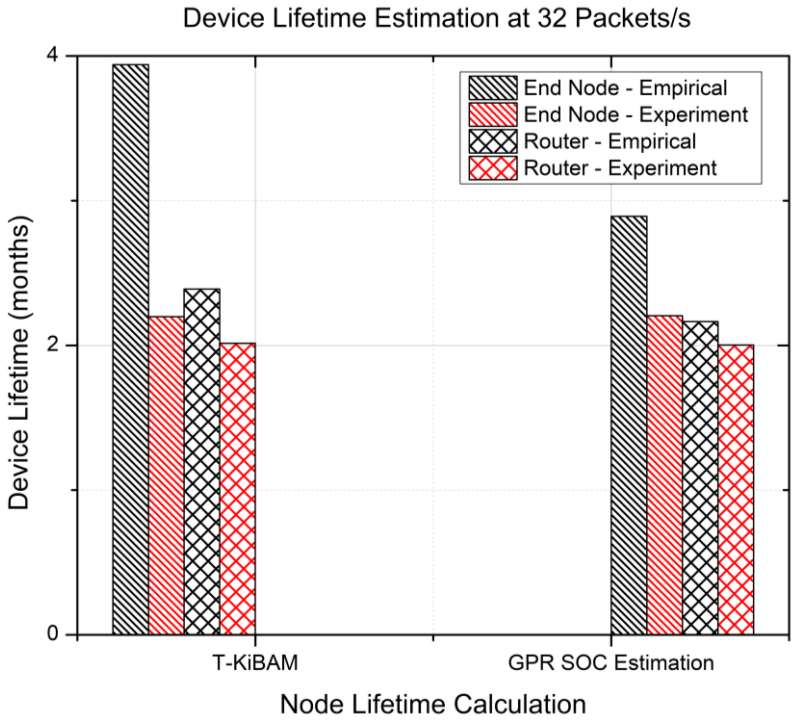


Figure 5.26: Experimental evaluation of remaining device lifetime

The proposed scheme illustrated a shorter device lifetime for both the sink and the source nodes, however, the experimental results from the actual device revealed very little deviations. This resulted in more accurate and predictable results for device operation. Furthermore, the proposed scheme incurred less additional computational costs as compared to TKiBAM and can therefore be used for accurate device estimation.

## 5.5 Summary

First, the Contiki operating system's networking capabilities and configurability were enhanced in order to implement a WSN environment. For the experimental observations, the environment parameters, environment, and network topology were introduced. In addition, the implementation of the proposed scheme in the Cooja simulation environment, as well as the KiBAM and TKiBAM energy estimation models were detailed. We've also discussed how different network traffic rates affect energy consumption and overall packet delivery rates.

Finally, TKiBAM and the proposed scheme were tested on the actual hardware for accurate energy estimation comparison. The proposed scheme was found to provide very accurate estimates of the remaining device lifetime, as demonstrated by simulations and laboratory experiments. It was also discovered that, when compared to the default Contiki energy estimation module, the proposed scheme had higher computational costs. The ability to accurately estimate various battery chemistries over a wide range of temperatures that lead to accurate energy estimation, which is required to design energy-aware algorithms for longer device lifetime. In the following chapter, the proposed energy-efficient algorithm (A-CCA) for extending node lifetime is presented.

# CHAPTER 6

# ADAPTIVE CLEAR CHANNEL ASSESSMENT (A-CCA)

In this chapter, the integration of previous objectives with the proposed energy conservation MAC algorithm is presented. To provide accurate SOC estimation under various environmental conditions, the focus was previously on battery characterization, SOC estimation, and on-device model implementation. The resulting schemes were able to accurately estimate the remaining battery energy, which will be used to create an energy-aware algorithm to extend the device's lifetime even further. An Adaptive Clear Channel Assessment (A-CCA) MAC layer algorithm is presented in this regard, which adjusts its CCA period, as well as the transmit power, to reduce CPU and radio activities in channel listening and data packet re-transmission. For accurate onboard energy estimation and longer WSN node lifetime, the proposed scheme can be implemented on any IEEE 802.15.4 based radio platform.

## 6.1 Energy detection and Clear Channel Assessment

The IEEE 802.15.4 provides multiple strategies for clear channel assessment using either a simple non-coherent energy detection scheme, carrier sensing scheme or the combination of the two. The maximum allowed symbol detection time is 128 μs or 8 symbol periods, where the PHY layer must inspect the channel for activity and report its finding to the upper MAC layers. To avoid outliers during channel inspection, the standard directs the use of two consecutive CCAs to reduce false wakeups probability and to ensure a successful active/sleep period of the nodes. The standard is flexible in other implementation details. Therefore, many CCA detection mechanisms and the number of CCAs implementations are observed in transceivers and MAC implementations. CCA implementations profoundly affect energy detection simplicity

and accuracy, which would aptly affect the overall network performance. IEEE 802.15.4 uses 32-bit long chip sequences with every packet containing 8 repetitions of the symbol code to generate the sequence. One of the easiest methods to detect these signals is based on non-coherent energy detectors. As in a typical narrow-band radio system, most of the signal power is wrapped around its centre or carrier frequency. The concentrated energy can be detected reliably over a moderate to low noise floor (Iyappan Ramachandran & Roy, 2007).

Simple non-coherent energy detection module, as presented in CC2650 transceivers, examines the incoming energy signal from multiple branches and compares it to the pre-defined threshold based on the network's selected transmit power levels. The quadrature receiver as given in equation (42), with two branches, namely I and Q, sums up the received signal strength on these branches ($x_{Ii}$ and $x_{Qi}$ respectively).

$$E = \Sigma_{i=1}^{N}\left({x_{Ii}}^2 + {x_{Qi}}^2\right) \tag{42}$$

Where N is based on the symbol period and chip sampling rate (N = 8 x 32 as per IEEE 802.15.4) CC2650 transceivers provide energy detection and carrier sensing receivers. However, many researchers identified that the transceiver provided CCA using an energy detection technique that suffered low Signal to Noise ratios (SNRs), thus making it link dependent and unreliable during heavy interference environments (King & Roedig, 2018; Petrova, Wu, Mahonen, & Riihijarvi, 2007). Due to this uncertainty and strong dependency on the link, inaccurate CCA can result in network traffic collisions. These false wakeups would waste a significant amount of network time and energy in combatting the false alarms. To this aim, the basis of this research is to study the implementation of CCA in a ContikiMAC protocol is studied for adaptive

CCA timing adjustments based on the traffic patterns and the accuracy of channel activity estimation with increased signal transmit powers.

### 6.1.1 MAC protocols for ED and collision avoidance in IEEE 802.15.4

Contention-based Medium Access Control (MAC) protocols are broadly classified into asynchronous and synchronous (scheduled) classifications. The scheduled MAC protocols share the active and sleep cycle with nodes within the network, forming an accurate active scheduling mechanism that requires accurate synchronization between nodes to achieve the desired accuracy. The nodes wakeup simultaneously and share the SYNC packets, which directs the active/sleep schedule cycle. Protocols such as S-MAC and T-MAC employ the packet synchronization schemes. S-MAC protocol reduces the idle listening channel period and achieves power efficiency by periodically switching into a sleep state (Wei, Heidemann, & Estrin, 2004).

In comparison, T-MAC toggles to sleep state more often if no activity is observed over a certain period (Van Dam & Langendoen, 2003; Wei et al., 2004). Synchronous mode protocols are most suited for real-time data transmission as the entire network converges and effectively delivers the packets on the prescribed schedule, which helps maintain high throughput and low latency. Asynchronous approaches do not share the schedule with the network nodes, thus requiring periodic wakeups and channel estimation for data transmission (J. Kim, On, Kim, & Lee, 2008). Asynchronous protocols rely on low power listening (LPL) methods, where the PHY layer samples the medium before transmission to link a sender to a receiver (Sha, Hackmann, & Lu, 2013). The receivers in asynchronous mode conserve energy by continuously duty cycling in its active/sleep states. Therefore, CCA must effectively probe the medium and report its state effectively to avoid any false wakeups that may

consume unwanted energy on the receiver side. Asynchronous MAC protocols can be further classified into Sender-initiated and Receiver-initiated protocols. Typical examples of sender-initiated asynchronous mode MAC protocols include B-MAC, X-MAC and ContikiMAC. In these schemes, the source strobes the medium with broadcast beacons to notify its intent to transmit. The broadcasts intervals are usually made longer than the receiver's wakeup periods, reducing the synchronization overhead. Continuously strobing the channel with broadcast beacons consumes much energy on the sender node and keeps the channel busy for a more extensive period. B-MAC protocol transmits a longer broadcast preamble to notify its intent to transmit, which keeps its radio awake for a more extended period consuming more energy. X-MAC continuously strobes the medium with shorter broadcast beacons, reducing energy consumption. A successful initial transmission stores the receiver address in its unicast frame, which further improves the latency and over-hearing problem (Buettner, Yee, Anderson, & Han, 2006). ContikiMAC takes an entirely different approach, whereby strobing actual data packets instead of the broadcast preambles to notify its intent to transmit (Dunkels, 2011; Michel & Quoitin, 2014).

Receiver initiated asynchronous MAC protocols are preferred in extensive, dense WSN deployments with interference and channel occupancy. Unlike the sender-initiated schemes, in these protocols such as RI-MAC, RIX-MAC and RICH-DP MAC, the sender listens and responds to the receiver broadcast beacons staying most of the time in an idle-listening state (Y. Sun et al., 2008). These schemes are most suited for event-driven and low-data rate WSN applications, allowing little channel occupancy. However, the sender consumes a significant amount of power while always listening to receiver packets in an idle-listening state. In this research, sender-initiated protocols

such as ContikiMAC are preferred for its ability to adapt to IEEE 802.15.4 based CCA timings requirement, low-latency, and high packet delivery ratio (PDR).

### 6.1.2 Investigating effect of CCA adjustment

In a default implementation of the IEEE 802.15.4 standard, a typical transceiver performs two consecutive CCAs (CCA1 and CCA2) to check for channel availability. The CCAs are 294 μs apart, looking for channel activity, where the radio is powered down until the next consecutive CCAs if there is no channel activity. The interval between the consecutive CCAs is set to be 500 μs, which reduces the node's additional energy consumption. In case the channel is idle, and the sender broadcast beacons are detected, the receiver initiates the transfer by sending an acknowledgment back to the sender. The transmission window controls the ongoing active transmission length, which is acknowledged by a third single CCA check. In the proposed scheme, the distance between two consecutive CCAs (CCA1 and CCA2) was increased based on sender broadcast beacons' length, thereby increasing the channel investigation window time for a better probability to receive frames. The length of the ongoing transmission window was adjusted based on the sender's data frame length, which allows the sender to mitigate overlapping data frame windows and receive acknowledgment for each data frame in the ongoing transmission.

In Contiki OS implementation, the nodes frequently listen to the channel for the incoming data packet, which is controlled by Channel Check Rate (CCR), with a default value of 8 Hz, but can be extended in increasing exponential power of 2 ($2^0$, $2^1$, $2^2$, … $2^5$ Hz). The proposed scheme with improved receiver wakeup estimations was evaluated using XMAC and ContikiMAC protocols with increasing channel check rates. The performance is evaluated in terms of radio duty cycle (RDC) and average packet delivery ratio (PDR), as given in Figure 6.1

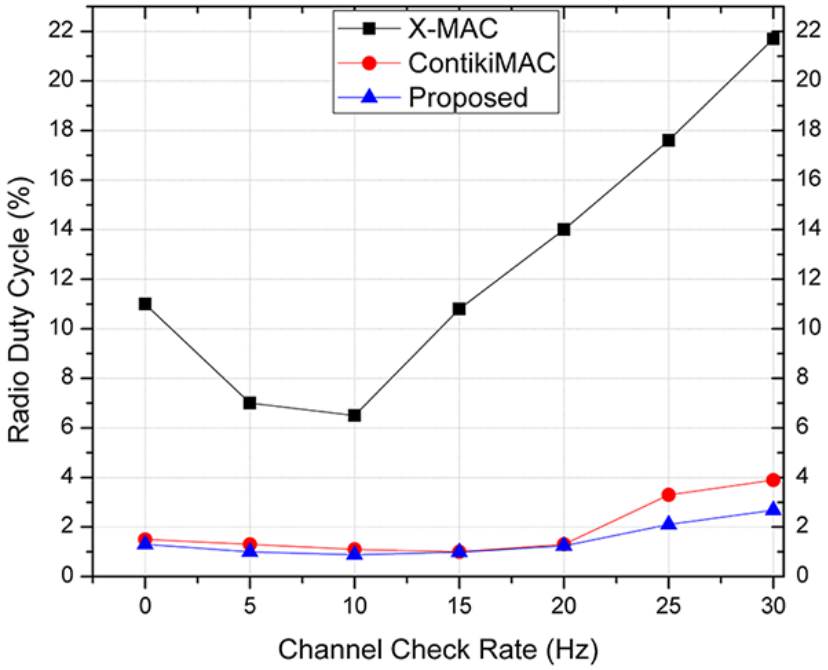


Figure 6.1: Impact on radio duty cycle with increased channel check rate.

The investigation revealed that the proposed scheme performs well for lower channel check rates. The proposed scheme also enabled the MAC protocol to maintain a low duty cycle due to the proposed CCA adjustments' performance improvements. However, as the CCR further increased, the receiver had to monitor the incoming data channel continuously. The increased length of the CCA frame overlapped with the sender data frame and could not communicate. The proposed scheme still performed significantly well as compared to the default ContikiMAC implementation. X-MAC protocol uses short preambles to achieve power efficiency by making use of an extended period in low power listening states. X-MAC also de-couples the sleep schedules at both transmitter and receiver sides, which further helps to improve its hop latency for bursty network traffic.

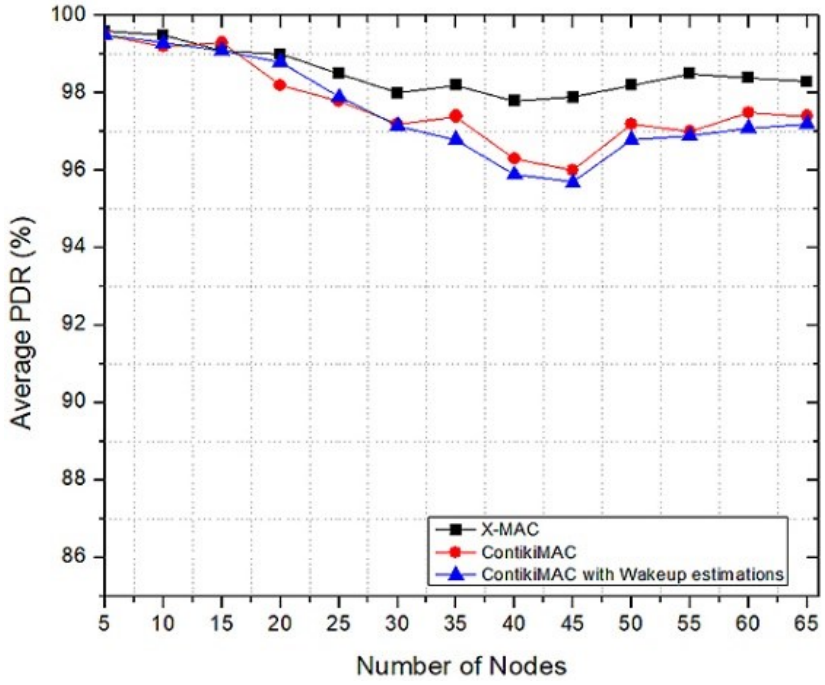



Figure 6.2: Impact on average PDR with increased network load

As evident from Figure 6.2, X-MAC maintains a higher PDR value for heavy traffic (as the number of nodes increase). With the increase in nodes (thus the network traffic), X-MAC initially drops the PDR, due to network congestion resulting in re-transmission. At the same time, in an X-MAC preamble, the target ID is sent as part of the short preamble, which reduces the overhearing problem in case of a busy network under collision. It is due to this scheme that X-MAC protocol need to wait a shorter time in case of re-transmission due to the length of its short preambles.

In a typical ContikiMAC protocol implementation, a larger number of short preambles are required to govern the data transmission. Initially, for a smaller network load (lesser nodes and traffic), the network collision is minimal, resulting in lesser re-transmissions and overhearing problem. ContikiMAC protocol, during its clear channel assessment (CCA) stage, estimates the channel availability based on last received signal strength indicator (RSSI) values. Initially, with the increase in network traffic, the collision increases thus resulting in a negative CCA operation. A significant number of data packets are retransmitted following the linear CSMA back-off interval that

increases with each transmission. Therefore, as the network traffic is increased further, the back-off interval is further increased, resulting in an overall reduction of network traffic.

Similarly, the response for increased network load was investigated to evaluate the performance of the successful transmission. It can be deduced that the proposed scheme performed well and showed an improvement of up to (2%) average PDR for less dense networks. However, as the number of nodes increased beyond a certain threshold, the average PDR started decreasing. With the increase in the number of nodes, a higher probability of sender traffic is observed. As the proposed scheme relies on its wakeup cycle adjustments based on sender broadcast beacons, with increased sender comes many broadcast beacons, mostly overlapping during CCA check. The receiver had to adjust its wakeup periods several times, as there was no fixed schedule. Therefore, the node could not communicate and did not initiate the data frame transfer, resulting in reduced network throughput.

These performance evaluations shed light on two significant factors that became the basis for the next set of experiments and performance evaluations of this study. The results highlighted:

- Radio wakeup estimations based on sender broadcast beacons can perform well on a uniform low data rate network
- With the increase in network density and traffic, the time-based CCA adjustment on the receiver suffers from estimating the sender broadcast beacon due to an increase in the noise floor

In summary, CCA's length had a direct impact on network performance. A fixed CCA length limits the receiver's ability to align its wakeup periods and may affect the overall network performance for different traffic conditions. In a dense network with

high traffic, the CCA detects the channel state (mostly busy) but struggles to detect the sender beacons, limiting the transceivers' ability to wait until the back-off period is over. These performance metrics enabled us to extend the CCA enhancement by looking into the following parameters to define A-CCA for the next experiments, as discussed below.

- Performance evaluation of ContikiMAC protocol in a various network traffic scenario
- Design of A-CCA algorithm to adaptively adjust CCA periods based on network traffic
- Increasing sender transmission power for better detection over the noise floor

## 6.2 Adaptive Clear Channel Assessment – System Design

Contiki Operating System was chosen for this study because of its ability to implement the complete TCP/IP stack, low-power routing and MAC protocols, and the ability to generate native code that can be run on the built-in Cooja simulator. Another appealing feature of Contiki OS is the ability to run native code on supported hardware, such as a Texas Instruments MSP430-based microcontroller (MCU) platform with an IEEE 802.15.4-based CC2650 transceiver for results validation. In Cooja Simulator, the experiments looked at the CCA behavior of the ContikiMAC protocol on T-mote sky platforms. For power transmission, range, and accuracy validation, the Texas Instrument Launchpad CC1352 board was used. The CPU, LPM, Tx, and Rx states of CC2650 transceivers are used to estimate the exact amount of energy consumed by the radio in each state. It is critical to keep track of protocols, performance evaluations, and optimization effects in each state. The CCA adjustments on the receiver side of the ContikiMAC protocol were heavily influenced by network traffic and CCR. To understand the ContikiMAC behavior in its default state, experiments were conducted

to evaluate the ContikiMAC protocol's performance for varying network traffic against multiple CCR.

A-CCA is an adaptive, lightweight, computationally flexible extension to ContikiMAC protocol designed to enhance CCA detection in light and heavy interference environments. It improves the receiver wakeup times accuracy by probing CCA timestamps between improved RSSI threshold values set above the noise floor to detect sender signals. A-CCA achieves this by sampling channel noise floor for activity and measuring the sender's RSSI signal strength with adaptive transmit power for heavy interference environments. A-CCA maintains the network throughput while reducing the duty cycle of the ContikiMAC protocol. A-CCA is designed to be implemented in a typical network deployment scenario on co-ordinator nodes, whereas the overall network throughput performance can be measured at the sink node. It is also worth mentioning that a fixed network with uniform CCR and packet is required for the proposed scheme. A-CCA logical flow is presented in Figure 6.3 by performing the following implementation steps:

- Probe the channel for activity and return CCA status
- Sample the channel for 8 symbol periods and calculate average RSSI
- Set RSSI threshold to last received RSSI value
- Adjust CCA back-off timings (reduce 10 symbol period of low RSSI)
- Check battery SOC (for power levels between 100 – 30%)
- Adjust Tx radio power and set the RSSI threshold to the new signal's energy
- Probe the channel for CCA status and timestamp Tx symbols to calculate the time interval between consecutive Tx broadcast packets
- Update the radio wakeup time intervals based on the last CCA check

The IEEE 802.15.4 standard relies on two consecutive CCA checks with a prescribed interval followed by the back-off period. The implementation of the standard in protocols and the devices shows a great deal of flexibility, where the CCA periods follow the standard. The average number of CCA checks to probe the channel, RSSI levels and threshold values see a great deal of variation in different implementations. Therefore, it is imperative to understand the default IEEE 802.15.4 implementation in the ContikiMAC protocol to analyse the CCA time check intervals, RSSI value calculations and symbol period detection strategy. In its default configuration, ContikiMAC RSSI values only summarize three categories. First, the RSSI values determine an idle channel with minimum interference, with the valid transmission (resulting in positive wakeup); second, the RSSI values are not valid due to channel interference (resulting in false wakeup); and finally, it fails to detect any activity (resulting in idle listening).

As mentioned earlier, the CCA results from the ContikiMAC protocol do not contain any information on channel conditions other than the states. To understand the protocol behavior, it is essential to investigate Contiki OS's network stack and associated functions required to perform these operations. Therefore, the ContikiMAC protocol, its network stack and RSSI calculations are described.

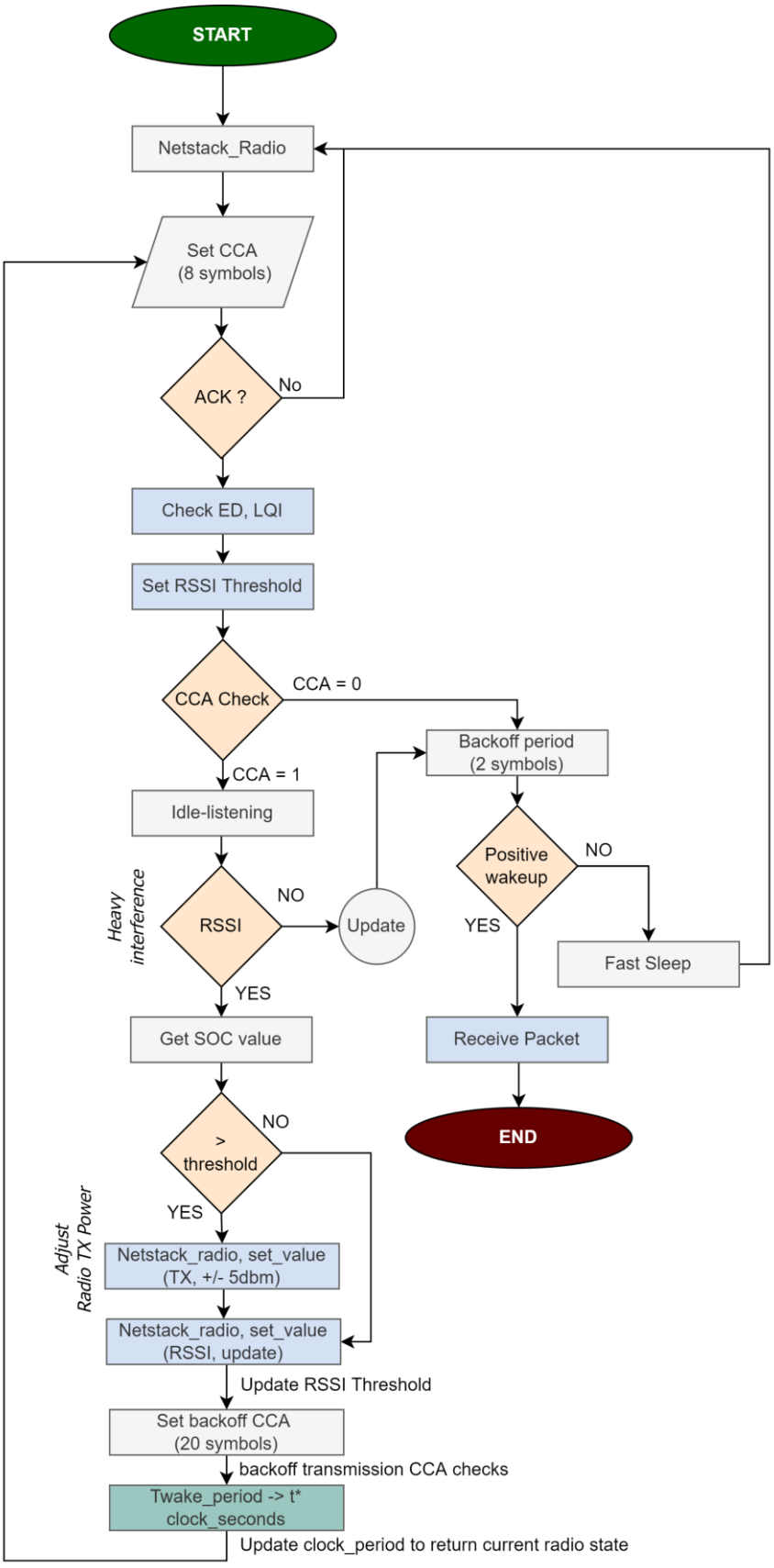



Figure 6.3: A logical flow diagram of the proposed A-CCA algorithm

### 6.2.1 Network Environment and Configurations

These experiments are based on T-mote sky WSN nodes that have an MSP430 MCU architecture. The experiments are performed using Contiki OS with the compiled byte code executed in its Cooja simulation environment. The T-mote Sky platform maintains its internal clock with MSP430F1611 modules providing a clock frequency of 32.768 kHz. It is crucial to consider the precise timing requirements for these experiments as it directly translates to the total number of CCA checks performed to measure channel activity. Contiki OS dispatches its processes scheduler based on a polling mechanism, where event handlers are attached to every process and pre-empted over fixed intervals of time. Contiki OS providers event timers called RTIMER that keeps track of run time event time calculations. Whenever a packet is scheduled to be transmitted, the ContikiMAC protocol performs a channel sensing over a prescribed length of time (set to be 8 symbol periods or 128 μs). Therefore, the real-time timer on T-mote Sky mote ticks approximately 16,384 times per second in Contiki OS. The system clock ticks and RTIMER values are then required to calculate the time intervals in handling every event and the associated processes. CC2650 radio transceiver initially investigates the channel for RSSI value by comparing it to a pre-defined programmable threshold value, resulting in three-channel states as discussed previously. Whereas, the RSSI values are an averaged sum of signal strength over 8 symbol periods. The combination of both measures the interval and the average noise floor level of the channel required by A-CCA to perform correctly.

#### 6.2.1(a) Network requirement

In this study, a mesh network topology comprising of 20 T-mote sky WSN nodes were simulated (to emulate the radio profiles based on CC2650 radio). One of the fundamental challenges in meshed networks is the network convergence time for

the routing protocols. In this work, the RPL protocol (Routing Protocol for Low-power and Lossy Networks) is used. The RPL protocol supports automatic mesh tree topology creation and fast convergence; both desired these experiments as the focus is on the MAC layer with a fully converged network. The nodes (1-19) are configured for RPL routing protocol running ContikiMAC protocol with its radio duty cycling layer to default configurations. Node 20 is configured with an RPL border (sink node) with ContikiMAC protocol configured to have a NULL-RDC layer that keeps the node alive and performs no duty cycling. The sink node acts as a border gateway router and receives the data packets from the network, as given in Figure 6.4.

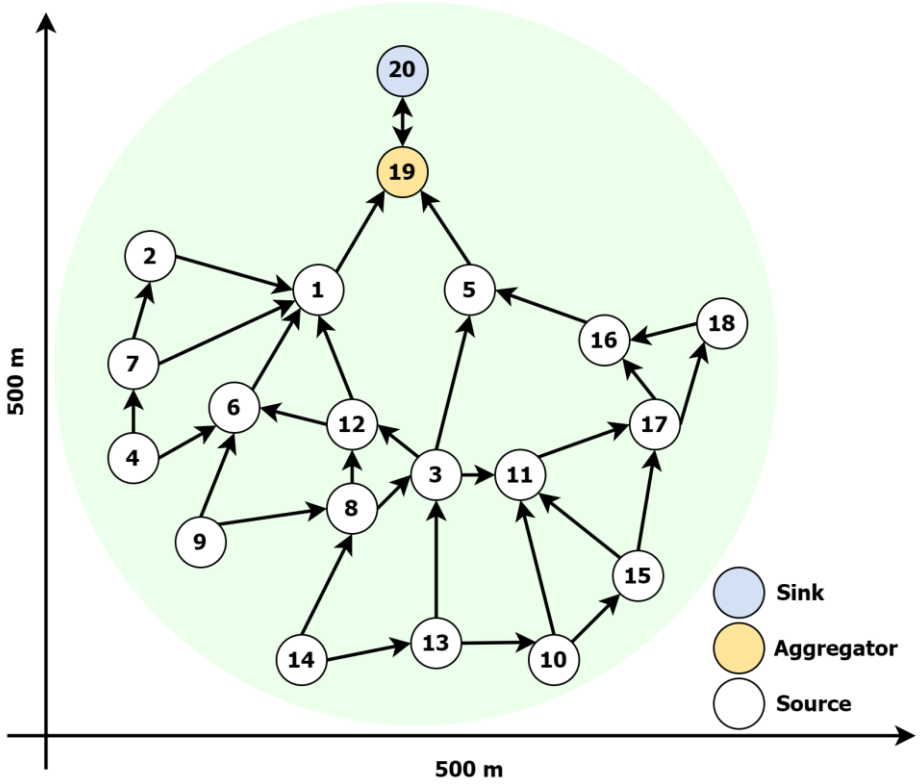



Figure 6.4: Network topology comprising of 20 T-mote sky notes for RPL routing

The above network topology along with the network mentioned above and platform configuration is used for two network scenarios, including:

- **Low-traffic** network (one packet sent every second)

- **High-traffic** network (32 packets sent every second)

It is worth mentioning that the MAC layer queue in Contiki OS can simultaneously store 6 packets; therefore, the network traffic was selected below this threshold to avoid any buffering delays. Each data packet was 127 bytes in size. In a typical communication scenario based on the above network topology, the nodes will encounter neighbouring nodes' interference. CCA values will toggle between positive wakeup, false wakeup, and idle listening states. As mentioned earlier, the Contiki OS will perform CCA checks to record the intervals between these states and RSSI values for further calculations. The Contiki OS performs these time-based calculations with several variables, each associated with a fixed schedule and interval, described in the following sub-section.

### 6.2.1(b) Contiki OS Implementation

The Contiki OS uses the NETSTACK_RADIO layer to power up the CC2420 chip and performs CCA checks. The time interval between consecutive CCA checks is handled by CCA_CHECK_TIME functions that monitor the CCA interval time, the time between two consecutive CCA checks as well as the non-activity periods, during which the radio is put to sleep CCA_CHECK_TIME returns the time calculations based on platform clock, which is then converted into ticks per seconds, based on RTIMER. After a false wakeup is detected in a typical CCA check activity, the radio is immediately put to sleep mode for an interval of above 10 CCA inactivity periods. The CCA activity time variables are given in The Contiki OS uses the NETSTACK_RADIO layer to power up the CC2650 chip and performs CCA checks. The time interval between consecutive CCA checks is handled by CCA_CHECK_TIME functions that monitor the CCA interval time, the time between two consecutive CCA checks as well as the non-activity periods, during which the radio is put to sleep CCA_CHECK_TIME

returns the time calculations based on platform clock, which is then converted into ticks per seconds, based on RTIMER. After a false wakeup is detected in a typical CCA check activity, the radio is immediately put to sleep mode for an interval of above 10 CCA inactivity periods. The CCA activity time variables are given in Table 6.1.

Table 6.1: ContikiMAC based CCA check time intervals

| **Parameters** | **CPU Cycles (Ticks)** | **Duration (ms)** |
|---|---|---|
| CCA_CHECK_TIME | 32768/8192 | 0.12 |
| CCA_SLEEP_TIME | (32768/2000) + 1 | 1.7 |
| MAX_NONACTIVITY_PERIODS | 10 x (CCA_CHECK_TIME + CCA_SLEEP_TIME) | 21 |

As mentioned earlier, the 32 kHz (32768) frequency was used for CC1352R board, whereas the RTIMER is set to 8192 (32768/4) resolutions. Although CC1352R supports a higher real-time clock-frequency, the radio drivers for ContikiMAC and Cooja simulator can only handle 32kHz. Therefore, this frequency was selected to investigate the effect on device lifetime during the simulations as well as the laboratory experiments. The system ticks and activity duration from Table 6.1 describe that the radio spends most of its time in an idle-listening state after a false wakeup is detected. This time indicates the amount of inactivity period where the radio is wasting energy. To estimate the positive wakeup and activity period of a node, Contiki OS measures false wakeup and inactive (idle listening) periods for a node; subtracting from the total node activity can yield the time spent inactive periods. This is an indirect approach, but it signifies that importance from an energy calculation viewpoint. Those states are focused, which can be monitored and optimized to improve network performance per-application basis. Therefore, to measure the false wakeup and idle-listening time, the simulation was run for 30 minutes, and false wakeup and idle-listening periods of sender nodes were calculated. The periods represent that the total time spent by a node in each

state (measured by CCA_SLEEP_TIME and MAX_NONACTIVITY_PERIODS) and are presented in Table 6.2.

Table 6.2: Idle-listening and false wakeup times of nodes running ContikiMAC

| Nodes | Idle-Listening | False Wakeup |
|---|---|---|
| 1 | 51,312 | 338 |
| 5 | 57,145 | 386 |
| 10 | 37,701 | 274 |
| 12 | 41,008 | 292 |
| 16 | 48,700 | 305 |

Using the node activity calculations, the time spent by a node in each state can be calculated by Equations (43), as:

$$T_{node} = T_{idle-listening} + T_{false-wakeup} T_{non-activity\ period} \quad (43)$$

Similarly, node energy consumption is calculated by Contiki OS's Energest module by measuring the time interval spent in each radio state, as previously given in Table 6.2. The energy consumption in Contiki OS is a linear power estimation method, where time spent in every radio state is used to measure the overall average energy consumption in the state. Since the time computation is dependent on system ticks, it can be argued that platforms with higher CPU ticks can help to deliver better resolutions. The energy consumption calculations are based on the node's datasheet parameters (as given in Table 5.2), where the battery source voltage is fixed at 3V at the start of the simulation. This voltage is then linearly reduced based on the amount of energy consumed in each node state. In a real-world scenario, different battery types, environmental factors, self-discharge, and output load are common factors that affect battery behavior. It is also widespread to see that the battery discharge profiles are non-linear, making the energy calculations using the Contiki OS Energest module less accurate. Therefore, a direct energy comparison using simulation based Energest module with real hardware may report a vast difference in power consumption and

overall network lifetime. However, the Energest module is a straightforward integrated implementation in the Cooja simulator, which can be used to calculate average power consumption of a node, as given in equation (44).

$$P_{Total} = P_{CPU} + P_{LPM} + P_{Tx} + P_{Rx} \quad (44)$$

In the next section, node time calculation in each state, overall network power consumption, duty cycle in most power-consuming states (Rx and Tx) and network throughput calculations are presented.

## 6.3 Results and analysis

### 6.3.1 Average Duty Cycle

The energy consumption and radio duty cycle estimations depend on the total time spent in each state. CC2650 registers keep track of the total CPU ticks for radio transceiver in every state used by the Energest module to compute the average time estimation per radio state. The experiments were conducted for two different scenarios where default ContikiMAC implementations were compared with the proposed A-CCA scheme. It is worth mentioning that all the calculations were made on the sender nodes by time-averaging the particular node time (in each transceiver state) and then calculating the average time of all the sender nodes (in each transceiver state). As the sender nodes were configured to use the RDC layer, the effect of default ContikiMAC configurations alongside the proposed A-CCA can be noticed. A total of 5 simulation runs were performed with each network scenario where CPU ticks were averaged iteratively. These results are presented in Table 6.3.

Table 6.3: Average time consumption of a in various transceiver states

| | Time | | | |
|---|---|---|---|---|
| **Protocol** | **CPU** | **LPM** | **Rx** | **Tx** |
| ContikiMAC (Low-Traffic) | 4598.12 | 32762.91 | 439.41 | 289.45 |
| ContikiMAC (High-Traffic) | 5099.35 | 37963.18 | 679.49 | 391.79 |
| A-CCA (Low-Traffic) | 4670.97 | 33014.19 | 401.85 | 299.78 |
| A-CCA (High-Traffic) | 5328.17 | 39116.08 | 593.21 | 387.02 |

As described, the results represent RSSI and CCA adjustments in the ContikiMAC layer. As the number of CCA checks (average of RSSI thresholds) is reduced, the transceiver spends most time in Low power states. The Rx states also show improvements, where in A-CCA scheme the radio spends less time in the Rx state for both network scenarios. The MAC protocol performance is typically measured in terms of its duty cycle, where a lower duty cycle with the desired throughput is considered ideal for power-efficient networks. The effect of CCA and radio duty cycling can be observed on Rx and Tx states (the two most power-hungry radio transceiver modes). The radio duty cycle for these states can be calculated by comparing the ratio of operation with other radio states. The radio duty cycles for Rx and Tx modes covering both network scenarios are given in Table 6.4.

Table 6.4:Average Radio Duty Cycle for Rx and Tx modes

| | Radio Duty Cycle (%) | |
|---|---|---|
| **Protocol** | **Rx** | **Tx** |
| ContikiMAC (Low-Traffic) | 1.17 | 0.77 |
| ContikiMAC (High-Traffic) | 1.57 | 0.91 |
| A-CCA (Low-Traffic) | 1.06 | 0.80 |
| A-CCA (High-Traffic) | 1.33 | 0.87 |
| Radio Duty Cycle improvement (Low-Traffic) | 1.10 | 1.03 |
| Radio Duty Cycle improvement (High-Traffic) | 1.18 | 1.04 |

The results indicate an overall improvement in radio duty cycles (Rx and Tx modes) for both network scenarios. The network performance improved slightly due to

CCA adjustments offering more radio transceiver time in low power listening mode. The results clearly show that together with increased time spent in LPM mode and reduced duty cycle in Rx mode, the node energy consumption is improved as described in the next sub-section.

### 6.3.2 Average Network Power Consumption

The Energest module computes each node's energy consumption by calculating the amount of time spent in each radio state and integrates it over the simulation period, thus returning average power utilization in each transceiver state. The power estimations are based on equations (38)- (41). The average power consumption is calculated for each transceiver state with the associated power consumption, as provided in Table 5.2. The effect of power consumption and PDR for both network scenarios are compared against the ContikiMAC protocol. Table 6.5 describes the overall network performance for ContikiMAC and the proposed A-CCA scheme.

Table 6.5: Average network Power consumption

| | **Power Consumption (mW)** | | | | | |
|---|---|---|---|---|---|---|
| **Protocol** | $P_{CPU}$ | $P_{LPM}$ | $P_{Tx}$ | $P_{Rx}$ | $P_{Total}$ | **PDR (%)** |
| ContikiMAC (Low-traffic) | 0.664 | 0.142 | 0.411 | 0.701 | 1.918 | 98.9 |
| ContikiMAC (High-traffic) | 0.639 | 0.143 | 0.483 | 0.951 | 2.216 | 98.6 |
| A-CCA (Low-Traffic) | 0.671 | 0.141 | 0.422 | 0.641 | 1.875 | 98.9 |
| A-CCA (High-Traffic) | 0.647 | 0.142 | 0.462 | 0.800 | 2.051 | 98.8 |

The output shows that the average power consumption trends for ContikiMAC and A-CCA scheme in LPM and CPU modes are very similar. A slight increase in A-CCA CPU mode power consumption is assumed to be the result of additional computations performed by the algorithm. The A-CCA scheme improves consumption in both Rx and Tx modes, where an improvement of (0.06 and 0.15 for Rx, 0.01 and

0.02 for Tx modes) mW is observed for low-traffic and high-traffic scenarios, respectively. Furthermore, both schemes can maintain significantly high network throughput in terms of PDR values. However, the A-CCA algorithm performs well for both low and high data rate traffic. It is also observed that the effect of an increase in transmit power helped reduce the false wakeups due to which the nodes spent extended time in low power modes, which helped conserve energy for the radios in the receiver state as they were put to fast sleep.

### 6.3.3 Node lifetime estimation

The simulations were run to estimate node lifetimes based on the network and node requirements previously described. ContikiMAC was first compared in its default configuration to the proposed technique, which uses a GPR-based algorithm to estimate battery SOC, which is then used in the A-CCA algorithm for device optimization. ContikiMAC kept track of how much time each transceiver spent in each state, which is necessary for power estimation and node lifetime estimation. The Energest module uses a linear power calculation algorithm that takes battery voltage into account and calculates how much power is consumed in each radio state. Because of the modular approach, the Energest module can be used in simulations to record each node's power consumption in the network. In a real-world system, however, batteries behave differently and can affect the device lifetime estimation. For the T-mote sky platform, the power consumption profile was used, and the time spent in each radio state was calculated over the simulation length. ContikiMAC's default configuration assumes an ideal scenario (ignoring battery type, electrochemistry, and external factors, and assuming a starting voltage of 3V), whereas the proposed scheme took non-linear battery discharge behaviour into account. To investigate the effects of both schemes, a Lithium-ion battery with a capacity of 230 mAH was used in these simulations.

Integrating the currents (mA) for each transceiver state for the specified time is used to estimate the node lifetime. For ContikiMAC and the A-CCA scheme, Table 6.6 shows the trends in the node's remaining life.

Table 6.6: Node lifetime estimation (simulations) comparison of ContikiMAC and proposed scheme.

| Protocol | $I_{Total}$ (mA) | Lifetime (months) |
|---|---|---|
| ContikiMAC (Low-traffic) | 14.348 | 3.50 |
| ContikiMAC (High-traffic) | 20.763 | 2.03 |
| A-CCA (Low-Traffic) | 13.8372 | 3.02 |
| A-CCA (High-Traffic) | 19.108 | 2.35 |

The load attached to the battery has a significant impact on the discharge profiles, which may or may not follow a linear pattern. In a real-world setting, the combined effect of these experiments can extend the node's life by several days. The findings suggest that A-CCA performs admirably and contributes to the node's longevity. Even though ContikiMAC uses a purely linear model to estimate the radio's energy consumption in each state, a few factors influence its overall accuracy. First, the actual voltage levels drop quickly as the voltage is iteratively reduced in each cycle based on the power consumption. Therefore, the amount of battery energy available at any given time decreases dramatically. The batteries spend nearly 80% of their time in the flat region, where voltage variations are almost non-existent, as explained in previous sections. This means that the batteries maintain the same voltage levels for nearly 80% of their operation, resulting in fewer drops in the power calculation equations and thus a higher estimate of available battery energy. This increase in available energy is more precise and translates to device lifetime. Second, as network traffic increased, the ContikiMAC used the Energest module in its default configuration to estimate energy consumption for each radio state. As the number of radio transitions increases, combined with a linearly decreasing voltage level, the available battery energy decreases, resulting in inaccuracies and a shorter device lifetime. The proposed

scheme not only estimates the available battery SOC accurately, but also optimises transmission by adjusting the CCA, transmission levels, and duty cycle. Therefore, fewer retransmissions are required, channel availability is improved, and overall energy consumption is reduced.

### 6.4 Real-time on-device Investigation

It is also likely that a bias may result based on the simulation environment where initial voltage, CPU resources, and network configurations may affect the node performance. Therefore, a real-time on-device comparison of these techniques was required to estimate the actual behavior with real-world device and environment variables. In this regard, a point-to-point communication scenario was designed for both ContikiMAC (in default configuration) as well as the proposed A-CCA scheme transmitting 32 packets per second. Two CC1352R boards were configured as source nodes, while the other two boards were configured as sink nodes separately, each running the above-mentioned schemes. This ensured a point-to-point communication between these devices using the schemes under investigation. Additionally, in order to eliminate direct radio interference of these devices, the laboratory experiments were performed independently for each scheme. However, it is pertinent to mention that these experiments were performed in an office environment to estimate the real-world impact on device lifetime due to a noisy channel floor. In these settings, office equipment such as WiFi devices, metal cabinets, electronic equipment may add to the radio interference and can present a real-world experiment. The sink nodes were connected to the PC (highlighted blue in the floor plan), whereas the source nodes were mounted on the walls (highlighted green in the floor plan). The PCs handling sink nodes were configured with Texas Instrument EnergyTrace modules to record received packets,

radio powers, channel interference, and transmission power levels, which were later used for various performance metric calculations. The floor plan for these experiments is presented in Figure 6.5.

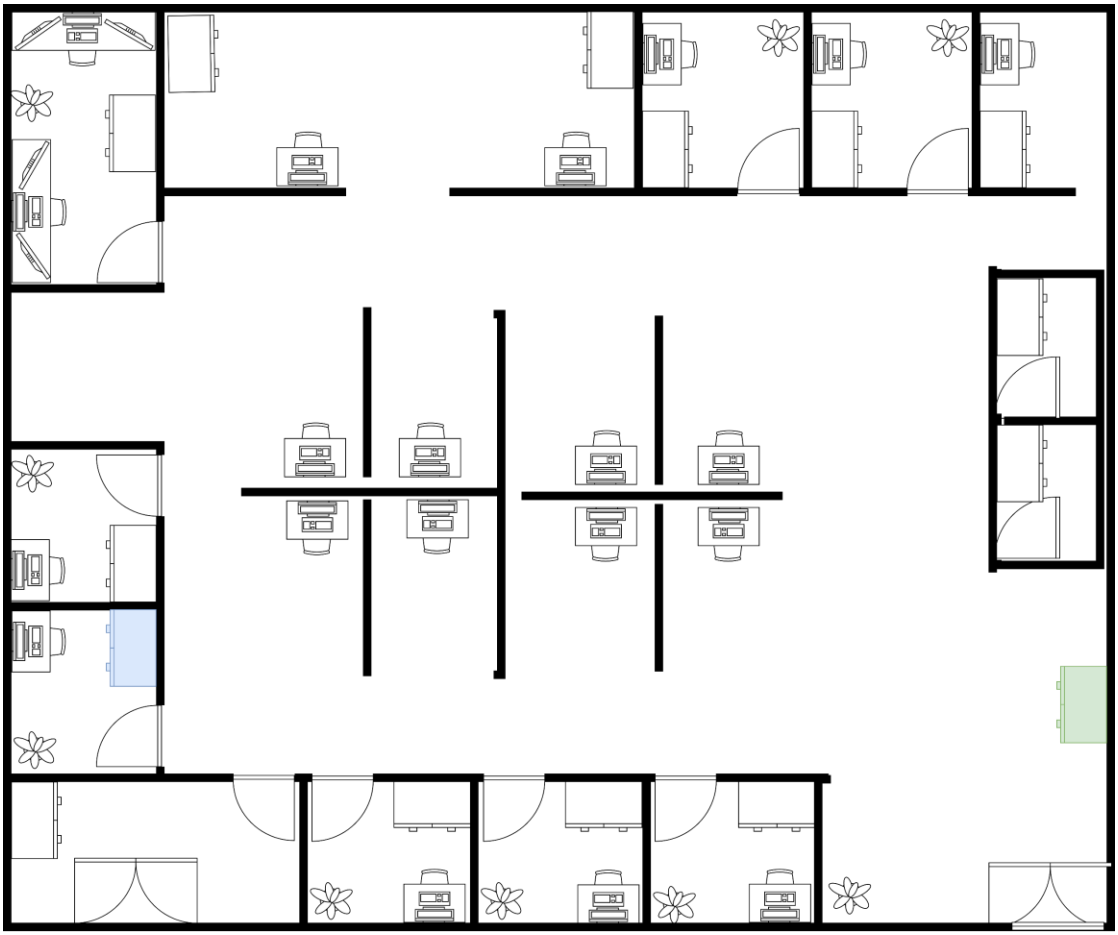


Figure 6.5: The floor plan for laboratory experiments used for node placement.

Once the initial configurations were verified, the nodes were then placed in their respective positions as illustrated by the Figure 6.5. The initial distance between nodes were 20 meters for all device lifetime experiments, whereas the distance later increased in next set of experiments to record the effect of adjusting transmission power on the device range. Furthermore, as previously described in section 5.4.5, a 230mAh Panasonic coin-cell battery was used to power the devices. The hardware configuration and devices used in these experiments are presented in Figure 6.6.

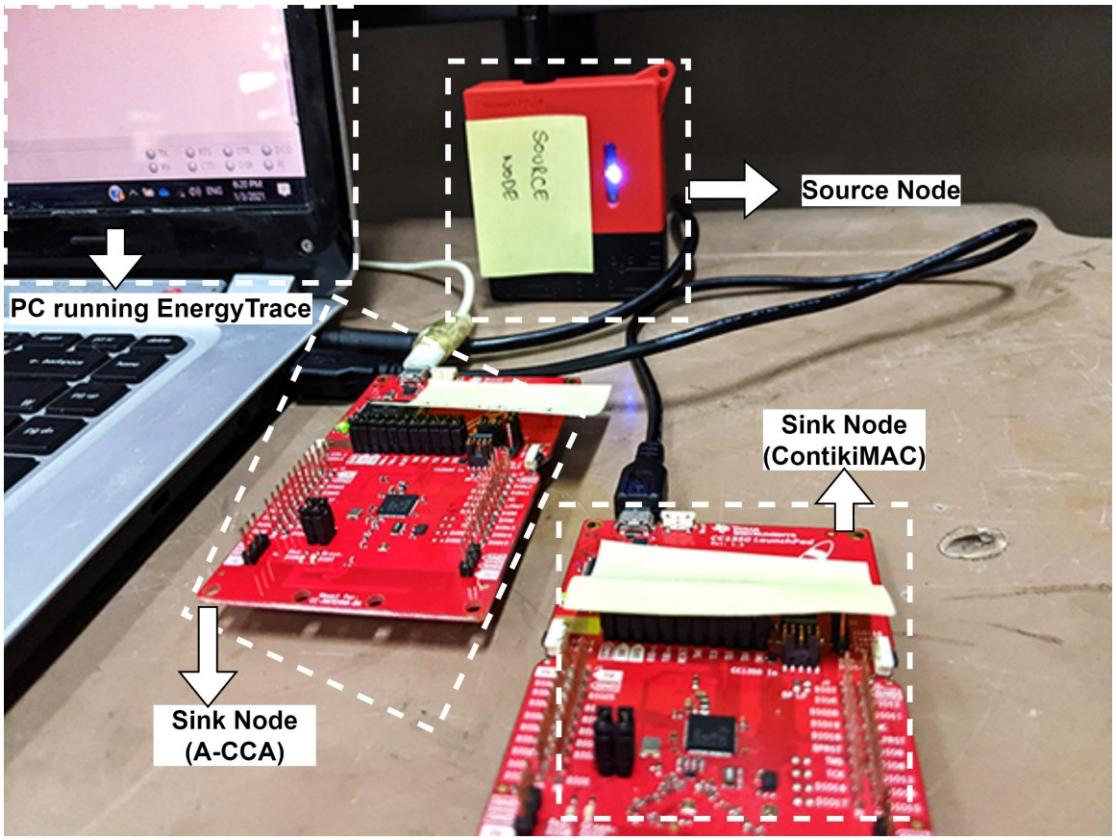


Figure 6.6: CC1352R boards configured with MAC protocols for laboratory experiments

### 6.4.1 A-CCA and its impact on device lifetime

In next of observations, a set of experiments were designed to investigate the effect of A-CCA on the device lifetime. Two CC1352R devices were connected in a point-to-point fashion that communicated using network and node configurations mentioned in the previous section. The devices were powered by similar Panasonic CRC 230 mAh Lithium-ion battery to compare the real-world performance with the simulations as described previously. The source nodes transmitted a constant stream of data at 32 pkt/s which was collected by sink nodes connected to workstation. Once the communication link was established, the experiments were performed until the onboard batteries were depleted to their cut-off voltages (identified on-board programmed led). The recorded values were used to compute the average power consumption and thus the overall node lifetime. These values are then compared with the simulation results described in Table 6.6. Furthermore, the recorded data also enabled to investigate the

effect of A-CCA on packet reception and radio-on states, which is later described in section 6.4.3. Table describes the recorded device lifetime during these experiments.

Table 6.7:Node lifetime estimation of proposed A-CCA scheme.

| | **Lifetime (months)** | | |
|---|---|---|---|
| **Node** | **ContikiMAC** | **A-CCA** | **Average change (%)** |
| Source | 2.03 | 1.97 | 2.96 ↓ |
| Sink | 3.02 | 3.45 | 12.46 ↑ |

The results demonstrated that a significant variation in real-world on-device values is observed as compared to the simulations. The source node running A-CCA experienced an overall decline of 2.96 % in device lifetime as compared to the ones running ContikiMAC. This decline in device lifetime is associated with several internal as well as external physical parameters. For instance, during the simulations, the network environment was assumed to have no temperature effect (i.e., operating at room temperature or 25 ˚C). The proposed battery SOC estimation algorithm utilized the room temperature to estimate the battery voltage, and capacity and therefore recorded no drift that could affect the performance. Whereas the devices were operated in the laboratory settings that resulted in temperature gradients during on and off-working hours throughout this study. This variation may impact the battery performance and thus its lifetime. Additionally, the presence of other radio signals such as WiFi, as well as the presence of metal objects also adds to the interference resulting in increased noise. The added increased noise levels triggered the proposed algorithm to increase the transmit power which also resulted in increased power consumption. This resulted in a combined effect of reduced device lifetime in comparison to the simulation. However, comparing the simulated and actual values of A-CCA revealed a marginal variation in device lifetime that validates the accuracy of the proposed scheme in energy consumption estimation.

The sink nodes running A-CCA algorithm demonstrated a significant increase in its lifetime as compared to the node running ContikiMAC. A significant contributing factor is the nature of the operations performed by these nodes. As the sink nodes are only collecting data, the algorithm mostly powers down the radio by operating in low power listening mode. Additionally, because A-CCA algorithm reduces the CCA checks, the frequent wakeup radio calls to check channel for activity is also reduced, resulting in less power consumption. Furthermore, as the source nodes adjusts its power, the neighboring nodes (sink node in this case) automatically adjusts their power and CCA acknowledgement in accordance resulting in fewer re-transmission. Overall, with CCA and power adjustments, together with accurate battery energy estimation using proposed GPR algorithm, A-CCA established an increase in device lifetime. ContikiMAC follows its default configuration (by using 20 CCA symbol periods) and linear energy estimation resulting in an early cut-off voltage, thus, end of device lifetime.

### 6.4.2 Transmission range evaluation

A-CCA improves radio duty cycle and reduces false wakeups by adjusting transmit power levels so that signals can be detected easily above the noise floor. The standard transmission initially sets the radio to operate at (0dbm), which uses more power. However, because the radio saves energy by spending more time in low power mode and avoiding false wakeups, the overall power rating improves. If the RSSI threshold indicates a less noisy channel, the algorithm adjusts the transmit power (-5 dbm). Depending on where the radios are located, an increase in transmit power may have a variety of effects. Increased transmit power typically results in increased transmission range in outdoor deployments with a direct line of sight. Increased transmission power may result in increased interference in dense network deployments,

particularly in indoor environments with other radio and device interferences (such as WiFi signals, Bluetooth devices, and microwave use) that use the similar 2.4 – 2.5 GHz frequency. The radio may achieve an increased transmission range in the presence of increased interference but decoding the transmitted packets may suffer significant inaccuracies and an overall reduction in received packet rates (PRR).

Therefore, two Texas Instruments CC1352R LaunchPad boards are placed in an office environment during working office hours. The devices are configured to transmit data packets (127 bytes length) in a one-to-one fashion, where the receiver is fixed to a PC terminal running Texas Instrument's Code Composer Studio's EnergyTrace module to detect several incoming packets with RSSI values for each communication. The sender module is mobile, therefore, used to test the transmission range in these experiments. As shown in Figure 6.7, the variation in transmit power has a significant impact on both the transmission range and the measured PRR value. An increase in transmit power may also increase signal noise, which can add to the channel noise floor (due to external radio interferences) and reduce overall SNR values. A lower SNR value reduces the IEEE 802.15.4 receiver's ability to decode the information in received packets. The decrease in SNR and PRR values reduces the link's overall performance. ContikiMAC uses a transmit power of ($P_T = 0$ dbm) in its default configuration, whereas A-CCA uses a transmit power of ($P_T = -5$ dbm). For these experiments, the packet transmission rate was set to 32 packets per second. The distance between the nodes was increased in 30m increments, and the associated PRR was measured for 5 minutes at each step increase (in the distance). The results show that, at first, A-CCA behaves similarly to ContikiMAC, while significantly reducing transmission range and PRR near the transmission cut-off point.

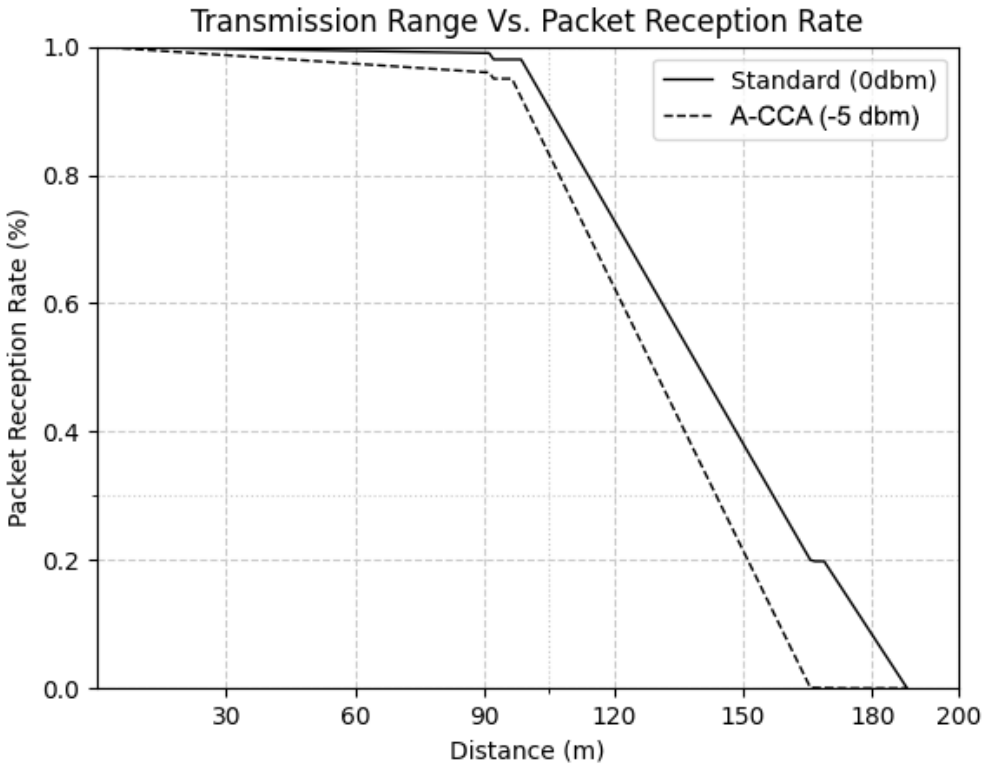



Figure 6.7:Transmission link distance and measured packet reception rate (with default ContikiMAC values and with transmit power variations in A-CCA)

To decode the transmitted information in typical low-power wireless communications, a wireless link quality of more than 85 percent is preferred. Both schemes perform admirably while maintaining a PRR of more than 90% and transmission ranges of 95 and 100 m, respectively. However, with each subsequent increase in distance, the PRR drops to 80%. The link quality degrades significantly as the distance increases, with the PRR dropping to 50% for transmission ranges of (125m and 140m) for A-CCA and ContikiMAC, respectively. Beyond this point, where the PRR of the overall link falls below 50%, it is assumed that the network has reached a cut-off point because the received packet cannot be correctly decoded. Therefore, it is concluded that A-CCA implementation results in a nearly 15% reduction in link transmission range. The reduced transmission range is a significant disadvantage of A-CCA, as it can limit the network's overall transmission range. It can also be argued that because of the reduced transmission range, a sparse network configuration may result

in network fragmentation; thus, the scheme is best suited for small to medium-sized networks.

However, because the proposed scheme is intended for low data-rate WSN applications, the routing aspects are outside the scope of this work. Furthermore, by lowering the transmit power, the radio current consumption drops from (18mA to 14mA), contributing to an increase in device lifetime. However, if the network is not fully converged and the nodes are forced to be placed at a greater distance, the reduction in radio transmission range will have drawbacks. Another important factor to consider is the adaptive nature of the proposed algorithm for adjusting radio transmit power. For example, if the WSN nodes are far apart, the RSSI between two directly connected nodes is automatically affected. A lower RSSI indicates that the network should increase its power, so the transmit powers are reset to 0 dbm and only CCA adjustments are made on channel activity. This ensures that, while the transmission range may be reduced initially, the algorithm compensates for this loss as soon as the RSSI threshold is reduced, reducing the likelihood of network holes or segmented networks.

### 6.4.3 The impact of A-CCA on packet reception and radio state

The ContikiMAC protocol is intended to save energy by duty cycling the radio into low-power listening modes for the majority of its operation. For packet transmission and reception, the channel is sensed via CCA. CCA requires a minimum symbol period of eight symbols to effectively identify channel traffic according to IEEE 802.15.4; however, different implementations include additional CCA symbols for improved detection. ContikiMAC scales CCA strobes up to 20 symbols on average in its basic setup. ContikiMAC nodes keep the radio powered off and periodically listen for incoming packets once every duty cycle to save power. Transmitters send a strobe sequence back-to-back to the recipient, listening for an acknowledgement in between

each packet. A-CCA maintains the minimum defined (8 symbols) interval after each packet reception, followed by two successive CCA to guarantee that the transmission has completed or not, as illustrated by Figure 6.8. This reduces the amount of CCAs while also lowering the frequency of false wakeups. A phase lock (PL) implementation also helps to optimize the protocol by learning the wakeup schedules of surrounding nodes, which helps to prevent false wakeups during radio sleep mode.

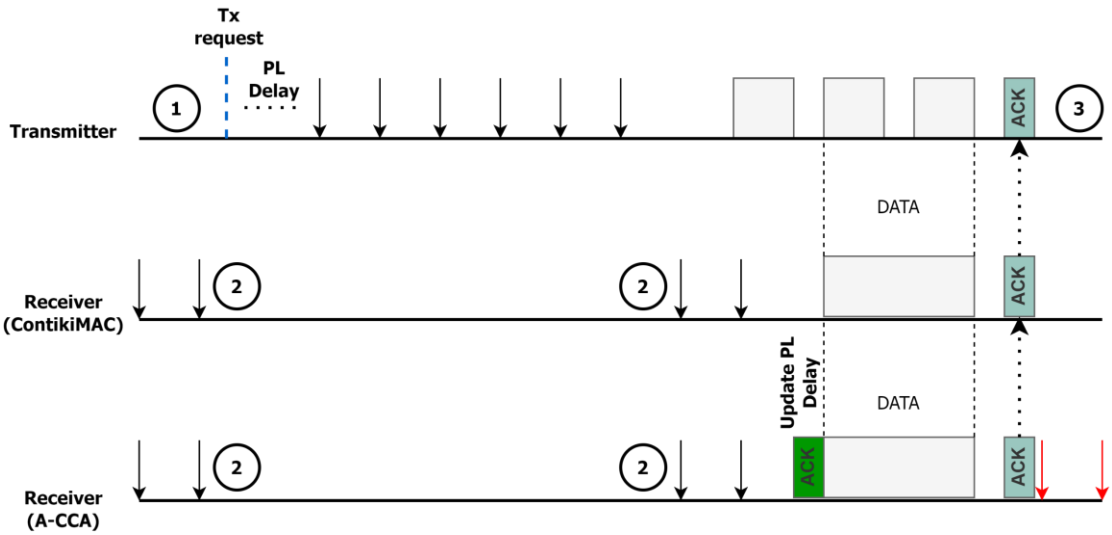



Figure 6.8: Standard ContikiMAC Versus A-CCA: 1) Tx request preceded by PL delay and CCA strobes to show transmission intent. 2) Periodic CCA checks to detect any incoming packets. 3) Data completion acknowledgements, resulting in no further CCA strobes.

A-CCA extends the PL function to detect a regular transmission attempt over the channel noise floor by monitoring changes in surrounding node transmission power levels. As illustrated in Figure 6.8, A-CCA algorithm achieves the improved energy conservation by:

- **Transmitter side adjustments:** The transmitter strobes the channel with CCA showing the intent to transmit. Furthermore, by detecting the presence of interference (through RSSI), it adjusts its Tx power for improved reception.

- **Receiver side adjustments:** The receiver reduces the CCA to 8 symbols and adjusts its wakeup delay based on the PL of transmit. In addition, it also acknowledges the updated Tx power levels of its neighboring nodes, thus differentiates between channel interference and actual transmissions. Finally, the receiver adds two additional CCAs at the end of a successful transmission to notice whether there are further transmission attempts from the transmitter. In case no further attempts are detected, the radio is put to fast-sleep mode and only wakeups after the transmitter's PL interval have lapsed.

In these evaluations, the standard ContikiMAC implementation is compared against the proposed A-CCA scheme for Packet Reception Rate (PRR) and Radio-on Time Per Packet Received (RoT) metrics. PRR helps to understand the radio and link quality behavior while maintain a specific committed communication, whereas RoT identifies the total amount of a radio is powered on during successful packet transfer. As radio in its RX and TX states consumes most amount of power, RoT metrics helps to establish the overall change in power consumption during successful network operation. The link performance and energy efficiency of ContikiMAC and A-CCA are measured in the first experiment under controlled interference conditions.

Two CC1352R boards were placed 15 meters apart in opposite corners of the laboratory. The sink node, which is connected to a host computer via USB, is used to detect the number of received packets and signal threshold levels. The transmitter node, sends 90-byte packets at a rate of 5 packets per second. The readings were recorded after every minute and the overall experiment during was set of 2.5 hours for both the schemes. During working hours, the laboratory environment is heavily used, with IEEE

802.11g access points, an IEEE 802.11n gateway, and a pantry microwave contributing to the noise that resembles IEEE 802.15.4 packet packets. The purpose of these tests was to look at node behaviour in indoor office settings with fixed node lengths. The nodes might be designed to use the most appropriate channel with the least amount of interference. Channel estimation and switching were not used since adaptive channel management is beyond the scope of this study. This also allowed for a more realistic simulation of the nodes, as network conditions can alter throughout peak and off-peak office hours.

Figure 6.9 reveals the results obtained from the experiments under natural IEEE 802.11 interference. ContikiMAC struggles to maintain the PRR as the incoming stream of packets progress through the channel with interference. As the standard ContikiMAC has no interference coping mechanism, the active channel interference threshold is normally configured while programming the nodes and therefore remains fixed. This results in loss of packets and re-transmission in the presence of the noise. Another consideration is the implementation strategy of the standard scheme where CCA are fixed, and the transmitter strobes the channel continuously before transmitting data. In the presence of network interference, the CCA results negative, that further delays the transmission resulting in more strobes and thus reduced PRR. It was observed that on average, the PRR for ContikiMAC in a busy environment falls below 50% which indicates a poor metric for link quality in its standard configuration.

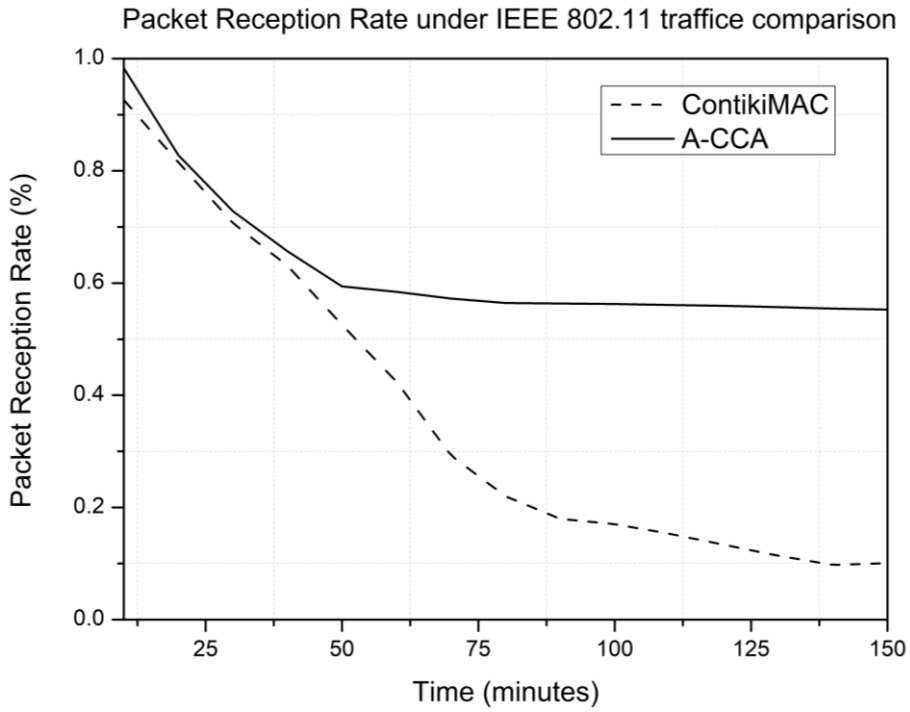


Figure 6.9: Packet reception rate comparison under IEEE 802.11 based network interference.

Whereas, A-CCA is adaptable, where Tx adjusts its radio power, whereas the Rx synchronizes its wakeup times, acknowledges the transmit power levels of neighboring nodes, and adds additional CCA checks to avoid re-transmission. This ensures minimal number of re-transmissions, which helps to improve the PRR. It was observed that A-CCA maintained a PRR of above 50% throughout these experiments, indicating a better and usable link quality. Furthermore, by contrasting the average received PRR values, it was observed the A-CCA performed over 70% better in contrast to standard ContikiMAC implementation.

In the next experiment, the network interference is maintained in a controlled environment using Distributed Internet Traffic Generator (D-ITG) as described in (Avallone, Guadagno, Emma, Pescape, & Ventre, 2004). An IEEE 802.11n gateway device is attached to a PC using ethernet connection which is controlled by D-ITG to program 90 bytes length packets at various random intervals. This programmed gateway served as the controlled interference source, which demonstrates the impact of

interference on WSN nodes in an office environment. However, the network configurations for both source and sink nodes and rest of the experiment configurations were unchanged.

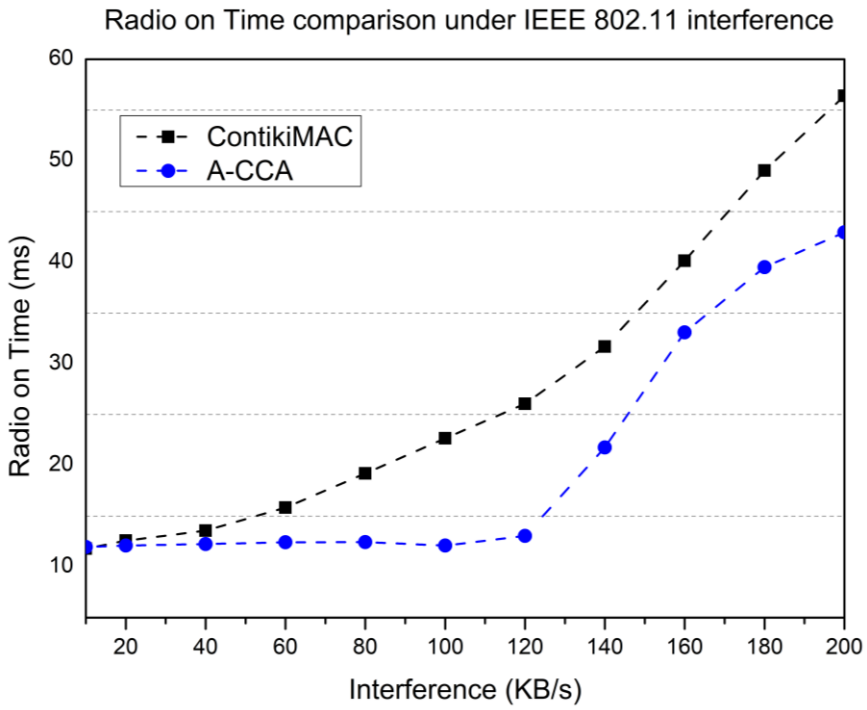



Figure 6.10: Radio on Time analysis under IEEE 802.11 interference.

The radio-on time per packet received grows proportionally with the interference rate, as seen in Figure 6.10. This is because as the amount of interference grows, the PRR decreases, increasing the amount of energy expended each packet. In addition, erroneous wake-ups are increasing, which leads to more idle listening. In both circumstances, the IEEE 802.15.4 minimum guard time is maintained, resulting in additional delays that keep the radio switched on. The transmitter strobes the channel incessantly and, due to interference, struggles to find an empty slot, remaining switched on. The receiver performs differently in both strategies. The receiver wakes up after the required interval in a conventional ContikiMAC implementation and probes for any incoming transmission activity. The receiver fails to register any activity and falls back

to sleep since the transmitter is unable to send the packet due to channel interference. Therefore, a pattern of false wakeups persists, consuming additional power.

A-CCA receivers synchronize their wakeups with surrounding transmitters. In addition, as the transmitter raises its radio power above the noise level, the receiver recognizes the power variation (owing to changes in RSSI values) and initiates and completes the packet transfer. Therefore, the radio stays on for shorter periods of time, conserving even more electricity. However, as radio interference intensifies, both schemes struggle to keep their radios on-air, using more power in the process. In compared to normal ContikiMAC implementation, A-CCA was shown to have a 25.19 percent lower ROT, as well as a considerable improvement in energy conservation, resulting in a longer WSN node lifespan.

## 6.5 Summary

In this chapter A-CCA scheme was proposed to improve the device lifetime while also improving the PRR as well as energy conservation for most of the settings. In addition, the integration of the proposed scheme with existing MAC protocol that supports IEEE 802.15.4 protocol standard using CCA was also explained. The results concluded that the proposed scheme maintains a consistent PDR and PRR rates without incurring additional costs toward its duty cycle, that may consume additional power. The laboratory experiments on the standard ContikiMAC implementation against A-CCA was compared and it was found that the proposed scheme extends the node lifetime by carefully managing CCA, transmit power, reduced retransmission, and ROT. It was found that using A-CCA, the ROT was reduced by 25.19% that helped to reduced node's energy consumption and thus an increase in 12.46% of node lifetime, in months.

# CHAPTER 7

# CONCLUSION AND FUTURE RECOMMENDATIONS

This dissertation focused on the problem of reducing energy consumption to prolong the lifetime of WSN. Even though this problem can be solved in various ways, this research focused on cross-layer optimization techniques at the PHY and MAC layers. The proposed solution is cognizant of several WSN constraints typically including limited on-board resources, and the impact of environmental factors to name a few. The contributions of this study are provided in Section 7.1 that describes how the research objectives were achieved, whereas the future recommendations are discussed in Section 7.2.

## 7.1 Conclusion

It is imperative to establish the available battery energy for WSN node power estimation calculations. This led to the design of the proposed technique for battery characterization based on typical WSN node traffic, to achieve the first objective. Batteries suffer from capacity losses due to passivation and recovery effect. It was observed that with the proposed battery pre-conditioning, up to 4% increase in capacity was possible. Furthermore, under extended laboratory experiments, a relaxation period of 8-hours reported an overall regain of 10% battery capacity. Additionally, the capacity approximation errors were investigated using the OCV and Ah methods at various operating temperatures to simulate a real-world scenario. It was discovered that OCV is more accurate, but it takes more time and resources. This limits the use of the OCV method for real-time applications on devices with limited resources. As a result, there is almost always a trade-off between accuracy and performance. Finally, the effect of load current on battery temperature was investigated, with typical smaller WSN currents

having no discernible effect on battery surface temperatures. Lithium-ion batteries provided higher voltages and larger flat regions across all discharge rates, according to the findings. Furthermore, higher discharge rates had no thermal effect on Lithium-based batteries, making these batteries the best choice for WSN applications.

The battery characterization that yielded an extended dataset led to the design of battery SOC estimation algorithm. Machine learning algorithms are very sensitive to its input (the dataset), its transformation and correlation. It was observed that correlation between features must be performed to select the best performing data transformation. Similarly, the impact of data split ratios on model accuracy was investigated. It was observed that an 80-20% data split ratio yielded the best performing models for every dataset. Finally, extensive research on various ML techniques was conducted to investigate the impact of data variables and external factors on model accuracy and predictability. It was observed that the most common ML techniques including FFNN, and SVM struggled to maintain accuracy over different battery datasets. Therefore, a lightweight GR-based model was developed that reported an overall accuracy of more than 98 percent for all the batteries under consideration. In addition, a novel code structure was created to deploy the GPR SOC estimation model on ARM Cortex-based platforms. Finally, the second research objective was achieved by performing real-time on-device SOC estimation using the proposed technique. It was discovered that the proposed technique was both lightweight, and accurate resulting in precise SOC estimations without incurring additional computational costs which is critical for conserving energy and thus improving the lifetime of WSN node.

Due to the dynamic and ever changing WSN applications deployment, it is critical to design large-scale networks that help to simulate and assess various operational conditions. In this regard, the performance validation was performed on real

hardware, as well as Contiki OS simulator, to emulate several energy estimation calculations that led to WSN lifetime forecasting. First, Contiki OS's default node energy estimation model was used to compare two mathematically modelled schemes, KiBAM and TKiBAM, and later to validate the performance of the proposed GPR-based scheme. When compared to the linear model presented by Contiki OS in its standard implementation, the mathematically modelled schemes provide a more accurate estimate of device remaining energy. Although Contiki OS's default implementation appeared to be the simplest, it had several shortcomings. For instance, its linear behaviour and inability to account for different battery chemistries and environmental factors such as temperature severely limits its operation and result accuracy. The kinetic-based models performed significantly better; however, these techniques require detailed parameter extraction which may eventually necessitate extensive laboratory testing. The proposed GPR-based scheme helped to the achieve the third research objective, that provided accurate SOC estimation for all battery types across a wide temperature range. A thorough examination of TKiBAM and the proposed scheme revealed that TKiBAM exhibits variability in simulation and experimental observations. Furthermore, the GPR-based scheme had a minor computational overhead while providing accurate SOC estimation. Finally, the degree of variation between simulated and laboratory experiments was found to be marginal, which is critical when designing an energy aware MAC protocol, A-CCA.

The techniques described above enabled accurate SOC estimation, which led to the design of energy-conserving A-CCA protocol. The A-CCA protocol served two purposes: it reduced the amount of CCA used for channel activity and it reduced packet re-transmission, which was especially important in a noisy environment. These two approaches work together to reduce radio wakeup times and re-transmissions, which

saves a significant amount of power. The adaptive algorithm first strobes the channel to measure channel activity, and the LQI and RSSI parameters aid in analysing channel state and link quality. The algorithm then adjusts the transmit radio power over the channel noise floor, making Tx packet recognition easier. The receiver recognises the transmitter's PL delay period as well as the increase in Tx power. This allows the receiver to update its wakeup period while also reducing power consumption because the stronger neighbouring signal is easier to detect. Re-transmissions are further reduced as the algorithm locks the wakeup period. Furthermore, at the end of each successful transmission, the receiver sends two additional CCA to determine whether the transmitter still intends to transmit, and if not, the microcontroller enters the fast sleep state. The laboratory tests of the A-CCA protocol in comparison to the standard ContikiMAC protocol revealed a 12.46 % increase in source lifetime, thus achieving the final research objective. This translates to an additional 5 days of node lifetime, which can be significant and lifesaving in mission critical WSN applications such as wildfire and tsunami detection. The study also found that by using adaptive radio power throttling in a noisy environment like an office or workspace, A-CCA based nodes demonstrated improved PRR. Furthermore, the overall radio on-time (RoT) was reduced by a staggering 25.19 %, aiding in energy conservation, and increasing node lifetime. The simulation results along with the actual real-world on-device investigation validate the proposed techniques and exhibit significant performance improvements in terms of accurate SOC estimation, enhanced node energy detection, improved PRR, and an overall increase in node lifetime.

## 7.2 Recommendations for Future Work

Battery cell design, characterization, SOC estimation, and MAC optimization are such broad areas of active research, most of the topics fell outside the scope of this study and did not align with the research objectives. Therefore, the following related research issues can be investigated in future work, including:

### 7.2.1 Estimating viability of advance battery chemistries

Although four battery chemistries were characterized that are commonly used in portable electronics, the current research trend is focused on new innovative technologies. In the research findings, Ni-MH batteries provided a flatter profile and a higher degree of accuracy on ML models. However, because of its extremely high self-discharge and inability to fully utilise its available capacity, Li-Ion, and Li-Ion polymer-based batteries were further investigated. Li-Ion research has almost come to a halt because there isn't enough room for further advancements. Because of their exceptional energy density, the scientific community is currently focusing on Li-metal, metal-sulfur, and metal-air batteries. With increased energy density, smaller, lighter, more cost-effective, and environmentally friendly batteries can be manufactured, making them ideal for WSN applications.

### 7.2.2 Calculating the effect of sub-zero temperature on performance

The current research examined the thermal effect on batteries caused by environmental factors as well as internal thermal connectivity that varies depending on load. Even though this proved to be significant measures to model the battery behaviour for the majority of WSN deployment scenarios. WSN nodes are frequently deployed in colder environments where the temperature falls below zero degrees Celsius. Batteries are commonly known to underperform in such environments. Therefore, future

extensions of this work will include battery characterization for WSN applications operating in sub-zero temperatures.

### 7.2.3 Modelling charging profiles for energy-harvesting applications

Only battery discharging behaviour was modelled and used for SOC estimation in this study, as is typical for WSN applications. Almost all WSN applications define the end of node lifetime as the battery dying. However, in recent years, energy harvesting using solar, wind, vibration, and magnetic sensors has opened up new research avenues. Because WSN applications use smaller batteries, a secondary energy harvesting source can extend their life for years. Therefore, future research will concentrate on the development of optimal battery charging algorithms. In addition to the discharge profiles, the SOC estimation for charging will be integrated to improve the SOH and lifetime of the batteries.

### 7.2.4 Investigation of A-CCA's impact on Routing layer

Finally, because the proposed A-CCA protocol improves device lifetime through adaptive transmit power adjustments, it will be very interesting to see how it affects the routing layer. Routing protocols, for example, are always looking for the best and most cost-effective route for packet dissemination. In this regard, one common technique is to divide the network into clusters, with cluster heads in charge of packet aggregation and dissemination within clusters. Typically, cluster heads are chosen based on a variety of metrics, one of which is remaining node energy. The effect of A-extension CCA's as transmit powers are adaptively adjusted on battery energy can influence cluster head selection. Therefore, future research must investigate the impact of A-CCA on other layers. It would also be very interesting to use the techniques defined in this research as a framework for future cross-layer optimizations, which may increase device lifetime even further.

## REFERENCES


Adam, M. S., Por, L. Y., Hussain, M. R., Khan, N., Ang, T. F., Anisi, M. H., . . . Ali, I. (2019). An Adaptive Wake-Up-Interval to Enhance Receiver-Based Ps-Mac Protocol for Wireless Sensor Networks. *19*(17), 3732.

Ahmad, A., & Hanzálek, Z. (2017). Distributed Real Time TDMA Scheduling Algorithm for Tree Topology WSNs. *IFAC-PapersOnLine, 50*(1), 5926-5933. doi:https://doi.org/10.1016/j.ifacol.2017.08.1484

Ahmad, A. B., Ooi, C. A., Ishak, D., & Teh, J. (2019). State-of-Charge Balancing Control for ON/OFF-Line Internal Cells Using Hybrid Modular Multi-Level Converter and Parallel Modular Dual L-Bridge in a Grid-Scale Battery Energy Storage System. *IEEE Access, 7*, 131-147. doi:10.1109/ACCESS.2018.2885083

Ahmeid, M., Muhammad, M., Lambert, S., Attidekou, P. S., & Milojevic, Z. J. J. o. E. S. (2022). A rapid capacity evaluation of retired electric vehicle battery modules using partial discharge test. *50*, 104562.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. J. C. n. (2002). Wireless sensor networks: a survey. *38*(4), 393-422.

Akyildiz, I. F. a. V., M.C. (2010). Physical Layer. In *Wireless Sensor Networks* (pp. 53-76).

Al-Obaidy, F., Momtahen, S., & Mohammadi, F. (2019, 5-8 May 2019). *Wireless Sensor Networks Analysis based on MAC Protocols*. Paper presented at the 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE).

Alfayez, F., Hammoudeh, M., & Abuarqoub, A. (2015). A Survey on MAC Protocols for Duty-cycled Wireless Sensor Networks. *Procedia Computer Science, 73*, 482-489. doi:https://doi.org/10.1016/j.procs.2015.12.034

Ali, O., Ishak, M. K., & Bhatti, M. K. L. (2021). Adaptive clear channel assessment (A-CCA): Energy efficient method to improve wireless sensor networks (WSNs) operations. *AEU - International Journal of Electronics and Communications, 131*, 153603. doi:https://doi.org/10.1016/j.aeue.2020.153603

Ali, O., Ishak, M. K., Bhatti, M. K. L. J. A.-I. J. o. E., & Communications. (2021). Adaptive clear channel assessment (A-CCA): Energy efficient method to improve wireless sensor networks (WSNs) operations. *131*, 153603.

Alvi, A. N., Bouk, S. H., Ahmed, S. H., Yaqub, M. A., Javaid, N., & Kim, D. (2015). Enhanced TDMA based MAC protocol for adaptive data control in wireless sensor networks. *Journal of Communications and Networks, 17*(3), 247-255. doi:10.1109/JCN.2015.000046

Amirinasab Nasab, M., Shamshirband, S., Chronopoulos, A. T., Mosavi, A., & Nabipour, N. (2020). Energy-Efficient Method for Wireless Sensor Networks Low-Power Radio Operation in Internet of Things. *9*(2), 320.

Anastasi, G., Conti, M., Di Francesco, M., & Passarella, A. J. A. h. n. (2009). Energy conservation in wireless sensor networks: A survey. *7*(3), 537-568.

Andrea, D. (2020). *Lithium-Ion Batteries and Applications: A Practical and Comprehensive Guide to Lithium-Ion Batteries and Arrays, from Toys to Towns, Volume 2, Applications* (Vol. 2): Artech House.

Arisha, K., Youssef, M., & Younis, M. (2002). Energy-aware TDMA-based MAC for sensor networks. In *System-level power optimization for wireless multimedia communication* (pp. 21-40): Springer.

Avallone, S., Guadagno, S., Emma, D., Pescape, A., & Ventre, G. (2004, 27-30 Sept. 2004). *D-ITG distributed Internet traffic generator.* Paper presented at the First International Conference on the Quantitative Evaluation of Systems, 2004. QEST 2004. Proceedings.

Basumatary, H., & Debbarma, M. K. J. I. T. R. (2021). Comparative Study on Different Parameters Used for Energy Conservation in Wireless Sensor Networks. 1-13.

BenSaleh, M. S., Saida, R., Kacem, Y. H., & Abid, M. J. J. o. S. (2020). Wireless sensor network design methodologies: A survey. *2020*.

Bhatia, A., & Hansdah, R. C. J. A. (2020). A Classification Framework for TDMA Scheduling techniques in WSNs. *abs/2002.00458*.

Bhatia, T., Kansal, S., Goel, S., Verma, A. J. C., & Engineering, E. (2016). A genetic algorithm based distance-aware routing protocol for wireless sensor networks. *56*, 441-455.

Bhattacharjee, A., Verma, A., Mishra, S., & Saha, T. K. J. I. T. o. V. T. (2021). Estimating State of Charge for xEV Batteries Using 1D Convolutional Neural Networks and Transfer Learning. *70*(4), 3123-3135.

Bohlen, O. (2008). *Impedance-based battery monitoring*: Shaker.

Brownlee, J. (2018). *Better deep learning: train faster, reduce overfitting, and make better predictions*: Machine Learning Mastery.

Buchli, B., Aschwanden, D., & Beutel, J. (2013). *Battery state-of-charge approximation for energy harvesting embedded systems.* Paper presented at the European Conference on Wireless Sensor Networks.

Buettner, M., Yee, G. V., Anderson, E., & Han, R. (2006). *X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks*. Paper presented at the Proceedings of the 4th international conference on Embedded networked sensor systems, Boulder, Colorado, USA.

Cabezas, A. C., & Patrón, J. J. S. (2018). Simulation evaluation of TMAC: A media access protocol for WSN. *ARPN Journal of Engineering and Applied Sciences, 13*(23), 9145-9150.

Chaari, L., & Kamoun, L. J. a. p. a. (2010). Wireless sensors networks MAC protocols analysis.

Chandran, V., K Patil, C., Karthick, A., Ganeshaperumal, D., Rahim, R., & Ghosh, A. J. W. E. V. J. (2021). State of charge estimation of lithium-ion battery for electric vehicles using machine learning algorithms. *12*(1), 38.

Chaoui, H., & Mandalapu, S. J. B. (2017). Comparative study of online open circuit voltage estimation techniques for state of charge estimation of lithium-ion batteries. *3*(2), 12.

Chen, K., & Li, X. J. J. o. P. S. (2014). Accurate determination of battery discharge characteristics–A comparison between two battery temperature control methods. *247*, 961-966.

Chen, M., & Rincon-Mora, G. A. J. I. t. o. e. c. (2006). Accurate electrical battery model capable of predicting runtime and IV performance. *21*(2), 504-511.

Chen, R.-C., Dewi, C., Huang, S.-W., & Caraka, R. E. (2020). Selecting critical features for data classification based on machine learning methods. *Journal of Big Data, 7*(1), 52. doi:10.1186/s40537-020-00327-4

Chin, C. S., Gao, Z., & Zhang, C. Z. J. J. o. E. S. (2020). Comprehensive electro-thermal model of 26650 lithium battery for discharge cycle under parametric and temperature variations. *28*, 101222.

Chincoli, M., & Liotta, A. (2018). Transmission Power Control in WSNs: From Deterministic to Cognitive Methods. In R. Gravina, C. E. Palau, M. Manso, A. Liotta, & G. Fortino (Eds.), *Integration, Interconnection, and Interoperability of IoT Systems* (pp. 39-57). Cham: Springer International Publishing.

Chincoli, M., Syed, A. A., Exarchakos, G., & Liotta, A. J. M. I. S. (2016). Power control in wireless sensor networks with variable interference. *2016*.

Corporation, A. P. P. (2015). *Ni-mh Rechargeable battery specifications*. Retrieved from Battery Space: https://www.batteryspace.com/prod-specs/174-MH-AAA1000APZ.pdf

Correia, L. H., Macedo, D. F., dos Santos, A. L., Loureiro, A. A., & Nogueira, J. M. S. J. C. N. (2007). Transmission power control techniques for wireless sensor networks. *51*(17), 4765-4779.

da Cunha, A. B., & da Silva, D. C. J. I. L. A. T. (2012). Behavioral model of alkaline batteries for wireless sensor networks. *10*(1), 1295-1304.

De, D., Song, W.-Z., Xu, M., Shi, L., & Tan, S. (2013). Chapter 1 - Advances in Real-World Sensor Network System. In A. Hurson (Ed.), *Advances in Computers* (Vol. 90, pp. 1-90): Elsevier.

Deng, D. (2015). Li-ion batteries: basics, progress, and challenges. *3*(5), 385-418. doi:https://doi.org/10.1002/ese3.95

Deng, Z., Hu, X., Lin, X., Xu, L., Che, Y., & Hu, L. J. I. A. T. o. M. (2020). General discharge voltage information enabled health evaluation for lithium-ion batteries. *26*(3), 1295-1306.

Ding, X., Zhang, D., Cheng, J., Wang, B., Chai, Y., Zhao, Z., . . . Luk, P. C. K. (2020). A Novel Active Equalization Topology for Series-Connected Lithium-ion Battery Packs. *IEEE Transactions on Industry Applications, 56*(6), 6892-6903. doi:10.1109/TIA.2020.3015820

Dinh, T., Kim, Y., Gu, T., & Vasilakos, A. V. J. C. n. (2016). L-MAC: A wake-up time self-learning MAC protocol for wireless sensor networks. *105*, 33-46.

DPTL. (2017). Polymer Li-Ion Rechargeable Battery. Retrieved from https://cdn.sparkfun.com/assets/5/6/e/1/5/SPE-DTP603450-1000mah-3.7V-En-1.0V.pdf

Duan, R., Zhao, Q., Zhang, H., Zhang, Y., & Li, Z. (2017). Modeling and performance analysis of RI-MAC under a star topology. *Computer Communications, 104*, 134-144. doi:https://doi.org/10.1016/j.comcom.2017.01.004

Dunkels, A. (2011). The contikimac radio duty cycling protocol.

Dunkels, A., Osterlind, F., Tsiftes, N., & He, Z. (2007). *Software-based on-line energy estimation for sensor nodes.* Paper presented at the Proceedings of the 4th workshop on Embedded networked sensors.

Duryea, S., Islam, S., & Lawrance, W. (1999). *A battery management system for stand alone photovoltaic energy systems.* Paper presented at the Conference Record of the 1999 IEEE Industry Applications Conference. Thirty-Forth IAS Annual Meeting (Cat. No. 99CH36370).

Dutta, R. (2016). Ultra low power and interference robust transceiver techniques for wireless sensor networks.

Effatparvar, M., Dehghan, M., & Rahmani, A. M. J. J. o. m. s. (2016). A comprehensive survey of energy-aware routing protocols in wireless body area sensor networks. *40*(9), 1-27.

Eleftheroglou, N., Mansouri, S. S., Loutas, T., Karvelis, P., Georgoulas, G., Nikolakopoulos, G., & Zarouchas, D. J. A. E. (2019). Intelligent data-driven prognostic methodologies for the real-time remaining useful life until the end-of-discharge estimation of the Lithium-Polymer batteries of unmanned aerial vehicles with uncertainty quantification. *254*, 113677.

Elshabrawy, T. (2015). Network throughput analysis of IEEE 802.15.4 enabled wireless sensor networks with FEC coding under external interference. *AEU - International Journal of Electronics and Communications, 69*(11), 1641-1649. doi:https://doi.org/10.1016/j.aeue.2015.07.017

Energizer. (2014). *LR6 Alkaline Battery*. Retrieved from https://data.energizer.com/pdfs/alk-power-aa.pdf

Ergen, S. C., & Varaiya, P. (2010). TDMA scheduling algorithms for wireless sensor networks. *16*(4 %J Wirel. Netw.), 985–997. doi:10.1007/s11276-009-0183-0

Fan, R.-E., Chen, P.-H., Lin, C.-J., & Joachims, T. J. J. o. m. l. r. (2005). Working set selection using second order information for training support vector machines. *6*(12).

Farahani, S. (2008a). Battery Life Analysis. In S. Farahani (Ed.), *ZigBee Wireless Networks and Transceivers* (pp. 207-224). Burlington: Newnes.

Farahani, S. (2008b). Chapter 3 - ZigBee and IEEE 802.15.4 Protocol Layers. In S. Farahani (Ed.), *ZigBee Wireless Networks and Transceivers* (pp. 33-135). Burlington: Newnes.

Fru, S. E., Tsafack, P., & Tanyi, E. J. J. o. E. S. (2021). The exploitation of open circuit voltage parameters and energy recovery after discharge, to decipher the state of health of lead acid batteries. *44*, 103477.

Fu, Y., Sha, M., Hackmann, G., & Lu, C. (2012). *Practical control of transmission power for wireless sensor networks.* Paper presented at the 2012 20th IEEE International Conference on Network Protocols (ICNP).

Garg, V. K. (2007). CHAPTER 21 - Wireless Local Area Networks. In V. K. Garg (Ed.), *Wireless Communications & Networking* (pp. 713-776). Burlington: Morgan Kaufmann.

Gholizadeh, M., & Salmasi, F. R. J. I. T. o. I. E. (2013). Estimation of state of charge, unknown nonlinearities, and state of health of a lithium-ion battery based on a comprehensive unobservable model. *61*(3), 1335-1344.

Gilani, M. H. S., Sarrafi, I., & Abbaspour, M. (2013). An adaptive CSMA/TDMA hybrid MAC for energy and throughput improvement of wireless sensor networks. *Ad Hoc Networks, 11*(4), 1297-1304. doi:https://doi.org/10.1016/j.adhoc.2011.01.005

Goodfellow, I., Bengio, Y., & Courville, A. J. C. M. (2017). Deep learning (adaptive computation and machine learning series). 321-359.

Gungor, S., Cetkin, E., & Lorente, S. J. A. T. E. (2022). Thermal and electrical characterization of an electric vehicle battery cell, an experimental investigation. 118530.

Gupta, S. K., Kuila, P., Jana, P. K. J. C., & Engineering, E. (2016). Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks. *56*, 544-556.

Hagan, M. T., Demuth, H. B., & Beale, M. (1997). *Neural network design*: PWS Publishing Co.

Hansen, T., & Wang, C.-J. J. J. o. P. S. (2005). Support vector based battery state of charge estimator. *141*(2), 351-358.

He, L., & Guo, D. (2019). An Improved Coulomb Counting Approach Based on Numerical Iteration for SOC Estimation With Real-Time Error Correction Ability. *IEEE Access, 7*, 74274-74282. doi:10.1109/ACCESS.2019.2921105

Heaton, J. (2015). *Artificial Intelligence for Humans: Deep learning and neural networks*: Heaton Research, Incorporated.

Hu, X., Feng, F., Liu, K., Zhang, L., Xie, J., & Liu, B. (2019). State estimation for advanced battery management: Key challenges and future trends. *Renewable and Sustainable Energy Reviews, 114*, 109334. doi:10.1016/j.rser.2019.109334

Idrees, A. K., Abou Jaoude, C., & Al-Qurabat, A. K. M. (2020). *Data reduction and cleaning approach for energy-saving in wireless sensors networks of IoT.* Paper presented at the 2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob).

Instrument, T. (2018). CC1352R SimpleLink High-Performance Multi-Band Wireless MCU (Datasheet). from Texas Instrument https://www.ti.com/lit/ds/swrs196g/swrs196g.pdf?ts=1603646192894&ref_url=https%253A%252F%252Fwww.ti.com%252Ftool%252FLAUNCHXL-CC1352R1

Instrument, T. (2020a). *2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*. Retrieved from Texas Instruments: https://www.ti.com/product/CC2420

Instrument, T. (2020b). *CC1352R SimpleLink™ High-Performance Multi-Band Wireless MCU*. Retrieved from Texas Instruments: https://www.ti.com/product/CC1352R

Instrument, T. (2021). CC1352R Simple Link High-Performance Multi-Band Wireless MCU. Retrieved from https://www.ti.com/lit/ds/swrs196i/swrs196i.pdf?ts=1620476954007&ref_url=https%253A%252F%252Fwww.ti.com%252Ftool%252FLAUNCHXL-CC1352R1

Instruments, T. (2007). 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. In T. Instruments (Ed.). TI Website.

Instruments, T. (2009). Low-Cost Low-Power 2.4GHz RF Transceiver. *1*, 97. Retrieved from Low Cost, Low-Power 2.4 GHz RF Transceiver Designed for Low-Power Wireless Apps in the 2.4 GHz ISM B website: https://www.ti.com/lit/gpn/cc2500

Jagriti, & Lobiyal, D. K. (2018). Energy Consumption Reduction in S-MAC Protocol for Wireless Sensor Network. *Procedia Computer Science, 143*, 757-764. doi:https://doi.org/10.1016/j.procs.2018.10.428

Jaguemont, J., Boulon, L., Venet, P., Dubé, Y., & Sari, A. J. I. T. o. V. T. (2015). Lithium-ion battery aging experiments at subzero temperatures and model development for capacity fade estimation. *65*(6), 4328-4343.

Jain, R. (2020). Comparative analysis of contention based and TDMA based MAC protocols for wireless sensor networks. *International Journal of Information Technology, 12*(1), 245-250. doi:10.1007/s41870-018-0152-x

Jia, J., Wang, K., Shi, Y., Wen, J., Pang, X., & Zeng, J. J. J. o. E. S. (2021). A multi-scale state of health prediction framework of lithium-ion batteries considering the temperature variation during battery discharge. *42*, 103076.

Jin, R., Che, Z., Wang, Z., Zhu, M., & Wang, L. J. I. W. S. S. (2015). Battery optimal scheduling based on energy balance in wireless sensor networks. *5*(6), 277-282.

Jin, Y., Zhao, W., Li, Z., Liu, B., & Wang, K. J. E. R. (2021). SOC estimation of lithium-ion battery considering the influence of discharge rate. *7*, 1436-1446.

Kansal, A., Somasundara, A. A., Jea, D. D., Srivastava, M. B., & Estrin, D. (2004). *Intelligent fluid infrastructure for embedded networks.* Paper presented at the Proceedings of the 2nd international conference on Mobile systems, applications, and services.

Karabulut, E. M., Özel, S. A., & İbrikçi, T. (2012). A comparative study on the effect of feature selection on classification accuracy. *Procedia Technology, 1*, 323-327. doi:10.1016/j.protcy.2012.02.068

Karlsson, J., & Skoglund, M. (2011). *Lattice-based source-channel coding in wireless sensor networks.* Paper presented at the 2011 IEEE International Conference on Communications (ICC).

Kaswan, A., Nitesh, K., & Jana, P. K. (2017). Energy efficient path selection for mobile sink and data gathering in wireless sensor networks. *AEU - International*

*Journal of Electronics and Communications, 73*, 110-118. doi:https://doi.org/10.1016/j.aeue.2016.12.005

Kaur, T., & Kumar, D. (2016, 14-16 Oct. 2016). *TDMA-based MAC protocols for wireless sensor networks: A survey and comparative analysis.* Paper presented at the 2016 5th International Conference on Wireless Networks and Embedded Systems (WECON).

Keshavarzian, A., Lee, H., & Venkatraman, L. (2006). *Wakeup scheduling in wireless sensor networks*. Paper presented at the Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, Florence, Italy. https://doi.org/10.1145/1132905.1132941

Khamayseh, Y. M., Mardini, W., & Halima, N. B. J. J. C. (2018). Evolutionary Algorithm for Scheduling in Wireless Sensor Networks. *13*(3), 262-270.

Kia, G., & Hassanzadeh, A. (2019). A multi-threshold long life time protocol with consistent performance for wireless sensor networks. *AEU - International Journal of Electronics and Communications, 101*, 114-127. doi:https://doi.org/10.1016/j.aeue.2019.01.034

Kim, J., Chang, S., & Kwon, Y. (2008, 23-25 Jan. 2008). *ODTPC: On-demand Transmission Power Control for Wireless Sensor Networks.* Paper presented at the 2008 International Conference on Information Networking.

Kim, J., On, J., Kim, S., & Lee, J. (2008, 25-31 Aug. 2008). *Performance Evaluation of Synchronous and Asynchronous MAC Protocols for Wireless Sensor Networks.* Paper presented at the 2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008).

Kim, S., & Eom, D. S. (2014). Link-state-estimation-based transmission power control in wireless body area networks. *IEEE J Biomed Health Inform, 18*(4), 1294-1302. doi:10.1109/JBHI.2013.2282864

Kim, W. J. W. p. c. (2013). Short clear channel assessment in slotted IEEE 802.15. 4 networks. *71*(1), 735-744.

King, A., & Roedig, U. J. A. T. o. S. N. (2018). Differentiating clear channel assessment using transmit power variation. *14*(2), 1-28.

Kori, R. H., Angadi, A. S., Hiremath, M. K., & Iddalagi, S. (2009). *Efficient power utilization of wireless sensor networks: A survey.* Paper presented at the 2009

International Conference on Advances in Recent Technologies in Communication and Computing.

Korthauer, R. (2018). *Lithium-ion batteries: basics and applications*: Springer.

Kumar, A., Zhao, M., Wong, K. J., Guan, Y. L., & Chong, P. H. J. (2018). A Comprehensive Study of IoT and WSN MAC Protocols: Research Issues, Challenges and Opportunities. *IEEE Access, 6*, 76228-76262. doi:10.1109/ACCESS.2018.2883391

Li, A., Pelissier, S., Venet, P., & Gyan, P. (2016). Fast Characterization Method for Modeling Battery Relaxation Voltage. *2*(2), 7.

Li, C., Zhu, L., Xu, S., Gao, P., & Xu, B. (2018). *Recurrent neural network based small-footprint wake-up-word speech recognition system with a score calibration method.* Paper presented at the 2018 24th International Conference on Pattern Recognition (ICPR).

Li, L., Wang, P., Chao, K.-H., Zhou, Y., & Xie, Y. J. P. O. (2016). Remaining useful life prediction for lithium-ion batteries based on Gaussian processes mixture. *11*(9), e0163004.

Li, S., & Kim, J. G. (2015). Maximizing the lifetime of wireless sensor networks with random forwarding. *AEU - International Journal of Electronics and Communications, 69*(1), 455-457. doi:https://doi.org/10.1016/j.aeue.2014.09.014

Li, Y., Guo, H., Qi, F., Guo, Z., & Li, M. (2020). Comparative Study of the Influence of Open Circuit Voltage Tests on State of Charge Online Estimation for Lithium-Ion Batteries. *IEEE Access, 8*, 17535-17547. doi:10.1109/ACCESS.2020.2967563

Li, Y., Stroe, D.-I., Cheng, Y., Sheng, H., Sui, X., & Teodorescu, R. J. J. o. E. S. (2021). On the feature selection for battery state of health estimation based on charging–discharging profiles. *33*, 102122.

Li, Z., Huang, J., Liaw, B. Y., & Zhang, J. (2017). On state-of-charge determination for lithium-ion batteries. *Journal of Power Sources, 348*, 281-301. doi:10.1016/j.jpowsour.2017.03.001

Lin, S., Miao, F., Zhang, J., Zhou, G., Gu, L., He, T., . . . Pappas, G. J. J. A. T. o. S. N. (2016). ATPC: Adaptive transmission power control for wireless sensor networks. *12*(1), 1-31.

Liu, J.-L., Ravishankar, C. V. J. I. J. o. M. L., & Computing. (2011). LEACH-GA: Genetic algorithm-based energy-efficient adaptive clustering protocol for wireless sensor networks. *1*(1), 79.

Liu, L., Chen, Z., Wang, C., Lin, F., & Wang, H. J. I. T. o. E. c. (2012). Integrated system identification and state-of-charge estimation of battery systems. *28*(1), 12-23.

Liu, T., Ge, S., Yang, X.-G., & Wang, C.-Y. (2021). Effect of thermal environments on fast charging Li-ion batteries. *Journal of Power Sources, 511*, 230466. doi:10.1016/j.jpowsour.2021.230466

Liu, Z., Dang, X., & Jing, B. (2019). A Novel Open Circuit Voltage Based State of Charge Estimation for Lithium-Ion Battery by Multi-Innovation Kalman Filter. *IEEE Access, 7*, 49432-49447. doi:10.1109/ACCESS.2019.2910882

Louail, L., & Felea, V. (2016). *Routing-aware time slot allocation heuristics in contention-free sensor networks.* Paper presented at the International Conference on Wired/Wireless Internet Communication.

Lu, G., Krishnamachari, B., & Raghavendra, C. S. (2004). *An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks.* Paper presented at the 18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.

Manwell, J. F., & McGowan, J. G. J. S. e. (1993). Lead acid battery storage model for hybrid energy systems. *50*(5), 399-405.

Masood, M. M. Y., Ahmed, G., & Khan, N. M. (2012). *A Kalman filter based adaptive on demand transmission power control (AODTPC) algorithm for wireless sensor networks.* Paper presented at the 2012 International Conference on Emerging Technologies.

McCrae, R. (2018). *The Impact of Cost on Feature Selection for Classifiers.* (Ph.D.), Nova Southeastern University, Ann Arbor. Retrieved from https://www.proquest.com/dissertations-theses/impact-cost-on-feature-selection-classifiers/docview/2158659866/se-2?accountid=14645 (13423087)

Memmert. (2021). *Climate Chambers*. Retrieved from https://www.memmert.com/fileadmin/products/documents/categories/BR_Memmert_Climate-chambers_EN.pdf

Messing, M., Shoa, T., & Habibi, S. (2019). *Lithium-ion battery relaxation effects.* Paper presented at the 2019 IEEE Transportation Electrification Conference and Expo (ITEC).

Michel, M., & Quoitin, B. J. a. p. a. (2014). Technical report: ContikiMAC vs X-MAC performance analysis.

Mihai, G., Alina, D. A., & Ion, B. (2010). *Performance analysis on T-MAC protocol over a body area network.* Paper presented at the 2010 3rd international symposium on electrical and electronics engineering (ISEEE).

Molnar, A., Lu, B., Lanzisera, S., Cook, B. W., & Pister, K. S. (2004). *An ultra-low power 900 MHz RF transceiver for wireless sensor networks.* Paper presented at the Proceedings of the IEEE 2004 Custom Integrated Circuits Conference (IEEE Cat. No. 04CH37571).

Moulton, B., Hanlen, L., Chen, J., Croucher, G., Mahendran, L., & Varis, A. (2010). Body-area-network transmission power control using variable adaptive feedback periodicity. *2010 Australian Communications Theory Workshop, AusCTW 2010*, 139-144. doi:10.1109/AUSCTW.2010.5426767

Mounib, K., Mouhcine, G., & Hussein T, M. J. W. S. N. (2012). Priority-based CCA periods for efficient and reliable communications in wireless sensor networks. *2012*.

Murnane, M., & Ghazel, A. J. A. d. (2017). A closer look at state of charge (SOC) and state of health (SOH) estimation techniques for batteries. *2*, 426-436.

Nasseri, M., Al-Olimat, H., Alam, M., Kim, J., Green, R., & Cheng, W. (2015). *Contiki Cooja simulation for time bounded localization in wireless sensor network*. Paper presented at the Proceedings of the 18th Symposium on Communications & Networking, Alexandria, Virginia.

Nazir, B., Hasbullah, H., & Madani, S. A. (2011). Sleep/wake scheduling scheme for minimizing end-to-end delay in multi-hop wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking, 2011*(1), 92. doi:10.1186/1687-1499-2011-92

Ng, M.-F., Zhao, J., Yan, Q., Conduit, G. J., & Seh, Z. W. (2020). Predicting the state of charge and health of batteries using data-driven machine learning. *Nature Machine Intelligence, 2*(3), 161-170. doi:10.1038/s42256-020-0156-7

Ng, M.-F., Zhao, J., Yan, Q., Conduit, G. J., & Seh, Z. W. J. N. M. I. (2020). Predicting the state of charge and health of batteries using data-driven machine learning. *2*(3), 161-170.

Nguyen, H. A., Förster, A., Puccinelli, D., & Giordano, S. (2011). *Sensor node lifetime: An experimental study.* Paper presented at the 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops).

Nikodem, M., Slabicki, M., Surmacz, T., & Wojciechowski, B. (2014). *Transmission power control based on packet reception rate.* Paper presented at the 2014 6th International Conference on New Technologies, Mobility and Security (NTMS).

Niri, M. F., Bui, T. M., Dinh, T. Q., Hosseinzadeh, E., Yu, T. F., & Marco, J. J. J. o. E. S. (2020). Remaining energy estimation for lithium-ion batteries via Gaussian mixture and Markov models for future load prediction. *28*, 101271.

Odeyinka, O. J., Ajibola, O. A., Ndinechi, M. C., Nosiri, O. C., Onuekwusi, N. C. J. I. J. o. S. S. T., & Applications. (2020). A Review on Conservation of Energy in Wireless Sensor Networks. *1*(2), 1-16.

Osamy, W., El-Sawy, A. A., & Khedr, A. M. (2020). Effective TDMA scheduling for tree-based data collection using genetic algorithm in wireless sensor networks. *Peer-to-Peer Networking and Applications, 13*(3), 796-815. doi:10.1007/s12083-019-00818-z

Otis, B., Chee, Y. H., Lu, R., Pletcher, N., Gambini, S., & Rabaey, J. J. L.-P. E. D. (2018). Highly Integrated Ultra-Low Power RF Transceivers for Wireless Sensor Networks.

Ovejas, V. J., & Cuadras, A. (2019). Effects of cycling on lithium-ion battery hysteresis and overvoltage. *Scientific Reports, 9*(1), 14875. doi:10.1038/s41598-019-51474-5

Panasonic. (2007). *Lithium-Ion Prismatic Rechargeable battery*. Retrieved from https://industrial.panasonic.com/cdbs/www-data/pdf/ACA4000/ACA4000PE3.pdf

Panasonic. (2020). NCR18650PF | Panasonic Industrial Devices. Retrieved from https://b2b-api.panasonic.eu/file_stream/pids/fileversion/3447

Panwar, G., & Misra, S. J. I. W. C. (2016). Building Wireless Sensor Networks, Theoretical and Practical Perspectives (Nandini Mukherjee, Sarmistha Neogy,

and Sarbani Roy; 2015)[Book Review]. *IEEE Wireless Communications, 23*(2), 4-5.

Park, C., Lahiri, K., & Raghunathan, A. (2005). *Battery discharge characteristics of wireless sensor nodes: An experimental analysis.* Paper presented at the 2005 Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005.

Paruchuri, V., Basavaraju, S., Durresi, A., Kannan, R., & Iyengar, S. S. (2004). *Random asynchronous wakeup protocol for sensor networks.* Paper presented at the First International Conference on Broadband Networks.

Peiris, V., Hoiydi, A. E., Ribordy, A., Le Roux, E., Melly, T., Ruffieux, D., . . . Kucera, M. J. A. C. D. (2007). WiseNET, an ultra low-power RF transceiver SoC and communication protocol solution for wireless sensor Networks. 345-375.

Petrova, M., Wu, L., Mahonen, P., & Riihijarvi, J. (2007). *Interference measurements on performance degradation between colocated IEEE 802.11 g/n and IEEE 802.15. 4 networks.* Paper presented at the Sixth International Conference on Networking (ICN'07).

Piller, S., Perrin, M., & Jossen, A. J. J. o. p. s. (2001). Methods for state-of-charge determination and their applications. *96*(1), 113-120.

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines.

Polastre, J., Hill, J., & Culler, D. (2004). *Versatile low power media access for wireless sensor networks.* Paper presented at the Proceedings of the 2nd international conference on Embedded networked sensor systems.

Pourpeighambar, S. B., & Sabaei, M. (2013). Spatial correlation aware protocols for efficient data aggregation of moving object in wireless sensor networks. *Scientia Iranica, 20*(3), 695-709. doi:https://doi.org/10.1016/j.scient.2012.11.010

Qadrdan, M., Jenkins, N., & Wu, J. (2018). Chapter II-3-D - Smart Grid and Energy Storage. In S. A. Kalogirou (Ed.), *McEvoy's Handbook of Photovoltaics (Third Edition)* (pp. 915-928): Academic Press.

Quintero, V. L., Estevez, C., Orchard, M. E., & Pérez, A. (2019). Improvements of Energy-Efficient Techniques in WSNs: A MAC-Protocol Approach. *IEEE Communications Surveys & Tutorials, 21*(2), 1188-1208. doi:10.1109/COMST.2018.2875810

Rajendran, V., Garcia-Luna-Aveces, J., & Obraczka, K. (2005). *Energy-efficient, application-aware medium access for sensor networks.* Paper presented at the IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005.

Ramachandran, I., & Roy, S. (2006, 27 Nov.-1 Dec. 2006). *WLC46-2: On the Impact of Clear Channel Assessment on MAC Performance.* Paper presented at the IEEE Globecom 2006.

Ramachandran, I., & Roy, S. J. I. W. C. (2007). Clear channel assessment in energyconstrained wideband wireless networks. *14*(3), 70-78.

Ramadan, K. F., Dessouky, M. I., Abd-Elnaby, M., & Abd El-Samie, F. E. (2018). Node-power-based MAC protocol with adaptive listening period for wireless sensor networks. *AEU - International Journal of Electronics and Communications, 84*, 46-56. doi:https://doi.org/10.1016/j.aeue.2017.10.034

Rashid, A., Khan, F., Gul, T., Khan, S., & Khan, F. J. I. J. A. C. S. A. (2018). Improving energy conservation in wireless sensor networks using energy harvesting system. *9*(1).

Rashid, R. A., Embong, W., Zaharim, A., & Fisal, N. J. E. J. o. S. R. (2009). Development of energy aware TDMA-based MAC protocol for wireless sensor network system. *30*(4), 571-578.

Rault, T., Bouabdallah, A., & Challal, Y. J. C. n. (2014). Energy efficiency in wireless sensor networks: A top-down survey. *67*, 104-122.

Ren, D., Lu, L., Shen, P., Feng, X., Han, X., & Ouyang, M. J. J. o. E. S. (2019). Battery remaining discharge energy estimation based on prediction of future operating conditions. *25*, 100836.

Ren, X., Liu, S., Yu, X., & Dong, X. J. E. (2021). A method for state-of-charge estimation of lithium-ion batteries based on PSO-LSTM. 121236.

Rodrigues, L. M., Montez, C., Budke, G., Vasques, F., Portugal, P. J. J. o. S., & Networks, A. (2017). Estimating the lifetime of wireless sensor network nodes through the use of embedded analytical battery models. *6*(2), 8.

Roscher, M. A., Bohlen, O., & Vetter, J. (2011). OCV Hysteresis in Li-Ion Batteries including Two-Phase Transition Materials. *International Journal of Electrochemistry, 2011*, 984320. doi:10.4061/2011/984320

Roscher, M. A., & Sauer, D. U. J. J. o. P. S. (2011). Dynamic electric behavior and open-circuit-voltage modeling of LiFePO4-based lithium ion secondary batteries. *196*(1), 331-336.

Rose, M. S. A., & Banu, M. N. S. (2019). Game Theory and Queue Learning Based Sleep/Wakeup Approach Scheduling in Wireless Sensor Network for Energy Efficiency.

Rukpakavong, W., Guan, L., & Phillips, I. J. I. S. J. (2013). Dynamic node lifetime estimation for wireless sensor networks. *14*(5), 1370-1379.

Rukpakavong, W., Phillips, I., & Guan, L. (2012). *Lifetime estimation of sensor device with AA NiMH batteries.* Paper presented at the Proc. ICICM.

Samant, T., Kumar, Y. S., & Swayamsiddha, S. J. S. N. M. f. S. A. (2020). Comparison Analysis of MAC Protocols for Wireless Sensor Networks: A Comprehensive Survey. 200-218.

Santi, P. (2005). Topology control in wireless ad hoc and sensor networks. *ACM computing surveys, 37*(2), 164-194.

Schurgers, C., Tsiatsis, V., Ganeriwal, S., & Srivastava, M. J. I. t. o. m. c. (2002). Optimizing sensor networks in the energy-latency-density design space. *1*(1), 70-80.

Schurgers, C., Tsiatsis, V., & Srivastava, M. B. (2002). *STEM: Topology management for energy efficient sensor networks.* Paper presented at the Proceedings, IEEE Aerospace Conference.

Sha, M., Hackmann, G., & Lu, C. (2013). *Energy-efficient low power listening for wireless sensor networks in noisy environments.* Paper presented at the Proceedings of the 12th international conference on Information processing in sensor networks.

Sharma, G., Kumar, A. J. C., & Engineering, E. (2018). Improved range-free localization for three-dimensional wireless sensor networks using genetic algorithm. *72*, 808-827.

Shen, D., Wu, L., Kang, G., Guan, Y., & Peng, Z. J. E. (2021). A novel online method for predicting the remaining useful life of lithium-ion batteries considering random variable discharge current. *218*, 119490.

Singh, K., Tjahjowidodo, T., Boulon, L., & Feroskhan, M. (2022). Framework for measurement of battery state-of-health (resistance) integrating overpotential effects and entropy changes using energy equilibrium. *Energy, 239*, 121942. doi:10.1016/j.energy.2021.121942

Singh, S. K., Kumar, P., & Singh, J. P. J. I. A. (2017). A survey on successors of LEACH protocol. *5*, 4298-4328.

Sodhro, A. H., Li, Y., & Shah, M. A. J. I. C. (2016). Energy-efficient adaptive transmission power control for wireless body area networks. *10*(1), 81-90.

Son, D., Krishnamachari, B., & Heidemann, J. (2004). *Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks.* Paper presented at the 2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.

Son, K. J., Hong, S. H., Moon, S.-P., Chang, T. G., & Cho, H. (2016). Segmentized Clear Channel Assessment for IEEE 802.15.4 Networks. *16*(6), 815.

Song, L., Zhang, K., Liang, T., Han, X., & Zhang, Y. J. J. o. E. S. (2020). Intelligent state of health estimation for lithium-ion battery pack based on big data analysis. *32*, 101836.

Sonkar, C. K., Sangwan, O. P., & Tripathi, A. M. (2013, 2013//). *Comparative Analysis of Contention Based Medium Access Control Protocols for Wireless Sensor Networks.* Paper presented at the Quality, Reliability, Security and Robustness in Heterogeneous Networks, Berlin, Heidelberg.

Srivastava, S., Singh, M., & Gupta, S. (2018). *Wireless sensor network: a survey.* Paper presented at the 2018 International Conference on Automation and Computational Engineering (ICACE).

Staub, T., Bernoulli, T., Anwander, M., Waelchli, M., & Braun, T. (2006). *Experimental lifetime evaluation for mac protocols on real sensor hardware.* Paper presented at the ACM Workshop on Real-World Wireless Sensor Networks REALWSN.

Stroe, D. I. (2014). Lifetime models for Lithium-ion batteries used in virtual power plant applications.

Su, R., Gong, Z., Zhang, D., Li, C., Chen, Y., & Venkatesan, R. J. I. T. o. V. T. (2021). An adaptive asynchronous wake-up scheme for underwater acoustic sensor networks using deep reinforcement learning. *70*(2), 1851-1865.

Su, R., Venkatesan, R., Li, C. J. W. C., & Computing, M. (2016). An energy-efficient asynchronous wake-up scheme for underwater acoustic sensor networks. *16*(9), 1158-1172.

Suh, C., & Ko, Y.-B. (2005). *A traffic aware, energy efficient MAC protocol for wireless sensor networks.* Paper presented at the 2005 IEEE International Symposium on Circuits and Systems.

Sun, J., Tang, Y., Ye, J., Jiang, T., Chen, S., & Qiu, S. J. E. (2022). A novel capacity and initial discharge electric quantity estimation method for LiFePO4 battery pack based on OCV curve partial reconstruction. *243*, 122882.

Sun, Y., Gurewitz, O., & Johnson, D. B. (2008). *RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks*. Paper presented at the Proceedings of the 6th ACM conference on Embedded network sensor systems, Raleigh, NC, USA.

Tayeh, G. B., Makhoul, A., Perera, C., & Demerjian, J. (2019). A Spatial-Temporal Correlation Approach for Data Reduction in Cluster-Based Sensor Networks. *IEEE Access, 7*, 50669-50680. doi:10.1109/ACCESS.2019.2910886

Technology, L. (2014). *LTC4150 - Coulomb counter/ Battery Gas Gauge*. Retrieved from Analog Website: https://www.analog.com/media/en/technical-documentation/data-sheets/4150fc.pdf

Teng, H., Liu, X., Liu, A., Shen, H., Huang, C., & Wang, T. (2018). Adaptive Transmission Power Control for Reliable Data Forwarding in Sensor Based Networks. *Wireless Communications and Mobile Computing, 2018*, 2068375. doi:10.1155/2018/2068375

Tirumala, S. S. (2021). Exploring Neural Network Layers for Knowledge Discovery. *Procedia Computer Science, 193*, 173-182. doi:https://doi.org/10.1016/j.procs.2021.10.017

Toor, A. S., & Jain, A. K. (2019). Energy Aware Cluster Based Multi-hop Energy Efficient Routing Protocol using Multiple Mobile Nodes (MEACBM) in Wireless Sensor Networks. *AEU - International Journal of Electronics and Communications, 102*, 41-53. doi:https://doi.org/10.1016/j.aeue.2019.02.006

Torabi, F., & Ahmadi, P. (2020). Chapter 1 - Battery technologies. In F. Torabi & P. Ahmadi (Eds.), *Simulation of Battery Systems* (pp. 1-54): Academic Press.

Tran, T. T. K., Lee, T., & Kim, J.-S. (2020). Increasing Neurons or Deepening Layers in Forecasting Maximum Temperature Time Series? *, 11*(10), 1072.

Tremblay, O., Dessaint, L.-A., & Dekkiche, A.-I. (2007). *A generic battery model for the dynamic simulation of hybrid electric vehicles.* Paper presented at the 2007 IEEE Vehicle Power and Propulsion Conference.

Tremblay, O., & Dessaint, L.-A. J. W. e. v. j. (2009). Experimental validation of a battery dynamic model for EV applications. *3*(2), 289-298.

Tytgat, L., Yaron, O., Pollin, S., Moerman, I., Demeester, P. J. E. J. o. W. C., & Networking. (2012). Avoiding collisions between IEEE 802.11 and IEEE 802.15. 4 through coexistence aware clear channel assessment. *2012*(1), 137.

Uwase, M.-P., Bezunartea, M., Tiberghien, J., Dricot, J.-M., & Steenhaut, K. J. I. s. j. (2017). Experimental comparison of radio duty cycling protocols for wireless sensor networks. *17*(19), 6474-6482.

Vallejo, M., Recas, J., & Ayala, J. L. (2015). Proactive and Reactive Transmission Power Control for Energy-Efficient On-Body Communications. *15*(3), 5914-5934.

Van Dam, T., & Langendoen, K. (2003). *An adaptive energy-efficient MAC protocol for wireless sensor networks.* Paper presented at the Proceedings of the 1st international conference on Embedded networked sensor systems.

Verma, A., Singh, M. P., Singh, J. P., & Kumar, P. (2015, 1-2 May 2015). *Survey of MAC Protocol for Wireless Sensor Networks.* Paper presented at the 2015 Second International Conference on Advances in Computing and Communication Engineering.

Vuran, M. C., & Akyildiz, I. F. J. I. A. T. O. N. (2006). Spatial correlation-based collaborative medium access control in wireless sensor networks. *14*(2), 316-329.

Wang, K., Zhao, X., Shi, Y., Xu, D., & Li, R. (2020). The energy-efficient MDA-SMAC protocol for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking, 2020*(1), 32. doi:10.1186/s13638-020-1642-7

Wang, L. (2006). *Physical layer and application layer issues of wireless sensor networks*: The University of Texas at Arlington.

Wang, L., Fang, Y., Wang, L., Yun, F., Wang, J., & Lu, S. J. I. A. (2020). Understanding discharge voltage inconsistency in lithium-ion cells via statistical characteristics and numerical analysis. *8*, 84821-84836.

Wang, T., Wu, Z., & Mao, J. (2007). *PSO-based hybrid algorithm for multi-objective TDMA scheduling in wireless sensor networks.* Paper presented at the 2007 Second International Conference on Communications and Networking in China.

Wang, Y., Tian, J., Sun, Z., Wang, L., Xu, R., Li, M., & Chen, Z. (2020). A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems. *Renewable and Sustainable Energy Reviews, 131*, 110015. doi:https://doi.org/10.1016/j.rser.2020.110015

Wang, Y., Yang, D., Zhang, X., & Chen, Z. J. J. o. P. S. (2016). Probability based remaining capacity estimation using data-driven and neural network model. *315*, 199-208.

Warner, J. T. (2015). *The handbook of lithium-ion battery pack design: chemistry, components, types and terminology*: Elsevier.

Wei, Y., Heidemann, J., & Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking, 12*(3), 493-506. doi:10.1109/TNET.2004.828953

Xia, Z., & Qahouq, J. A. A. J. I. A. (2021). Lithium-ion battery ageing behavior pattern characterization and state-of-health estimation using data-driven method. *9*, 98287-98304.

Xiao, S., Dhamdhere, A., Sivaraman, V., & Burdett, A. (2009). Transmission power control in body area sensor networks for healthcare monitoring. *27*(1 %J IEEE J.Sel. A. Commun.), 37–48. doi:10.1109/jsac.2009.090105

Xiong, R., Cao, J., Yu, Q., He, H., & Sun, F. (2018). Critical Review on the Battery State of Charge Estimation Methods for Electric Vehicles. *IEEE Access, 6*, 1832-1843. doi:10.1109/ACCESS.2017.2780258

Xu, B., Diao, W., Wen, G., Choe, S.-Y., Kim, J., & Pecht, M. J. J. o. P. S. (2021). Decoupling the thermal and non-thermal effects of discharge C-rate on the capacity fade of lithium-ion batteries. *510*, 230390.

Xu, T., Peng, Z., & Wu, L. J. E. (2021). A novel data-driven method for predicting the circulating capacity of lithium-ion battery under random variable current. *218*, 119530.

Xu, X., Yu, C., Tang, S., Sun, X., Si, X., & Wu, L. J. E. (2019). Remaining useful life prediction of lithium-ion batteries based on Wiener processes with considering the relaxation effect. *12*(9), 1685.

Yang, A., Wang, Y., Yang, F., Wang, D., Zi, Y., Tsui, K. L., & Zhang, B. J. J. o. P. S. (2019). A comprehensive investigation of lithium-ion battery degradation performance at different discharge rates. *443*, 227108.

Yang, X., & Vaidya, N. H. (2004). *A wakeup scheme for sensor networks: Achieving balance between energy saving and end-to-end delay.* Paper presented at the Proceedings. RTAS 2004. 10th IEEE Real-Time and Embedded Technology and Applications Symposium, 2004.

Ye, D., & Zhang, M. J. I. t. o. c. (2017). A self-adaptive sleep/wake-up scheduling approach for wireless sensor networks. *48*(3), 979-992.

Ye, W., Heidemann, J., & Estrin, D. (2002). *An energy-efficient MAC protocol for wireless sensor networks.* Paper presented at the Proceedings. Twenty-first annual joint conference of the IEEE computer and communications societies.

Yeo, I. K., & Johnson, R. A. J. B. (2000). A new family of power transformations to improve normality or symmetry. *87*(4), 954-959.

Yu, J., Tang, W., Tang, D., & Liu, J. J. M. R. (2018). A prediction method for discharge voltage of lithium-ion batteries under unknown dynamic loads. *88*, 1206-1211.

Zhang, C., Jiang, Y., Jiang, J., Cheng, G., Diao, W., & Zhang, W. (2017). Study on battery pack consistency evolutions and equilibrium diagnosis for serial-connected lithium-ion batteries. *Applied Energy, 207*, 510-519. doi:10.1016/j.apenergy.2017.05.176

Zhang, T., Yi, L., Cui, X., Behl, A., Dong, F., Cheng, Z., . . . Continua. (2020). RFID based non-preemptive random sleep scheduling in wsn. *65*(1), 835-845.

Zhang, X., Ansari, J., & Mähönen, P. (2009). *Traffic aware medium access control protocol for wireless sensor networks.* Paper presented at the Proceedings of the 7th ACM international symposium on Mobility management and wireless access.

Zhang, Y., Xiong, R., He, H., & Shen, W. (2017). Lithium-Ion Battery Pack State of Charge and State of Energy Estimation Algorithms Using a Hardware-in-the-Loop Validation. *IEEE Transactions on Power Electronics, 32*(6), 4421-4431. doi:10.1109/TPEL.2016.2603229

Zheng, X. (2019). Improvement strategy of S-MAC protocol for wireless sensors in IoT based on data and energy priorities. *Systems Science & Control Engineering, 7*(2), 1-7. doi:10.1080/21642583.2019.1608474

Zhou, W., Zheng, Y., Pan, Z., & Lu, Q. J. P. (2021). Review on the Battery Model and SOC Estimation Method. *9*(9), 1685.

Zhu, Y.-H., Jia, L., & Zhang, Y. J. S. (2022). Enhancing Channel Contention Efficiency in IEEE 802.15. 4 Wireless Networks. *22*(4), 1600.

Zibakalam, V. (2012). A New TDMA Scheduling Algorithm for Data Collection over Tree-Based Routing in Wireless Sensor Networks. *ISRN Sensor Networks, 2012*, 864694. doi:10.5402/2012/864694

Zuniga, M., Irzynska, I., Hauer, J.-H., Voigt, T., Boano, C. A., & Roemer, K. (2011). *Link quality ranking: Getting the best out of unreliable links.* Paper presented at the 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS).

Zuo, H., Zhang, B., Huang, Z., Wei, K., Zhu, H., & Tan, J. J. E. (2022). Effect analysis on SOC values of the power lithium manganate battery during discharging process and its intelligent estimation. *238*, 121854.

## Appendix A: Battery parameters and specifications for system design

### Battery parameters consideration for system design

Choosing the best battery for an application requires balancing the most important battery metrics. A lead acid battery, for example, performs admirably in a vehicular starter battery, in which it provides the essential high-rate performance. However, due to its toxicity and poor energy density, it would be an unsuitable material for a mobile computing application. Therefore, it is crucial to comprehend all essential battery parameters and align them to device usage and requirements. Some of these key parameters are discussed as under:

**Primary versus Secondary batteries**

Establishing whether the system needs primary (single use) or secondary (rechargeable) batteries is one of the first steps in battery selection. The use of a primary battery is recommended for applications with sporadic use (such as a smoke detector, a digital thermometer, and pacemakers) and disposable applications where charging is impractical. A rechargeable battery is preferable if the battery will be used continuously and for long periods of time, such as in a laptop, mobile phones, or smartwatch.

**Battery capacity and discharge characteristics**

The capacity of a battery is usually expressed in ampere-hours (Ah) or milli-Ampere-hours (mAh). The discharge rate of a battery is frequently specified in terms of C-rate, where C is the battery's capacity divided by hours. For example, the C-rate for a 1000 mAh battery is 1A discharge rate for an hour, C/2 is 500 mA, and C/10 is 100mA. Another way to think about it is in terms of how much energy a battery can store for a given amount of time. For example, under ideal operating conditions, a 1000mAh battery can deliver a discharge current of 1A for 1 hour. In applications like WSN,

where energy-aware algorithms are designed to extend node lifetime, C-rate and discharge characteristics are critical.

**Battery energy versus Power**

The capacity of a battery and the rate at which it is discharged determine its runtime. When comparing different chemistry batteries, it is indeed worthwhile to look at the energy output. To calculate the energy content of a battery, multiply the capacity (Ah) by the voltage (V) to get energy in Watt-hours (Wh). For comparison purposes, a 1000 mAh Lithium-Ion battery with 3.7 V nominal voltage and a Nickel-metal hydride battery with 1.2 V may have the same capacity, but the lithium-ion battery's higher voltage will provide more energy (Andrea, 2020; Korthauer, 2018; Warner, 2015).

Manufacturers frequently state the capacity of their batteries at a specific discharge rate, temperature, and cut-off voltage. All three factors will influence the specified capacity. The discharge rate is the most important of the three. If the current discharge for the application is higher, a battery that initially appeared to have a high capacity on the manufacturer's specification sheet may underperform. A battery rated at 1 Ah for a 10-hour discharge, for example, cannot deliver a consistent 1 A for 1 hour and will only provide a fraction of its capacity.

**Maximum discharge currents**

The maximum discharge current, also known as peak load current, is the maximum current at which a battery can be discharged continuously. This limit is typically set by the manufacturer to avoid excessive discharge rates from damaging the battery or reducing its capacity. The voltage of the battery varies based on the discharge current in use (load). At higher current, the cell voltage drops more deeply. Therefore, discharge at high current reduces the battery's effective capacity. The discharge

characteristics of a Panasonic NCR18650PF cylindrical cell Lithium-Ion battery is presented in Figure A1.

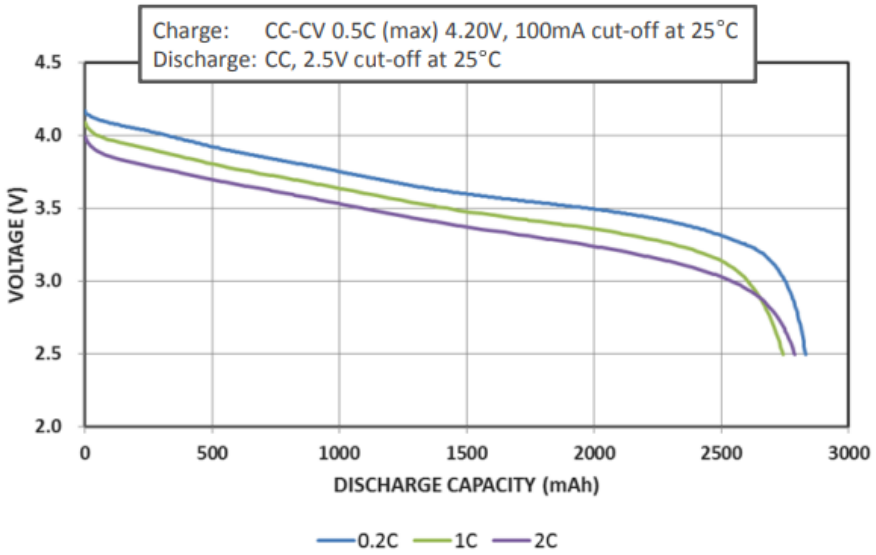



Figure A1: Discharge characteristics of Li-Ion battery (Panasonic, 2020)

**Cell Voltage**

Another important consideration is battery operating voltage, which is determined by the electrode materials used. Consider aqueous or water-based batteries versus lithium-based chemistries as a valuable battery classification. Water-based electrolytes are used in lead acid, zinc carbon, and nickel metal hydride, with nominal voltages ranging from 1.2 to 2 V. Organic electrolytes are used in lithium-ion batteries, which have nominal voltages ranging from 3.2 to 4 V (for both primary and secondary).

The cell voltage varies depending on the chemistry of the battery. Depending on the design, selecting the correct battery with the appropriate cell voltage can affect the efficiency, complexity, and cost of the device's power circuitry. Although batteries are labelled with their nominal voltage, our initial characterization revealed that the open circuit voltage (OCV) on a fully charged battery can be up to 7% higher. Many electronic components require a minimum voltage of 3 V to operate. Because lithium-based chemistries have a higher operating voltage, a single cell can be used instead of

two or three aqueous-based cells in series to achieve the desired voltage. This results in cost-reduction and eliminates further point of failures due to individual cells within a battery pack.

**Operating Temperature Range**

WSN nodes are frequently deployed in challenging locations with considerable temperature fluctuations or extreme temperatures. Because of the battery chemistry, the self-discharge rate is much higher than the nominal rate in a high-temperature environment. Similarly, in low temperature conditions, the chemical reactions of a battery may be inhibited, limiting the battery's capacity to maintain a suitable voltage for the connected load. For instance, Zinc-Carbon or Alkaline batteries underperform and sometimes even stop operation at temperatures below 0°C. Lithium batteries on the other hand, can even operate at temperatures well below 0°C, however performance suffers significantly. Similarly, the charging is also dependent on the internal battery temperature. In the case of Lithium-ion batteries, the optimal charging rate can be achieved within 20°C to 45°C temperature range.

**Specifications by battery chemistry**

Powering electronic applications efficiently and safely requires well-designed batteries. Design parameters such as energy capacity, voltage, cycle life, temperature effects and cost must be considered to achieve the best results. To evaluate key intrinsic and operational parameters specified by individual battery characteristics, standard definitions are required. This sub-section describes some of these parameters.

**Specific energy density**

The term "energy" refers to the total amount of heat or work produced. The capacity and voltage of a cell or battery are directly proportional to the amount of energy

stored in it. The energy of a cell at constant voltage can be approximated by Equation A1.1.

$$Energy\ (Wh) = Voltage\ (V)\ \times Capacity\ (Ah) \qquad \text{(A1.1)}$$

The nominal OCV for voltage (at 50 percent SoC) and nominal capacity for battery capacity calculation was used. Since the OCV varies with the SoC, and the terminal voltage is dependent on the discharge rate, the actual energy is slightly different from this value. The energy density is normally evaluated to select size and weight of the batteries which is mostly the case with WSN. The energy density varies within battery chemistries that is illustrated by Figure A2.

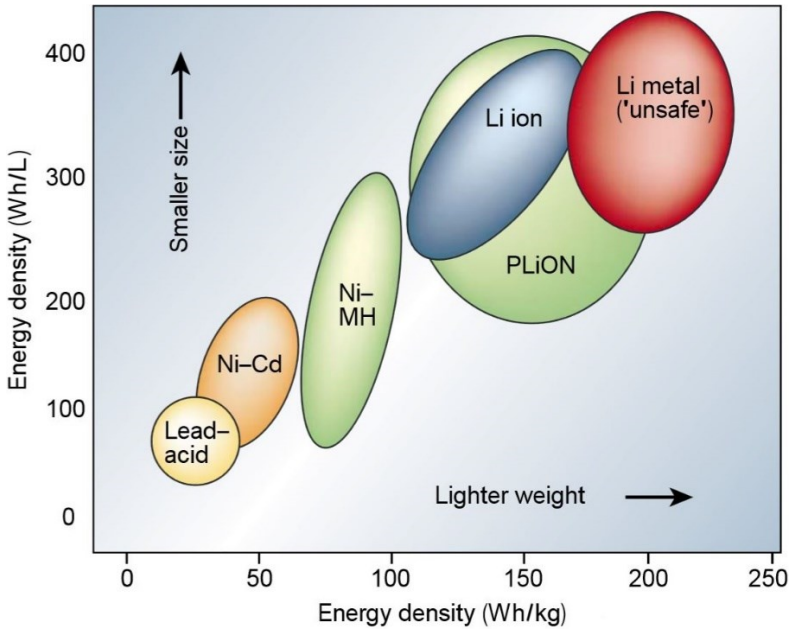



Figure A2: Energy densities of various battery chemistries (D. Deng, 2015)

The energy density plays a vital role in deciding the form-factor of a battery for intending applications. For instance, Lithium-based batteries offer the highest energy density per unit mass, making them the most chosen battery types for electronic devices these days.

**Self-discharge**

Inside the cell, a chemical reaction generates electricity. This usually occurs when it is connected to a power source. When the battery's electrodes are not attached,

the reaction is smaller. Self-discharge occurs when the battery's charge decreases over time. Unavoidable battery self-discharge depends on the battery's type and contents. The environment affects the rate and intensity of this chemical reaction. Chemical reactions slow down at low temperatures, but humidity and moisture can permanently damage batteries. Batteries should be stored between 10°C and 25°C to reduce self-discharge.

**Cycle life**

The number of charge and discharge cycles a battery can complete before losing performance is referred to as its cycle life. The depth of discharge has a significant impact on the cycle life of the batteries. The depth of discharge (DOD) refers to how much of a battery's capacity is used. A battery that has been discharged to 30% of its full energy capacity, for example, has a much longer cycle life than the one that has been discharged to 85% of its capacity, leaving only 15% of its full energy charge. Figure A3 explains the effect of DOD on the cycle life of Li-Ion batteries.

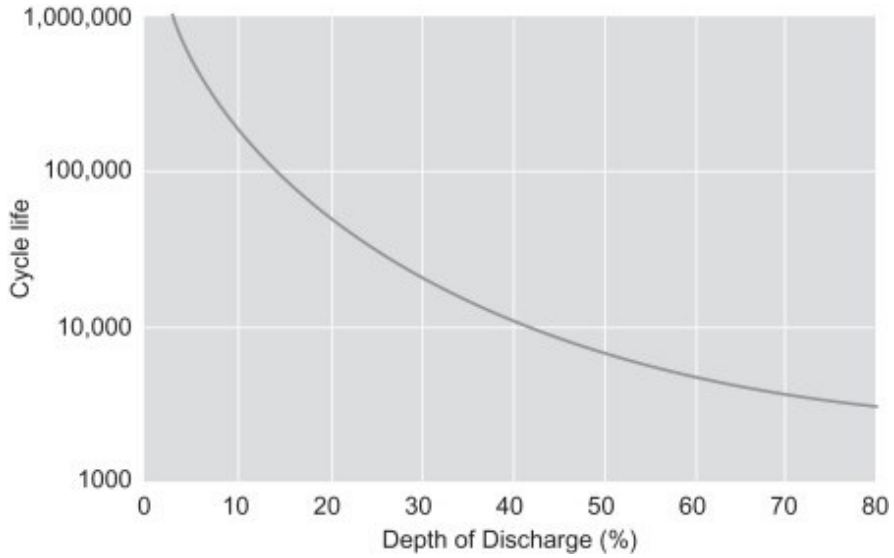



Figure A3: Typical DOD cycle life of a Li-Ion (Qadrdan, Jenkins, & Wu, 2018)

Some of the battery specifications that are typically used by system designers for the selection of the most suited battery chemistry are given by Table A.1.

Table A.1: Specifications of performance parameters by battery chemistry

| Specifications | Alkaline | Ni-MH | Lithium-based |
|---|---|---|---|
| Specific energy density (Wh/kg) | 80 | 60-120 | 120-190 |
| Cell Voltage (nominal) | 1.5 V | 1.2 V | 3.3 V |
| Charge cutoff voltage (V/cell) | 0.9 V | Full charge detection | 3.60 V |
| Peak load current | 0.2C – 0.5C | 0.5C – 5C | 1C – 10C |
| Internal resistance (mΩ) | 200 – 2000<br>6V pack | 200-300<br>6V pack | 25-75<br>7.2V pack |
| Cycle life (80% discharge) | 50 | 300-500 | 500 - 1000 |
| Self-discharge (per month) | 0.3% | 30% | <10% |
| Safety requirements | Not required | Thermally stable | Mandatory protection circuit required |

This data shows the battery's energy capacity and economic and technical viability for the applications in question. However, comparing individual battery parameters to other battery chemistries can be difficult. In this case, the Radar chart, also known as the Spider chart, is used. Radar charts help compare different battery technologies. For clarity, no more than four parameters can be compared in a single chart. A radar chart comparing various battery chemistries is presented in Figure A4.

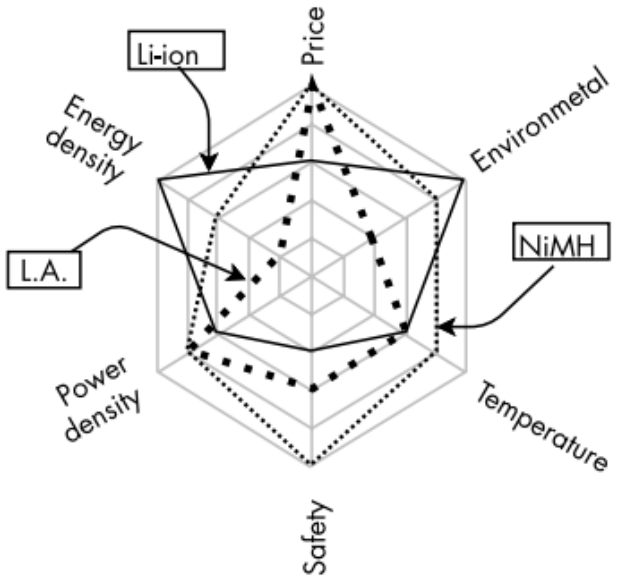



Figure A4: A radar chart comparing various battery chemistries.

## Appendix B: Feedforward Neural Network architecture requirements

### Feed Forward Neural Networks (FFNN)

One of the ML techniques used in this study is a feedforward neural network (FFNN). A FFNN has several layers: an input layer, n hidden layers, and an output layer. Each layer has a set number of nodes, perceptrons, and weights. The most common activation functions for perceptrons are sigmoid and linear transfer functions. Asymmetric output neurons are used for pattern recognition, while linear output neurons are used for fitting. Weights of inter-layer connections are trained along with the network. The input layer receives the feature vector X (capacity and voltage), while the output layer estimates the SOC for each input vector. MATLAB's default architecture is a single layer FFNN. However, the number of perceptrons per layer and the number of layers affect the type of neural network architecture.

### Network Architecture and Configurations

The number of layers in a Neural Network determines its ability to learn specific patterns. Adding layers to the NN requires more resources. An input layer sends weighted inputs to a layer of receiving nodes, or even to just one receiving node in some cases. This single-layer design laid the groundwork for much more complex structures. A single active layer NN can only learn linearly separable problems. A multilayer perceptron (MLP) is a type of FFNN with one or more hidden layers. MLP NN can separate data patterns with multiple lines that form different shapes (usually required in classification problems). A NN with three active layers can separate data into any shape, implying that an MLP with three layers can solve any problem. Deep neural networks are networks with three or more hidden layers (DNN). For a given NN architecture, output requirements typically determine the number of hidden layers, perceptrons per layer, and activation functions (Hagan, Demuth, & Beale, 1997).

However, it is best to design a network by carefully modelling the degree of freedom based on the dataset. This results in a more efficient and computationally light network design. Contrary to popular belief, increasing the network's resources may reduce its accuracy. It's intriguing to consider how a neural network model's capacity influences the mapping functions it can learn. Poor models cannot learn the training dataset and thus underfit. Overfitting occurs when a model's capacity is exceeded. Because these models typically use a lot of resources, careful planning is required (Brownlee, 2018; Goodfellow, Bengio, & Courville, 2017).

The number of layers and neurons per layer determine the model's capacity. A layer's width is its node count. The depth is the number of layers. So, there are two NN architectures. For starters, depth-wise architectures increase model layers, resulting in shallow or DNN models. Two-layered models result from increasing architecture width. Increasing model depth increases capacity but may result in computationally expensive and resource-intensive models. Wide networks are less complex and appear to be suitable for embedded platforms. Single-layered networks may also underperform for complex problems due to the effect of middle layers on model accuracy (Tirumala, 2021).

In this context, model adaptability and scalability are critical. Despite its computational complexity, a depth-wise or deep network solves configuration issues. This improves learning and adaptability to a variety of datasets. Wide networks are computationally efficient but selecting the optimal number of neurons is difficult. These models also lack a diverse feature set, limiting model capacity. So, to keep the design simple and computationally efficient, a few rules were followed layer by layer. In this regard, Figure shows several NN configurations that are further explained at each layer level.

**Neuron Model – Input Layer**

Any NN architecture starts with a single input neuron or perceptron. The sum function takes the scalar input **p** and multiplies it by the scalar weight ***w***. Moreover, input is multiplied by a bias ***b***. Net input from summer produces a scalar neuron output (a). In some papers, transfer functions are called "activation functions" and bias is called "offset". A neuron typically consists of more than a single input. Figure B1(right) depicts a neuron with ***R*** inputs. Individual inputs $p_1, p_2, \ldots, p_R$ are weighted by corresponding weight matrix ***W*** with elements $w_{1,1}, w_{1,2}, \ldots, w_{1,R}$.

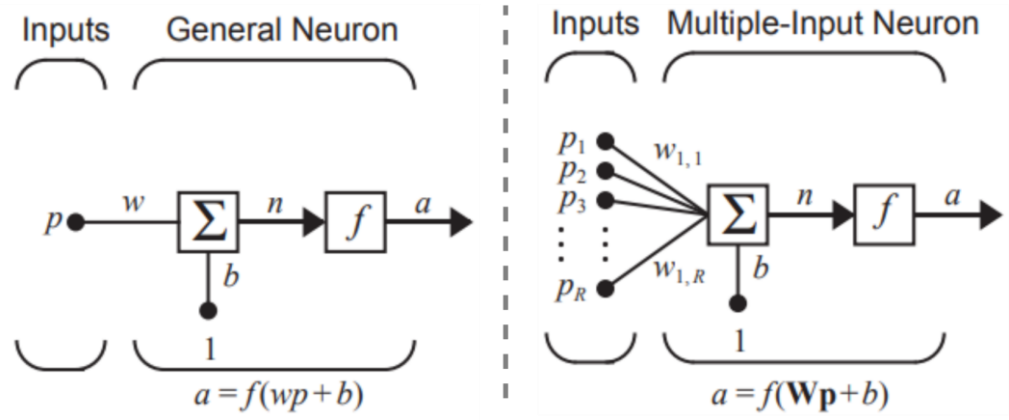


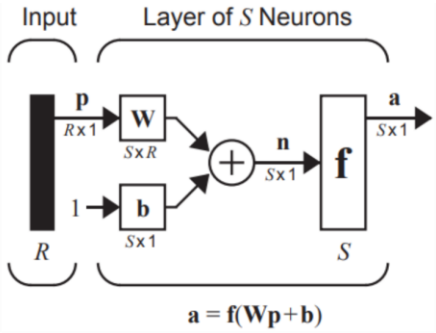

Figure B1: An illustration of Single-Input Neuron (left) versus Multiple-Input Neuron (right)

Once the shape or degree of the training data is known, the number of neurons comprising this layer is completely and uniquely determined. The number of neurons in that layer is specifically equal to the number of features (columns) in the dataset. It is very common to see some NN configurations with one extra node for a bias term.

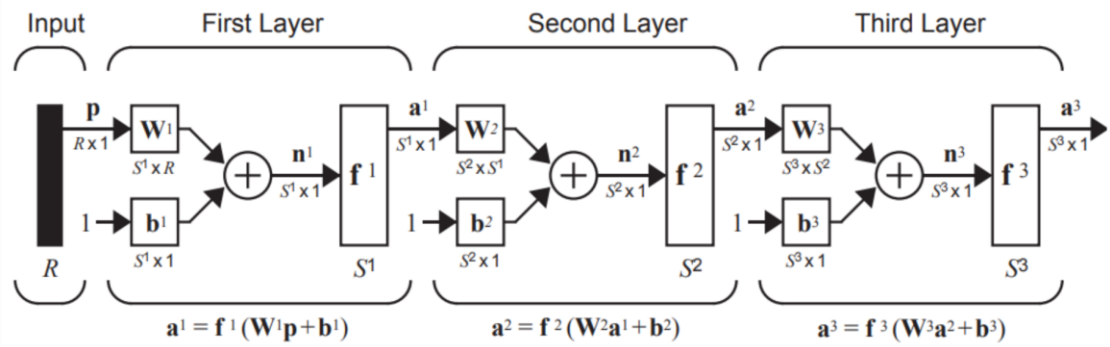
**Network or Hidden Layers**

A machine learning model can theoretically be applied to any linear or nonlinear problem. However, it is critical to consider whether the application necessitates the use of complex algorithms and computing resources. For instance, for linearly separable datasets, the inputs can be directly mapped to the activation functions and the output, obviating the need for hidden layers entirely. Because the hidden layer defines the

degree of complexity of a NN, a typical question for NN designers always requires careful selection of (i) number of hidden layers, and (ii) perceptrons per hidden layer. The choice of a specific number of hidden layers has a direct impact on the model's performance. It is very common to see a slight improvement in model performance after adding a second, third, or more layers (as in the case of DNN). Figure B2 depicts a simple illustrative comparison of single layer versus multi-layer network architecture.




(a) Abbreviated notation of a Single-layer Multi-Input Neuron Architecture




(b) Abbreviated notation of a Multi-layer Multi-Input Neuron Architecture

Figure B2: An illustrative schematic of network layers in NN architecture

As evident, each layer connects to the next layer increasing the degree of freedom and therefore the complexity of the network architecture. Therefore, this necessitates an empirical understanding of the problem on an application-by-application basis. It is critical to note that there are no hard and fast rules for designing the NN architecture, with the majority of the optimization based on trial-and-error (Heaton, 2015). However, a few rules regarding hidden layer configurations are usually followed, and they are as follows:

- For simple problems requiring no additional second step optimization, number of hidden layers can be set to one
- The number of neurons in that layer is the arithmetic mean of those in the input and output layers.
- Avoiding over-fitting and under-fitting can be achieved by using Equation B1.1.

$$N_h = \frac{N_s}{\left(\propto * \ (N_i + \ N_o)\right)} \qquad \text{(B1.1)}$$

Where $N_h$is the number of hidden layers, $N_s$ is the number of samples in the training dataset, $N_i$ is the number of input neurons, $N_o$ is the number of output neurons, and $\propto$ is the scaling factor, usually selected between (2-5). In this regard, the degree of freedom is said to be based on number of samples, and number of samples in each dimension, as given in Equation B1.2.

$$N_s(N_i + \ N_o) \qquad \text{(B1.245)}$$

Model optimization techniques such as backpropagation are used to continuously monitor the output of a NN and tune the hidden layer parameters.

**The Output layers**

The ML problem type governs the selection of output layer and neurons per output layer in the majority of scenarios. Some NN architectures, for example, can have multiple output layers for a classification-based problem. However, because regression-based problems typically predict (estimate) a single value, the output layer in these architectures is comprised of a single node.

## Appendix C: Performance evaluation metrics for ML models

### Mean Absolute Error (MAE)

The average difference between the original values and the predicted values is the Mean Absolute Error (MAE). It indicates how far the forecasts differed from the actual outcome. However, MAE provides no information on the progression of these errors, which is sometimes required to analyze whether the model is overestimating or underestimating the data. MAE can be expressed as given by Equation C1.1.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \quad \text{(C1.1)}$$

Where $y_i$ and $x_i$ are the predicted and true values. As the MAE metric is essentially the absolute average of the error between predicted and true values, a smaller MAE score is considered a better model performance metric.

### R-Squared ($R^2$)

R-squared ($R^2$) is a statistical metric that indicates the proportion of the variation explained by an independent variable or variables in a regression model for a dependent variable. In contrast to the correlation that indicates the strength of correlated predictor and target variables, $R^2$ indicates the variance between them. Therefore, an $R^2$ value close to 1 indicates the least variance and is thus considered for a best-fit model. In other words, this metric explains the overall goodness of fit for a regression model. It can be mathematically expressed using Equation C1.2.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \quad \text{(C1.2)}$$

Where, $y_i$ and $\hat{y}_i$ are actual and predicted values, respectively.

### Root Mean Squared Error (RMSE)

Mean Squared Error (MSE) is very similar to MAE, with the exception that MSE calculates the average of the square of the difference between the original and

projected values. RMSE extends MSE by calculating the rate of change that appears on the fitted model. In other words, it focuses on the residuals by calculating the standard deviation. Therefore, the lower the RMSE values, the lesser the residuals, and thus a good model fit. RMSE calculation is given by Equation C1.3.

$$RMSE = \frac{\sqrt{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{n} \quad \text{(C1.3)}$$

RMSE is mostly considered the most effective metric for regression-based ML models evaluation. Typically, RMSE and MAE values define the variance of residuals, whereas $R^2$ values describe the goodness of the fit.

## Appendix D: Model Equations for Energy calculations

Equivalent battery model for battery parameter extraction. The derivations discussed in this section form a basis of equivalent electric model, which later uses Shepherd and Trembley model to derive battery parameter extraction. These differential equations were further used in KiBAM and TKiBAM and proposed model implementations.

In this model, the effect of battery polarization is not completely ignored, which means that the displacement of electrode potential from the equilibrium value is negligible. Using Shepherd's equation, the overall battery potential is measured by:

i. Overall cathode potential (Ec)

ii. Overall anode potential (Ea)

iii. Internal resistance, ionic discharge, and polarization resistance

The Shepherd battery model discusses the electrochemistry properties, where ionic discharge is a key factor in measuring the displacement of potential between cathode and anode. Ionic discharge between the initial overpotential and steady state can be given as,

$$i = i_o e^{\frac{\alpha \eta ZF}{RT}} - i_o e^{-\frac{(1-\alpha)\eta ZF}{RT}} \quad \text{(D1.1)}$$

Where $\alpha$= transfer coefficient, Z= number of electrons transferred, $i_o$= apparent exchange current density, F = Faraday constant, R = gas constant.

Some of these parameters require detailed results from manufacturing datasheet or advanced measurements, adding both to the complexity and the cost of the experiments. Therefore, a simplified numerical approach is considered where ionic discharge is modeled as the function of polarization voltages across polarization resistance. If the previously stated factors are ignored, the discharge potential can be given as,

$$E_c = E_{sc} - K_c i_{am} \quad \text{(D1.2)}$$

Where, $E_c$ = cathode potential, $E_{sc}$ = constant potential, $K_c$ = cathode polarization constant per unit active material current density, $i_{am}$ = active material current density. In case of porous electrodes (which is mostly the case for Li-Ion batteries), $i_{am}$ can be given as,

$$i_{am} = \left(\frac{Q_c}{Q_c - it}\right) i \quad \text{(D1.3)}$$

Where, Qc = amount of active cathode material (Ah/m), t = observed time interval during discharge, $it = \int_0^t i dt$, is the current consumption over time interval *t*. Substituting (D1.3) into (D1.2) yields,

$$E_c = E_{sc} - K_c \left(\frac{Q_c}{Q_c - it}\right) i \quad \text{(D1.4)}$$

Similarly, for anode potential,

$$E_a = E_{sa} - K_a \left(\frac{Q_a}{Q_a - it}\right) i \quad \text{(D1.5)}$$

For a well-designed balanced cell, the potentials across cathode and anode are equal due to the balanced number of active materials per unit area across the terminals. This assumes that $Q_a \cong Q_c$. Therefore, simplifying (D1.4) and (D1.5) to gain a simple effective equation for both the terminals.

$$E = E_s - K \left(\frac{Q}{Q - it}\right) i \quad \text{(D1.6)}$$

Where, $E = E_a + E_c$ is the cell potential (neglecting internal resistance), $E_s = E_{sa} + E_{sc}$ is the constant potential, $K = K_a + K_c$ is polarization coefficient ($\Omega cm^2$).

It is very important to note that *Q* is determined by the amount of available active material per unit area on the controlling electrode, which fails first. If the internal resistance of the battery is considered, there must be some potential drop across it. The

internal resistance cannot be ignored as it will be used to model the effect of temperature on the battery surface. Therefore, the potential drop due to internet resistance be accounted for, as,

$$E = E_s - K\left(\frac{Q}{Q-it}\right)i - Ni \tag{D1.7}$$

Where N, is the internal resistance per unit area ($\Omega cm^2$). Using Equation (D1.7), a simplified 1RC equivalent battery model can be modeled. This model can adapt to internal resistance effects for no initial potential drops. However, in real-world applications, batteries are highly non-linear devices and always experience initial potential drop. This non-linear potential drop can be evaluated by using empirical constants whose value strictly depend on experimental values and may differ from battery to battery (types and internal chemistry). Including initial potential drop resistance to extend Equation (D1.7), yields,

$$E = E_s - K\left(\frac{Q}{Q-it}\right)i - Ni + A^{(-BQ^{-1}it)} \tag{D1.8}$$

The above-mentioned model was further simplified by Tremblay by including an expanded version of polarization resistance whose effect was measured over time. By doing so, the filtered current through polarization resistance helped to accurately determine the SoC during a discharge cycle. This helped to achieve the battery voltage equation, as given in (D1.9)

$$V_{batt} = E_o - K\left(\frac{Q}{Q-it}\right)it - R.i + Aexp(-B.it) - K\left(\frac{Q}{Q-it}\right).i^* \tag{D1.9}$$

Where, $V_{batt}$ is the battery voltage (V), $E_o$ is the battery constant voltage (V), K is polarization constant (V/Ah), Q is battery capacity (Ah), $it = \int_0^t idt$ is the actual battery charge (Ah), A is exponential zone amplitude (V), B is exponential zone time constant inverse ($Ah^{-1}$), R is internal resistant **Ω**, *i* is battery current (A), and $i^*$ is filtered current (A). This model works with the following assumptions:

- The internal resistance remains constant during entire discharge cycle
- The battery capacity remains same with current (No Peukert effect)
- No self-discharging (operating ranges are considered between 0 – 50˚C)
- No effect of temperature on battery behavior
- No battery memory effect (signifying that it can fully charge and retain up to 100%)

The internet resistance of the battery effects the output voltage as well as the capacity of the battery, which can be approximated in terms of SoC and lifetime respectively. As for most of the battery types, the steady state region shows almost a linear region (if discharged with constant current). Over this zone the variations in voltage are very minor and the effect of voltage drops due to internal resistance are insignificant. However, the corresponding changes in SoC exhibits a larger change. It is also interesting to know that the effect of temperature, changes the internal resistance which can in part affect the output voltage. In this model extension, a single cell with small constant load current is assumed, whereas secondary phenomenon such as internal temperature, irreversible heat generation and thermal runaway can be neglected. Therefore, the Tremblay model also accounts for the effect of internal resistance on output voltage and efficiency, which can be given as,

$$\eta = 1 - \frac{I_{nom} - R.I_{nom}}{V_{nom}.I_{nom}} \quad \text{(D1.10)}$$

$I_{nom}$ can be calculated from experimental observations, where a fixed discharge current is maintained for the entire cycle. This makes $I_{nom}$ calculation very simple and could be approximated by equation (D1.11).

$$I_{nom} = Q_{nom}.(C - rate\ per\ hour) \quad \text{(D1.11)}$$

Substituting (D1.11) into (D1.10) to get ($\eta$),

$$\eta = 1 - \frac{(C - rate\ per\ hour).R.Q_{nom}}{V_{nom}} \quad \text{(D1.12)}$$

Using the previous equations, simplified modelling equations for various battery types can be achieved.

**a) Lead-Acid**

$$V_{batt} = E_o - R.i - K\left(\frac{Q}{Q-it}\right).(it + i^*) + Exp(t) \quad \text{(D1.13a)}$$

**b) Li-Ion**

$$V_{batt} = E_o - R.i - K\left(\frac{Q}{Q-it}\right).(it + i^*) + AExp(-B.it) \quad \text{(D1.13b)}$$

**c) NiMH and NiCd**

$$V_{batt} = E_o - R.i - K\left(\frac{Q}{Q-it}\right).(it + i^*) + Exp(t) \quad \text{(D1.13c)}$$

Equations D1.13 (a-c) represent battery discharge models for four very common types normally used in low-power electronics applications. However, these models do not estimate the effect of temperature on these batteries. In case of WSNs, the nodes are deployed in various scenarios where the landscape and climate zone ranges from 0°C to 50°C mostly. In other conditions external environmental factors such as rain and humidity can also affect the battery discharge patterns, thus its capacity. Due to mostly single cell usage, low discharge currents, low duty-cycle discharge time intervals and almost fixed loads for each stage, the internal heat generation effects are ignored. Therefore, in addition to previous model equations, the ambient temperature effects on battery surface temperatures are introduced in Equations D1.13 (a-c) to increase model accuracy and incorporate the effect of temperature changes. This leads to temperature dependent model equations D1.14 (a-c).

$$V_{batt} = E_o(T_{surf}) - R(T_{surf}).i - K\left(\frac{QT_{surf}}{QT_{surf}-it}\right).(it + i^*) + Exp(t) \quad \text{(D1.14a)}$$

$$V_{batt} = E_o(T_{surf}) - R(T_{surf}).i - K\left(\frac{QT_{surf}}{QT_{surf}-it}\right).(it + i^*) + AExp(-B.it) \quad \text{D1.14b)}$$

$$V_{batt} = E_o(T_{surf}) - R(T_{surf}).i - K\left(\frac{QT_{surf}}{QT_{surf}-it}\right).(it + i^*) + Exp(t) \quad \text{(D1.14c)}$$

With,

$$E_oT_{surf} = E_o\left|T_{amb} + \frac{\partial E}{\partial T_{surf}}(T_{surf} - T_{amb}\right)$$

$$KT_{surf} = K|T_{amb}.exp\left(\propto\left(\frac{1}{T_{surf}} - \frac{1}{T_{amb}}\right)\right)$$

$$QT_{surf} = Q\left|T_{surf} + \frac{\Delta Q}{\Delta T_{surf}}.(T_{surf} - T_{amb}\right)$$

$$RT_{surf} = R|T_{amb}.exp\left(\beta\left(\frac{1}{T_{surf}} - \frac{1}{T_{amb}}\right)\right)$$

Where, $T_{surf}$ is the battery surface temperature, $T_{amb}$ is the ambient temperature, $\propto$ is the Arrhenius rate constant for polarization resistance, and $\beta$ is the Arrhenius rate constant for internal resistance. The battery surface temperature is calculated during complete discharge cycle with constant current load. Equations D1.14 (a-c) were used in energy estimation calculations that were compared with Contiki OS's energy module, as well as KiBAM and TKiBAM models for approximations.

## Appendix E: Effect of CCA adjustments on Radio Duty Cycle of nodes

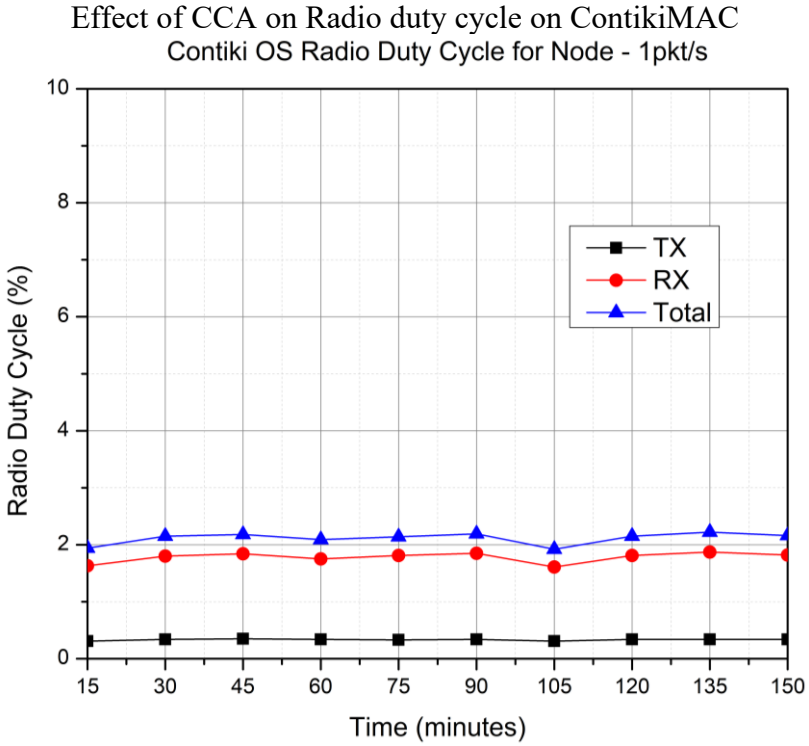



**FIGURE E1: EFFECT ON DUTY CYCLE FOR NODE AT 1PKT/S**

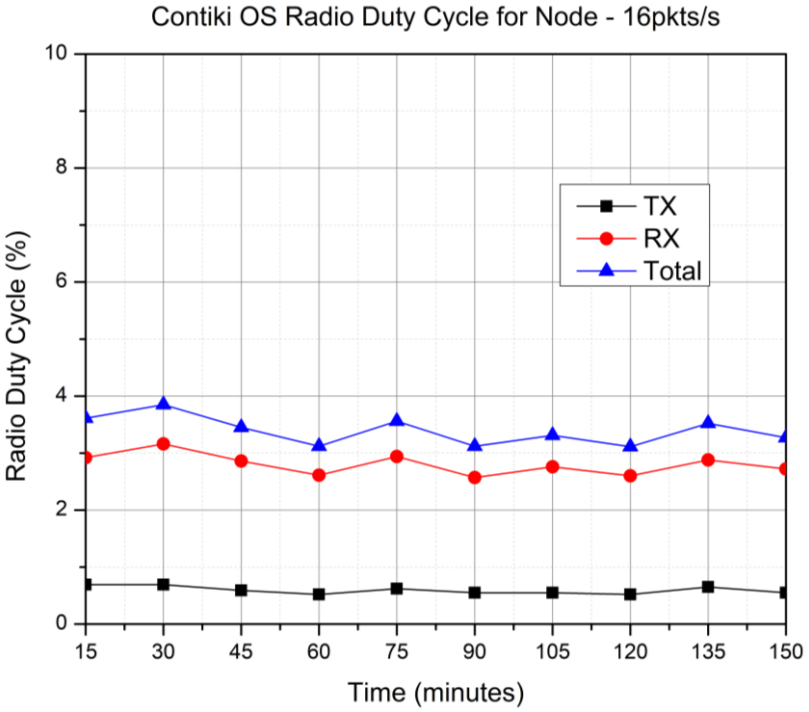


**FIGURE E2: EFFECT ON DUTY CYCLE FOR NODE AT 16PKT/S**

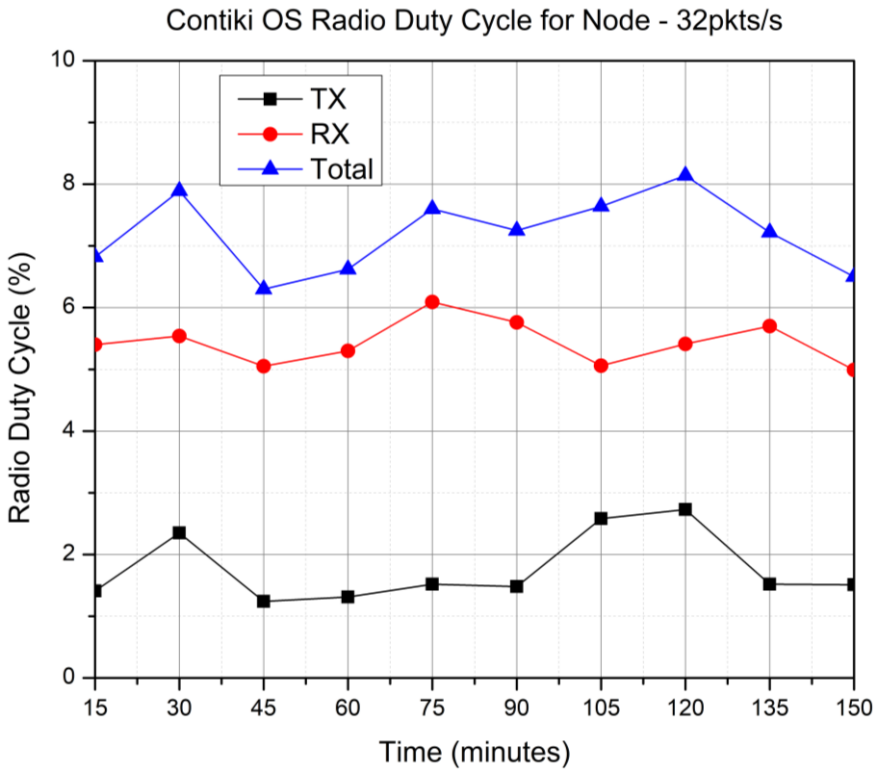



**FIGURE E3: EFFECT ON DUTY CYCLE FOR NODE AT 32PKT/S**

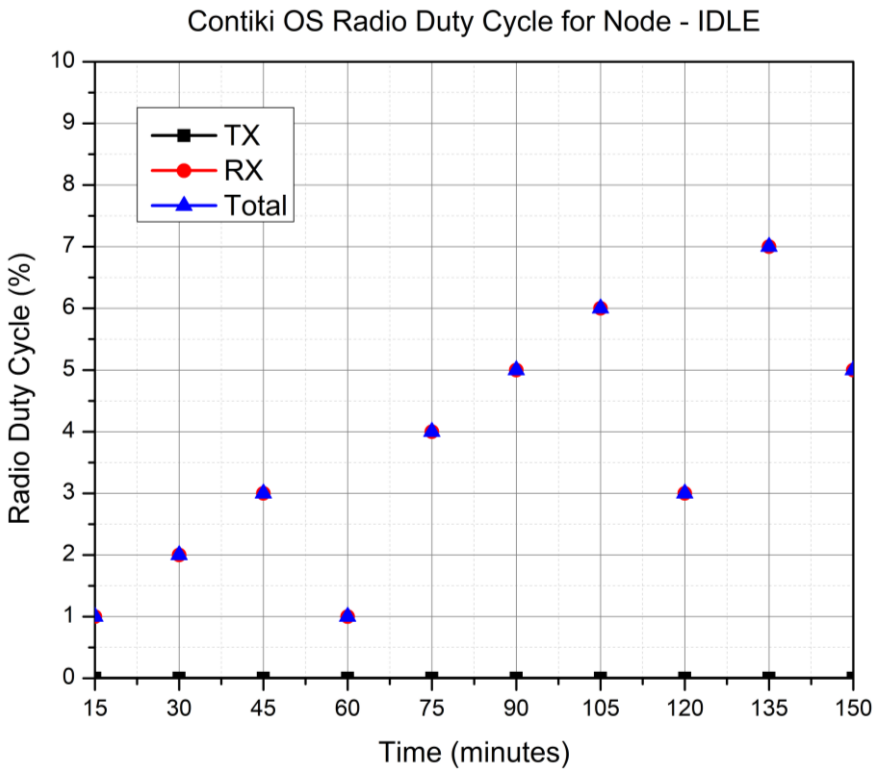



**FIGURE E4: EFFECT ON DUTY CYCLE FOR NODE WHEN IDLE**

Effect of CCA on Radio duty cycle on KiBAM scheme

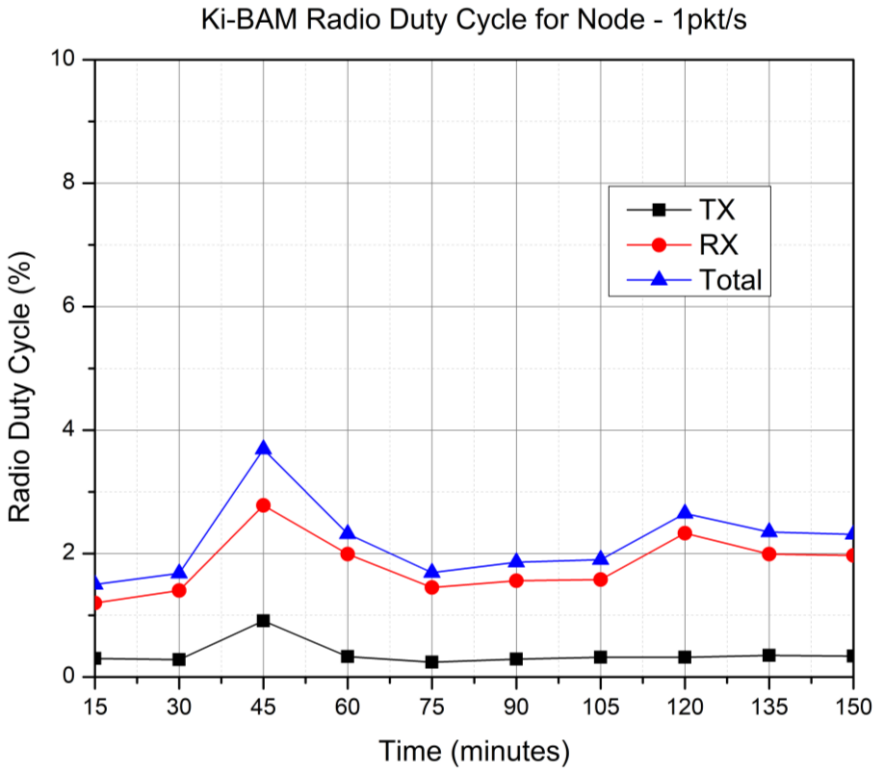



**FIGURE E5: EFFECT ON DUTY CYCLE FOR NODE AT 1PKT/S**

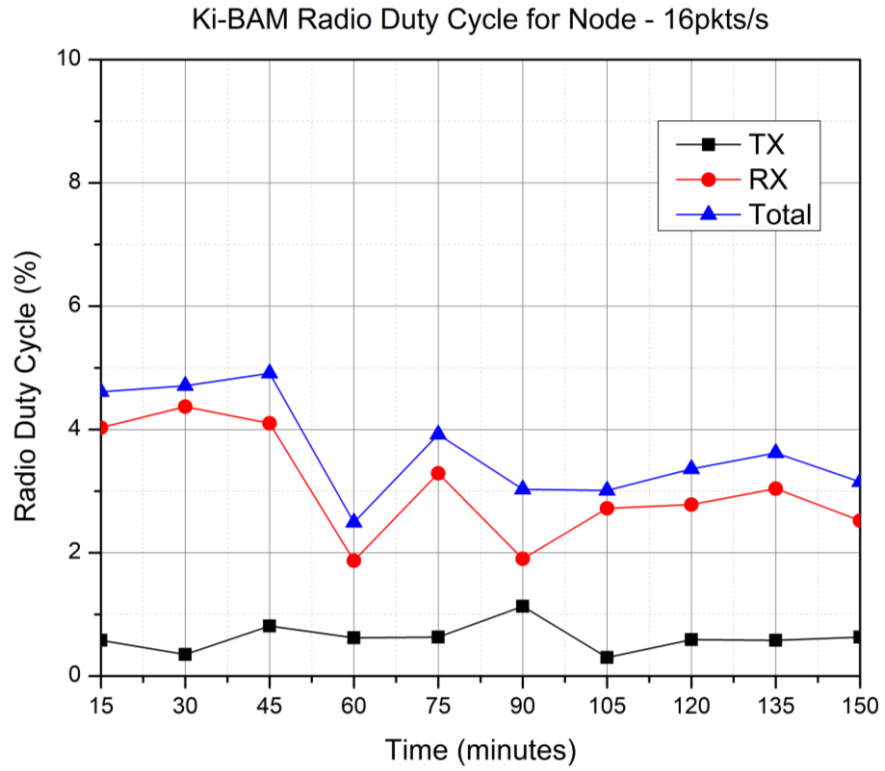



**FIGURE E6: EFFECT ON DUTY CYCLE FOR NODE AT 16PKT/S**

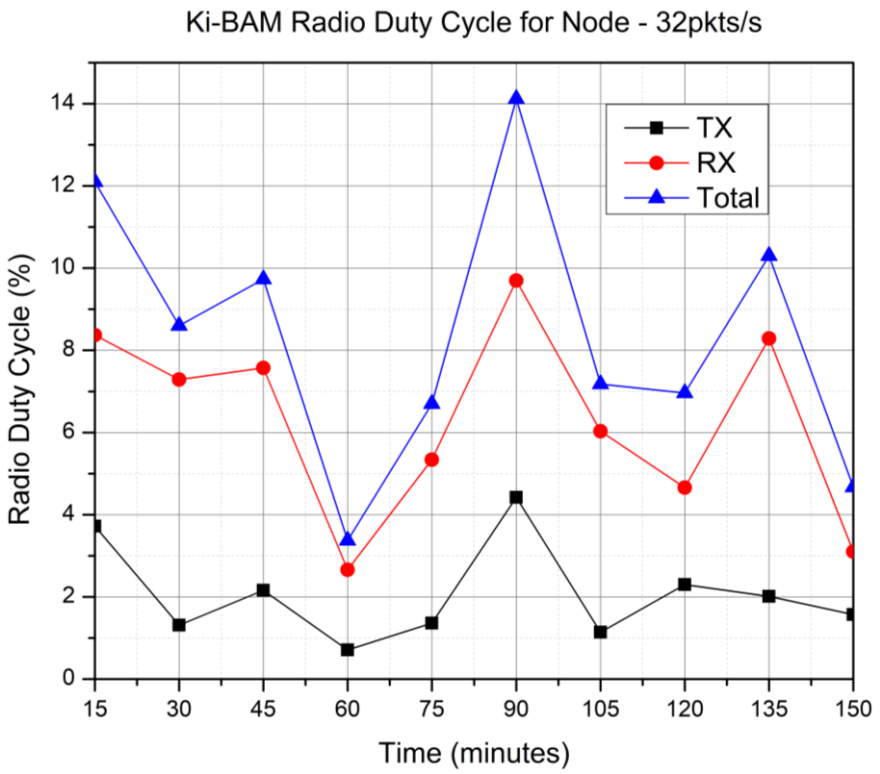



**FIGURE E7: EFFECT ON DUTY CYCLE FOR NODE AT 32PKT/S**

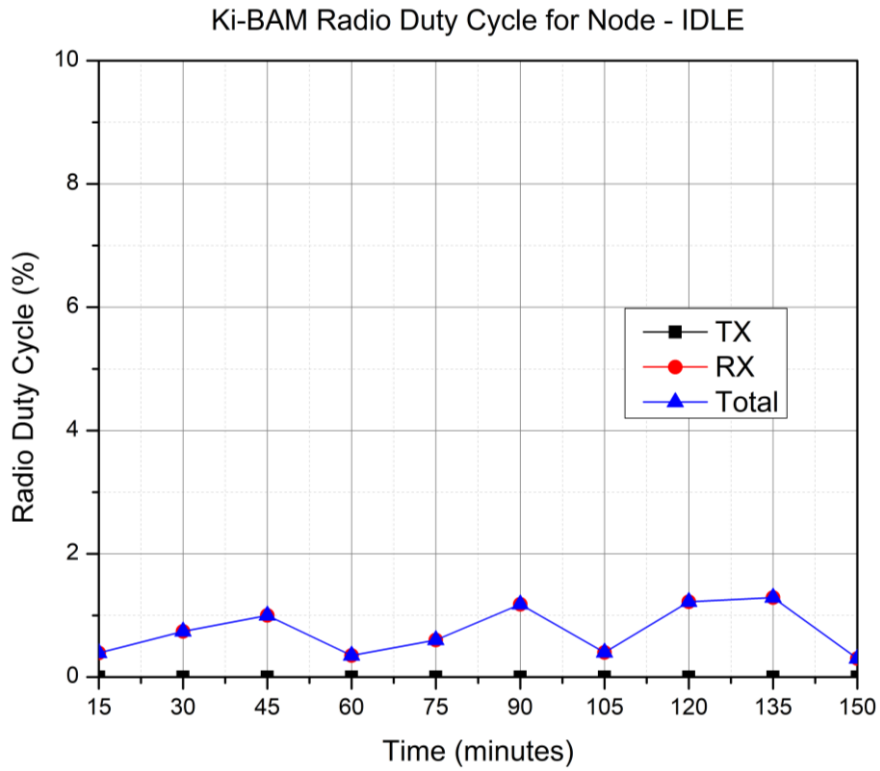



**FIGURE E8: EFFECT ON DUTY CYCLE FOR NODE WHEN IDLE**

Effect of CCA on Radio duty cycle for T-KiBAM scheme

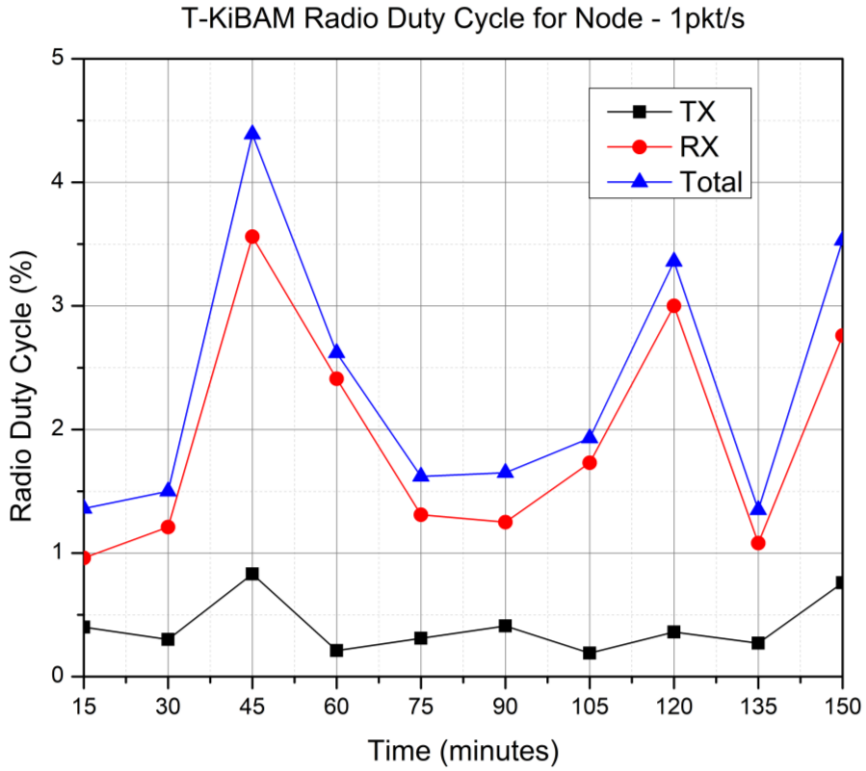



**FIGURE E9: EFFECT ON DUTY CYCLE FOR NODE AT 1PKT/S**

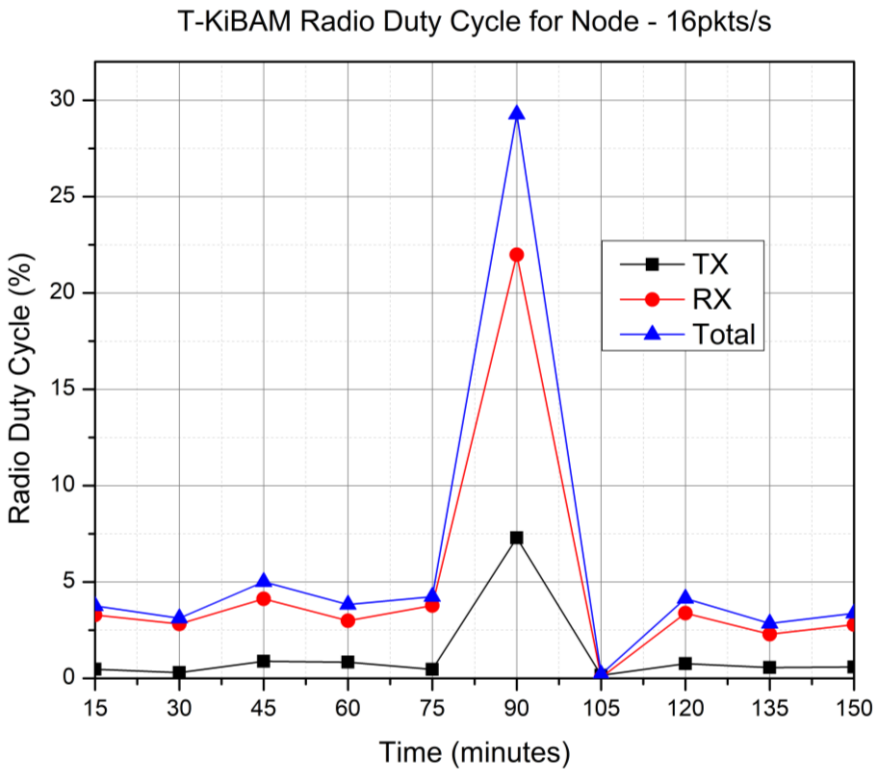



**FIGURE E10: EFFECT ON DUTY CYCLE FOR NODE AT 16PKT/S**

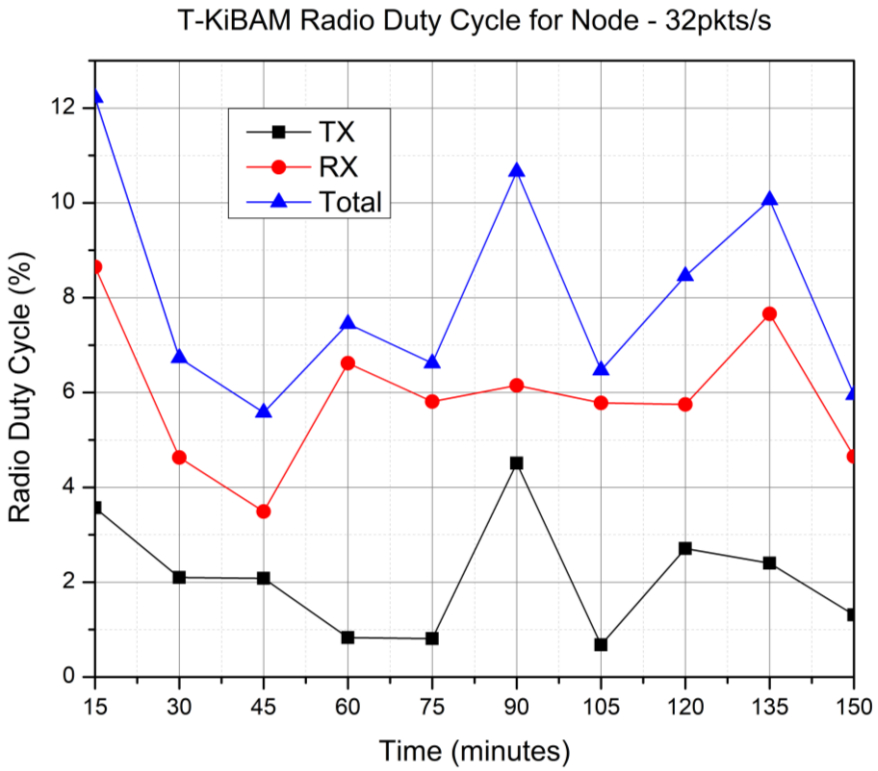



**FIGURE E11: EFFECT ON DUTY CYCLE FOR NODE AT 32PKT/S**

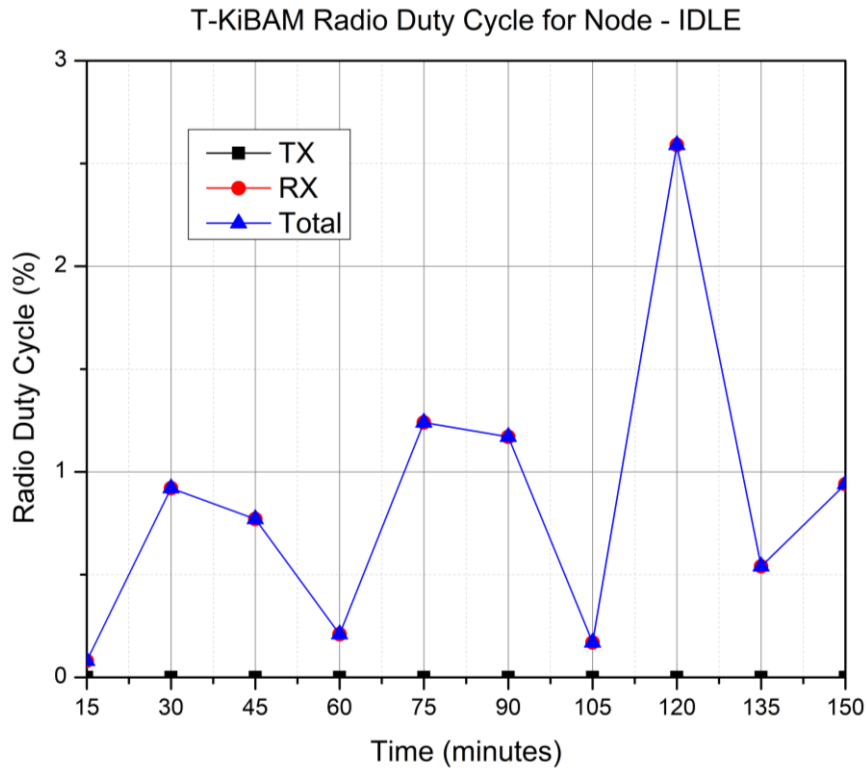



**FIGURE E12: EFFECT ON DUTY CYCLE FOR NODE WHEN IDLE**

Effect of CCA on Radio duty cycle for proposed GPR-RBF scheme

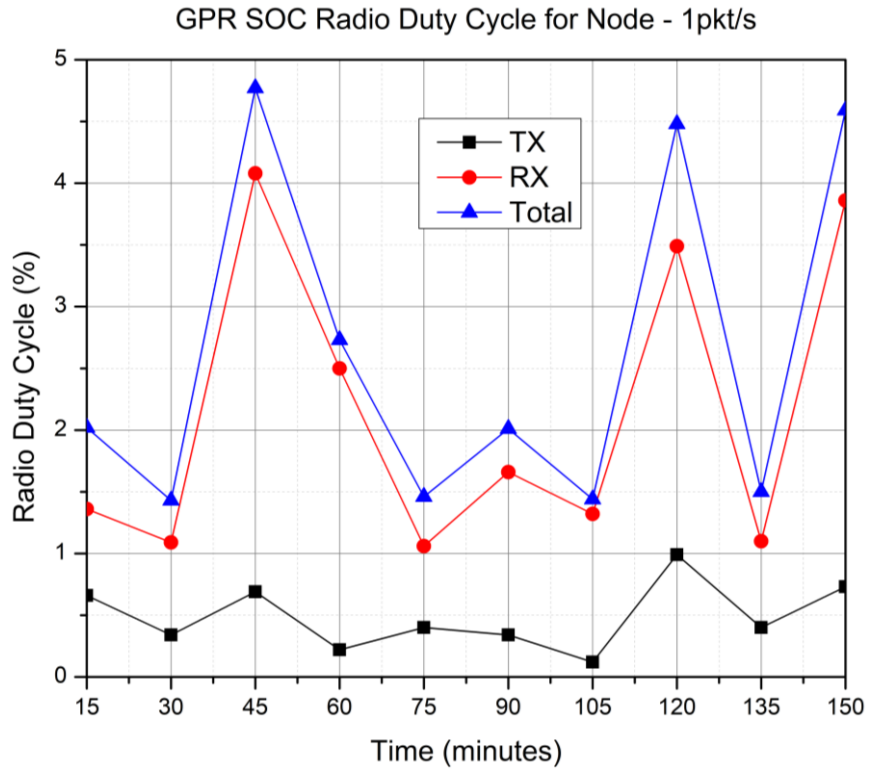



**FIGURE E13: EFFECT ON DUTY CYCLE FOR NODE AT 1PKT/S**

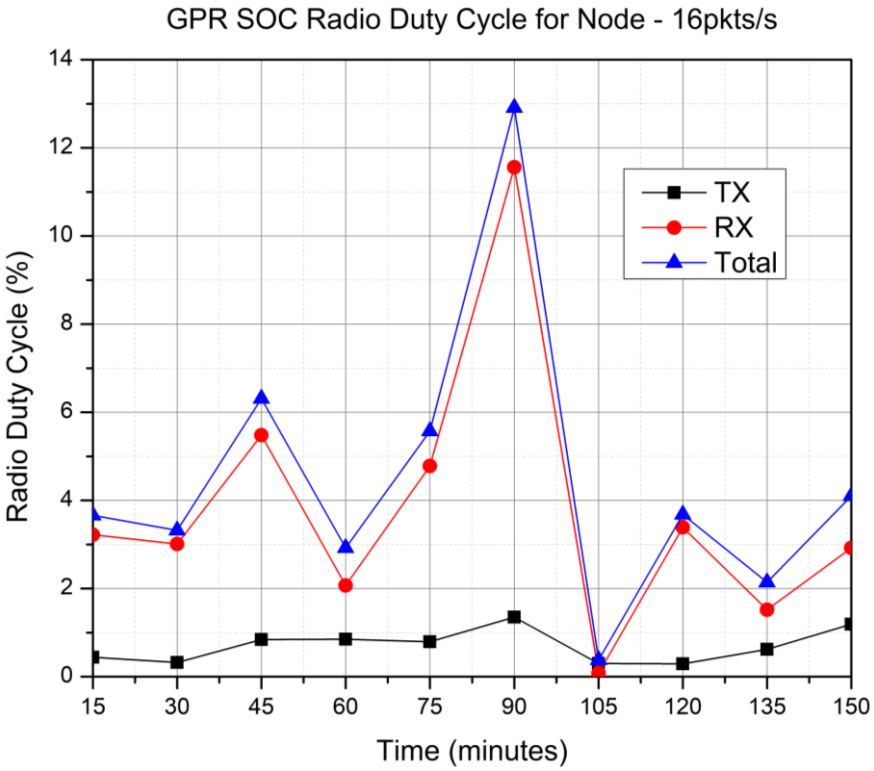



**FIGURE E14: EFFECT OF DUTY CYCLE FOR NODE AT 16PKT/S**

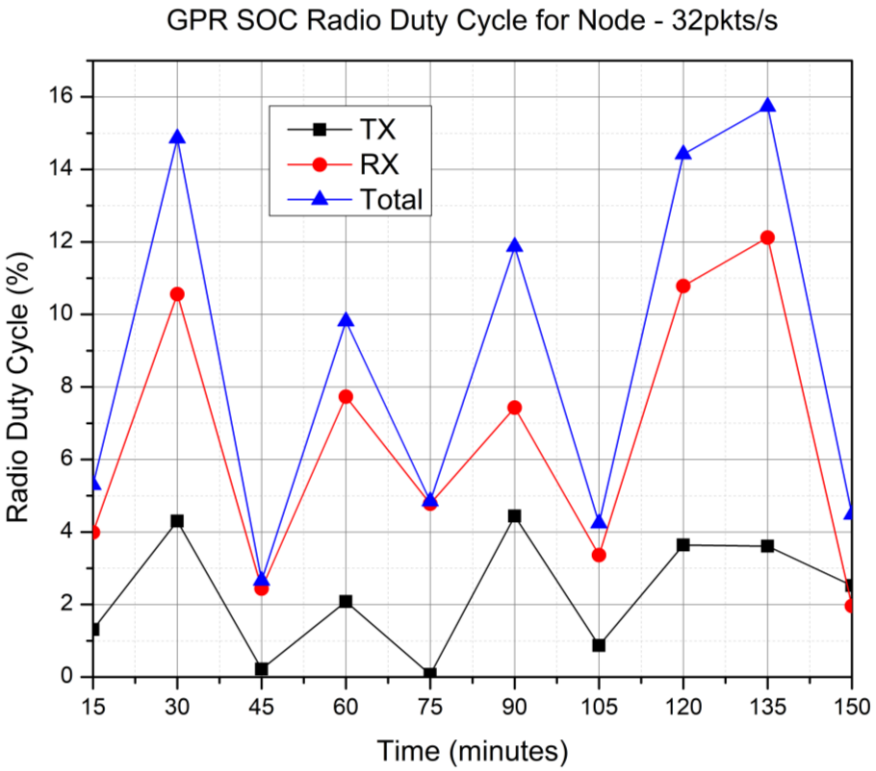



**FIGURE E15: EFFECT ON DUTY CYCLE FOR NODE AT 32PKT/S**

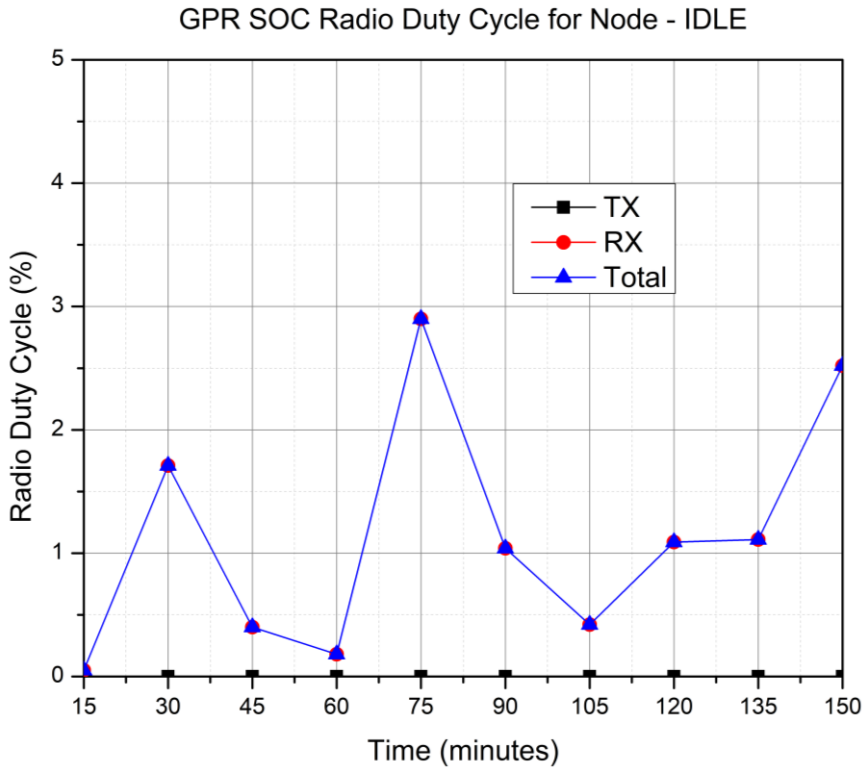



**FIGURE E16: EFFECT ON DUTY CYCLE FOR NODE WHEN IDLE**

## Appendix F: Effect of CCA adjustments on node energy consumption

Effect on node energy consumption for ContikiMAC

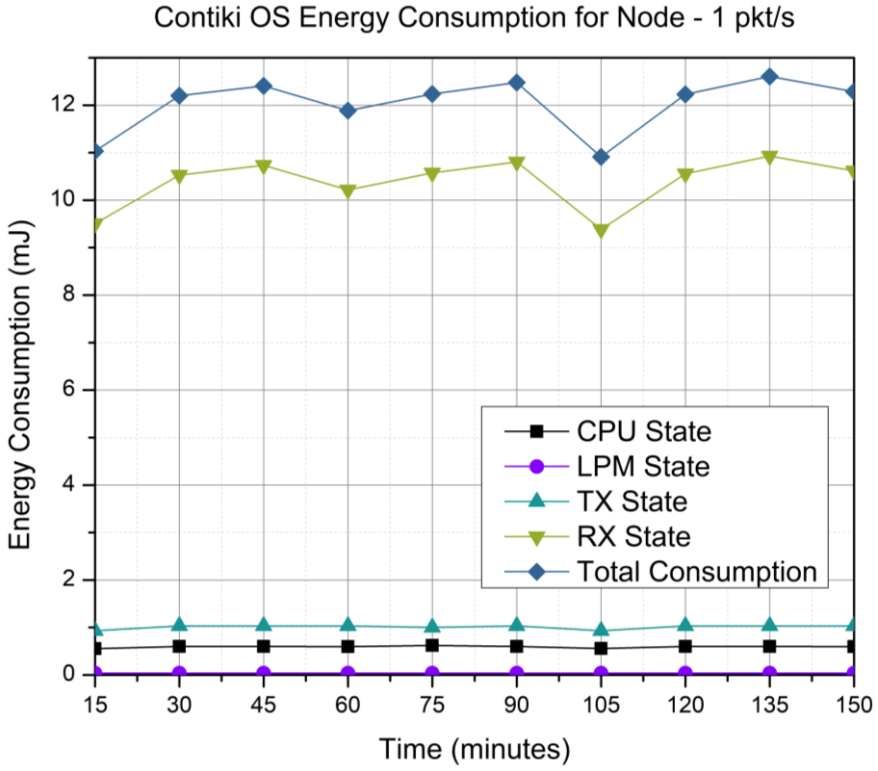



**FIGURE F1: ENERGY CONSUMPTION FOR NODE AT 1PKT/S**

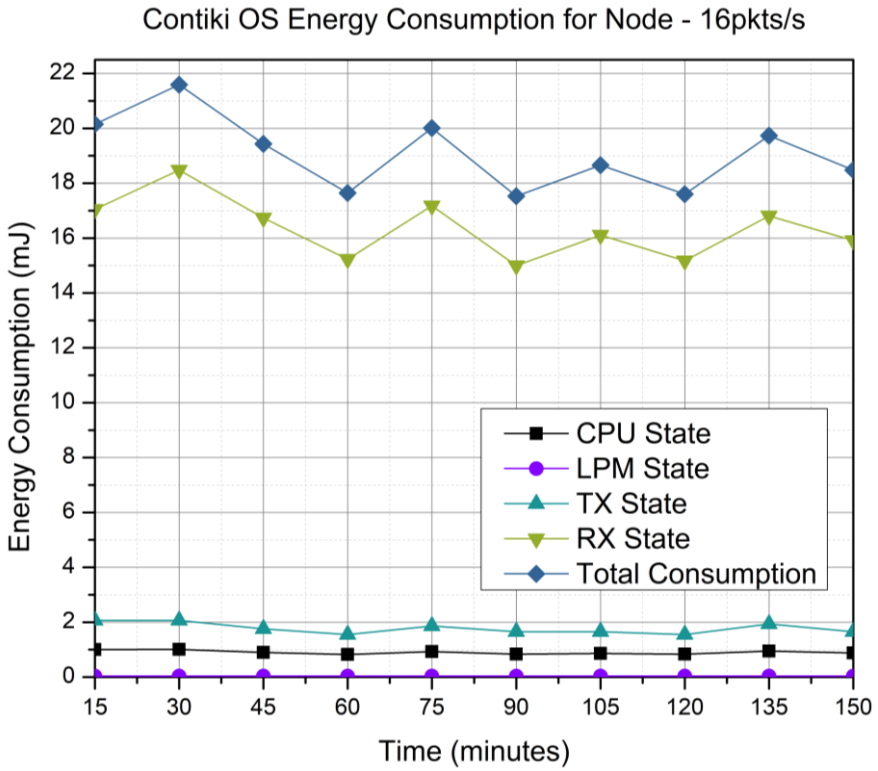



**FIGURE F2: ENERGY CONSUMPTION FOR NODE AT 16PKT/S**

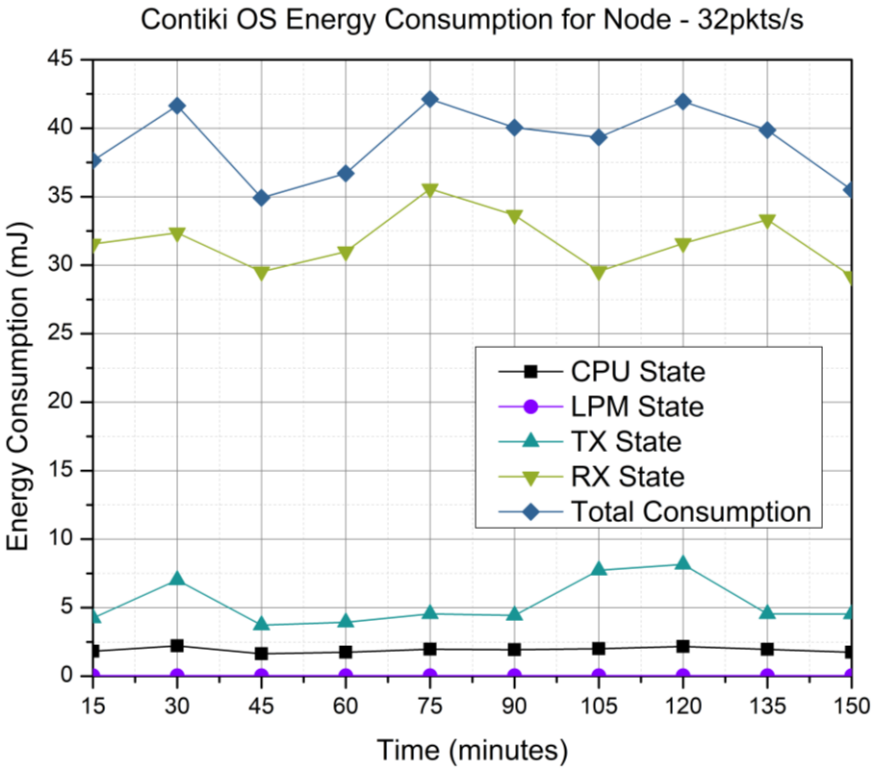



**FIGURE F3: ENERGY CONSUMPTION FOR NODE AT 32PKT/S**

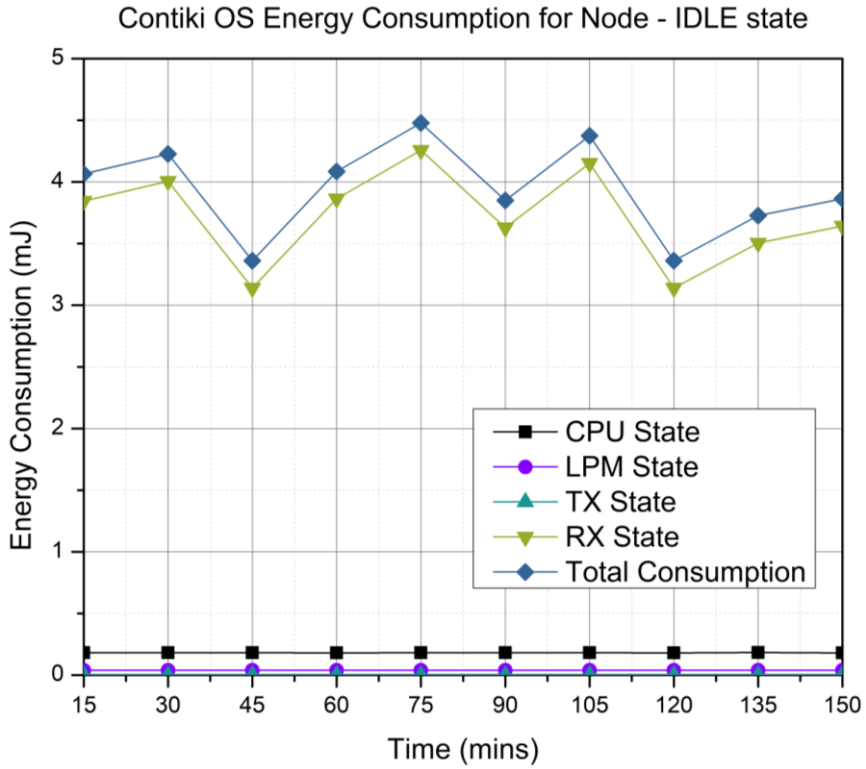



**FIGURE F4: ENERGY CONSUMPTION FOR NODE WHEN IDLE**

Effect on node energy consumption for KiBAM scheme

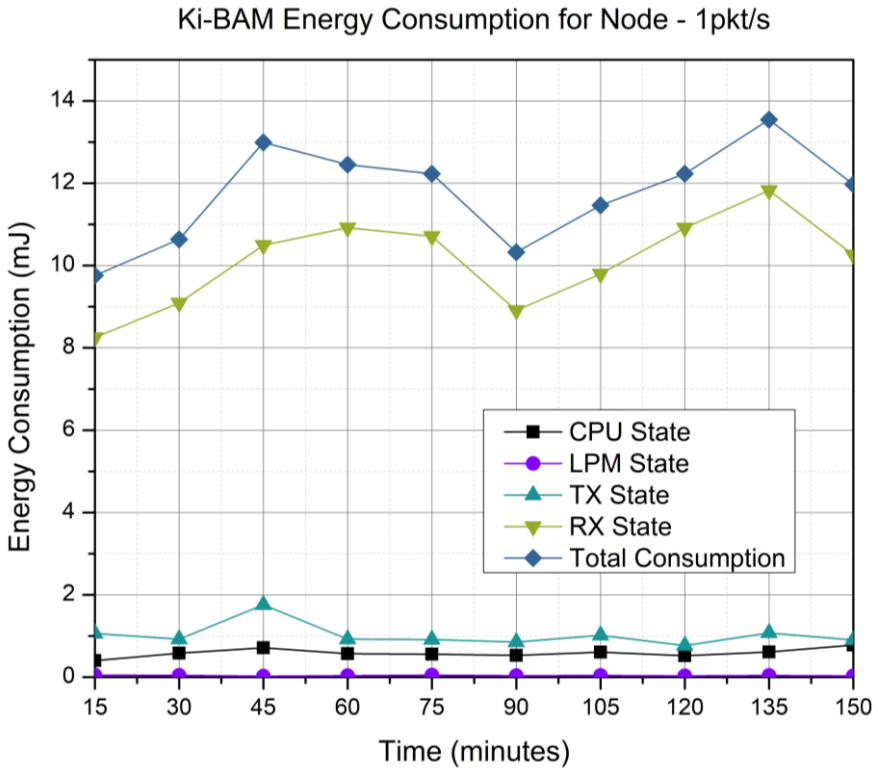



**FIGURE F5: ENERGY CONSUMPTION FOR NODE AT 1PKT/S**

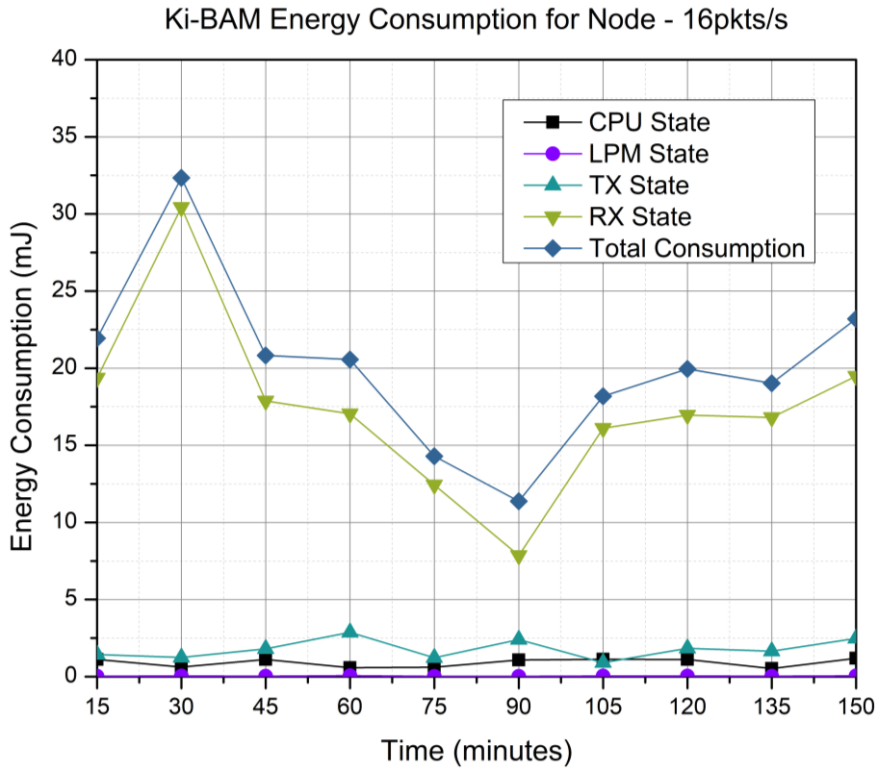



**FIGURE F6: ENERGY CONSUMPTION FOR NODE AT 16PKT/S**

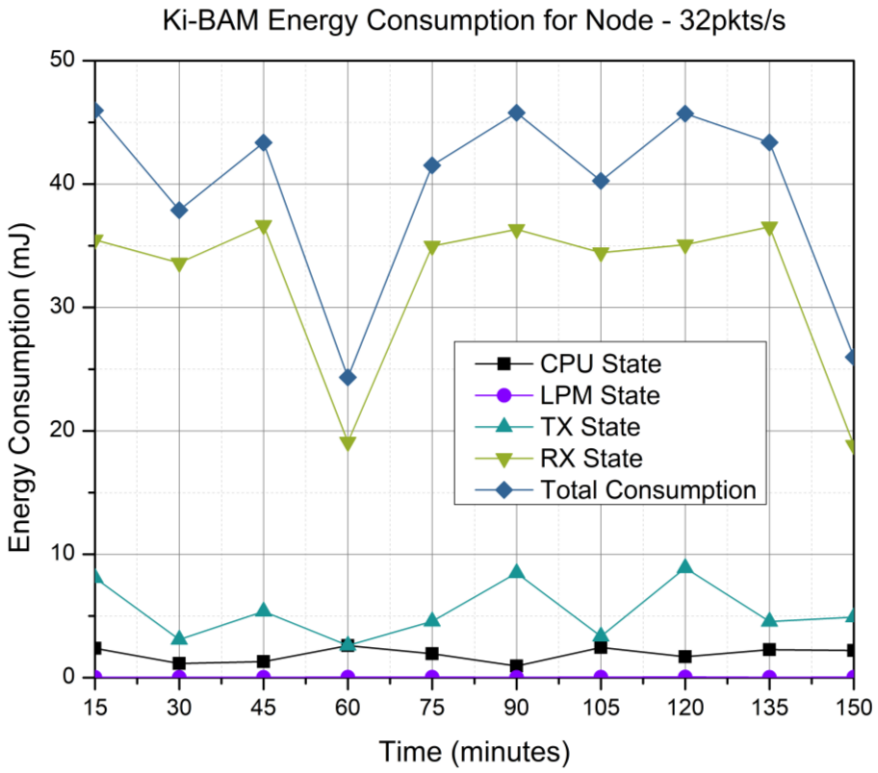



**FIGURE F7: ENERGY CONSUMPTION FOR NODE AT 32PKT/S**

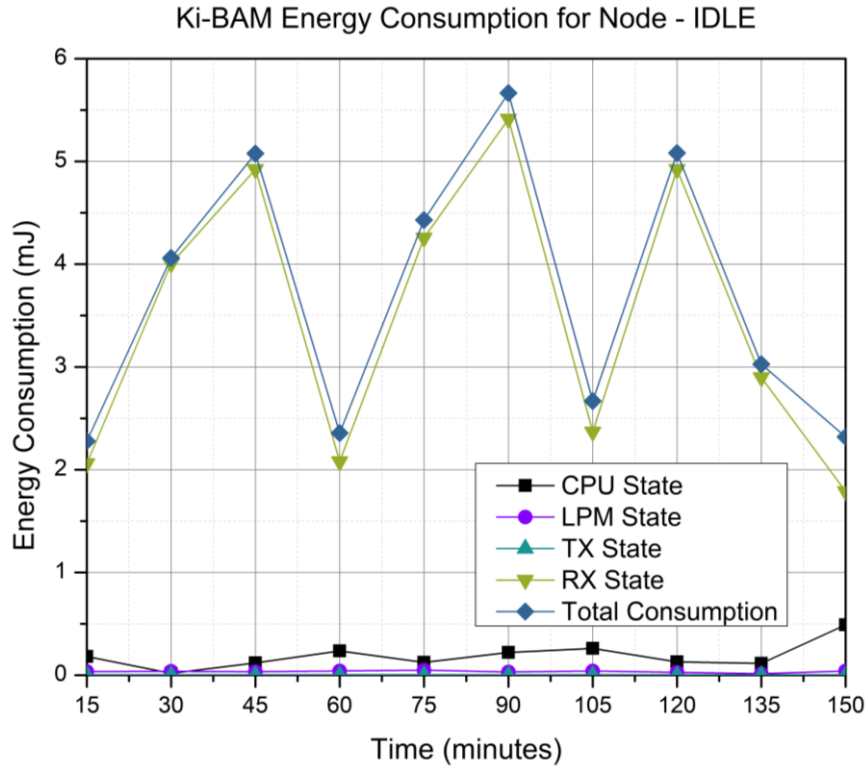



**FIGURE F8: ENERGY CONSUMPTION FOR NODE WHEN IDLE**

Effect on node energy consumption for T-KiBAM scheme

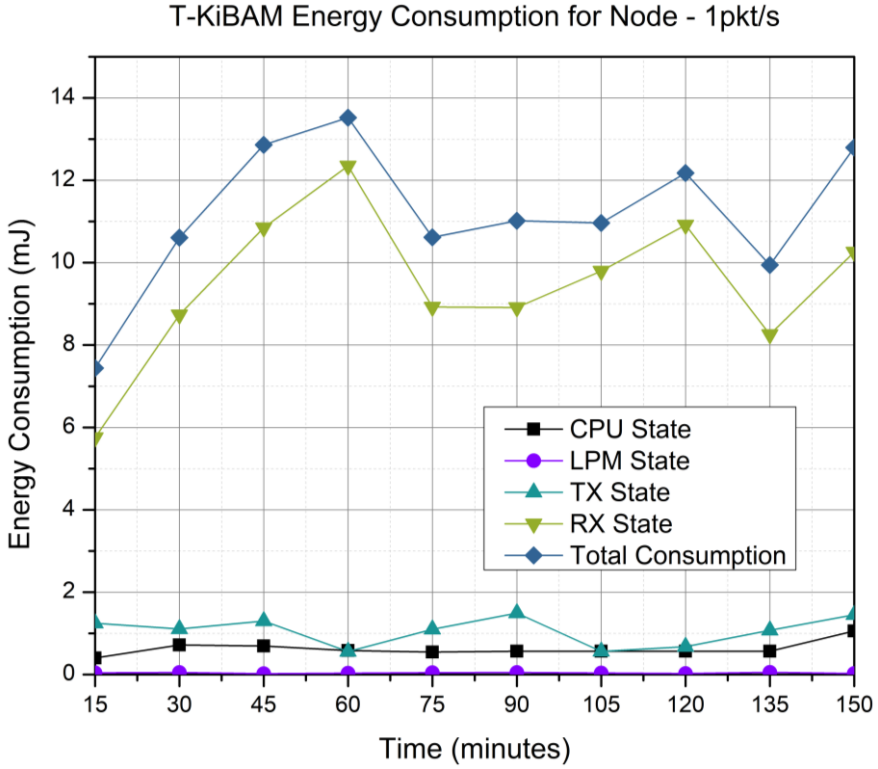



**FIGURE F9: ENERGY CONSUMPTION FOR NODE AT 1PKT/S**

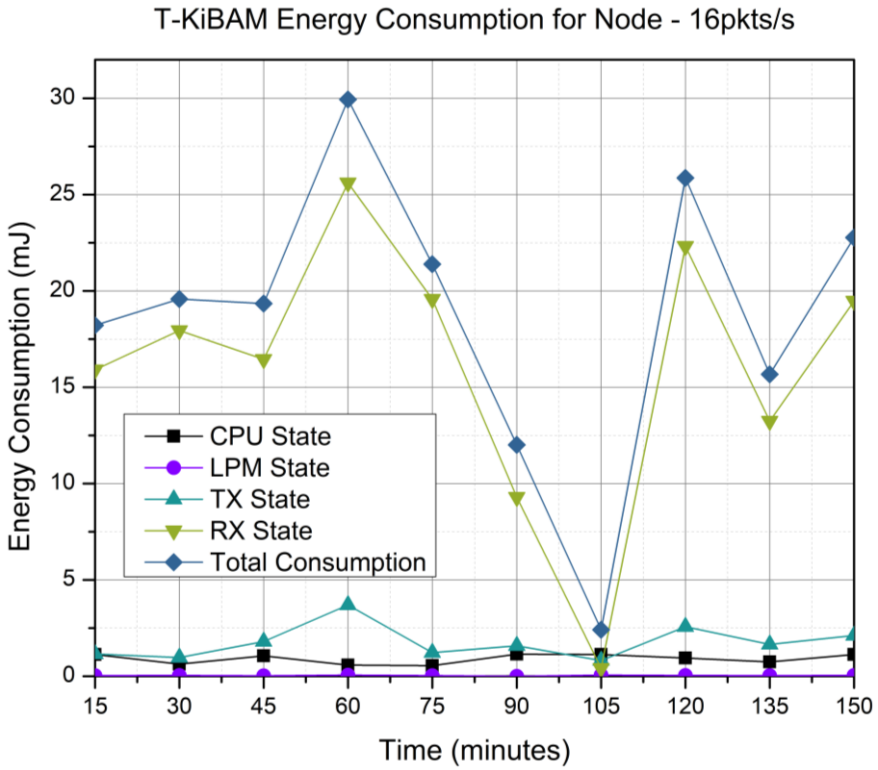



**FIGURE F10: ENERGY CONSUMPTION FOR NODE AT 16PKT/S**

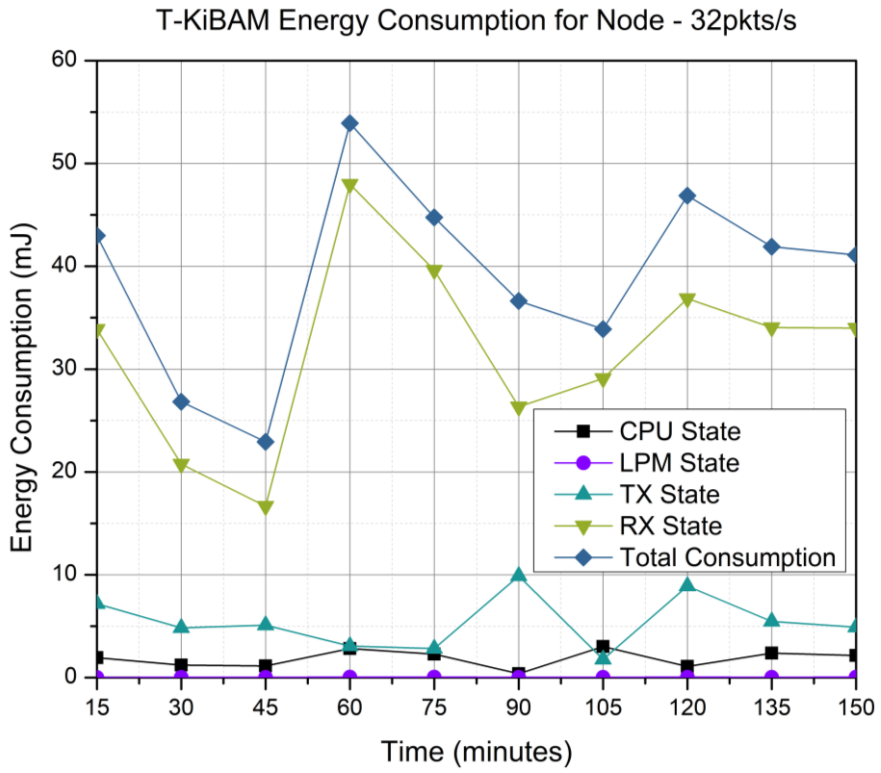



**FIGURE F11: ENERGY CONSUMPTION FOR NODE AT 32PKT/S**

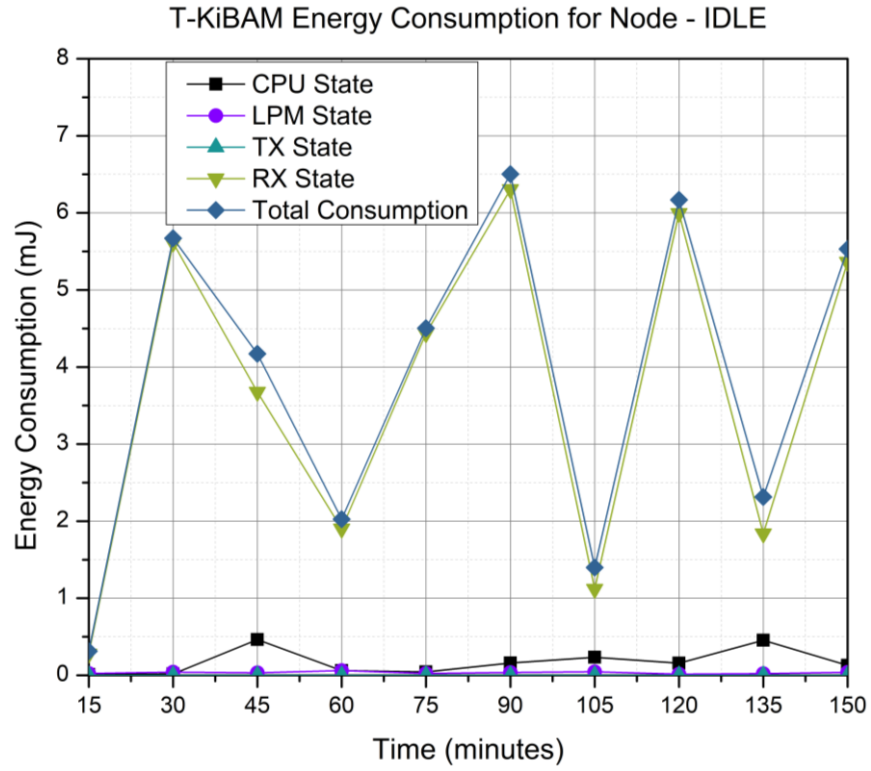



**FIGURE F12: ENERGY CONSUMPTION FOR NODE WHEN IDLE**

Effect on node energy consumption on proposed GPR-RBF Scheme

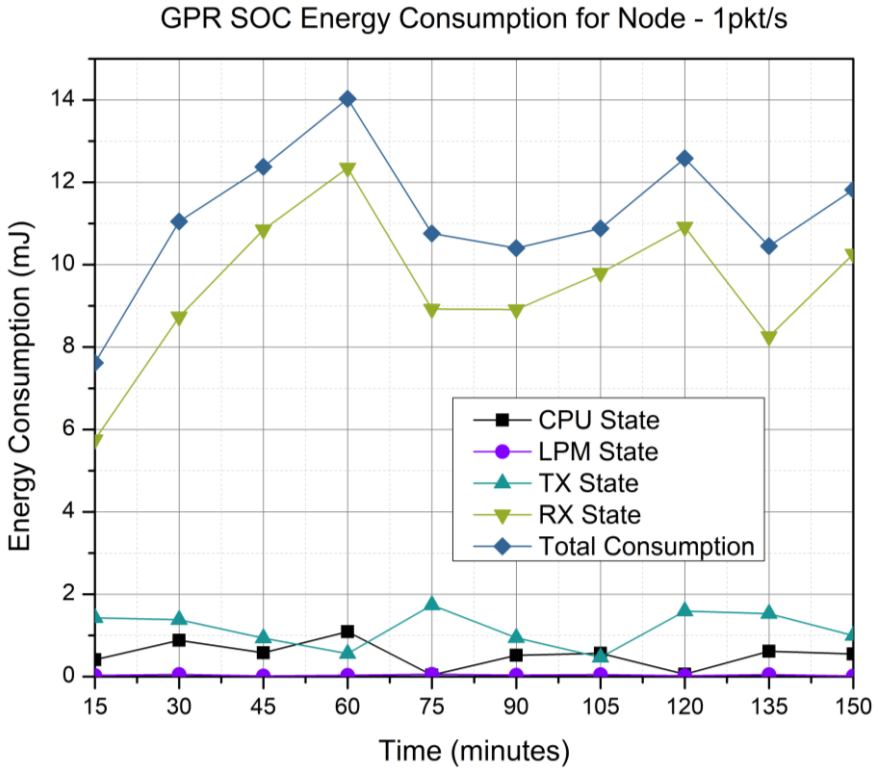



**FIGURE F13: ENERGY CONSUMPTION FOR NODE AT 1PKT/S**

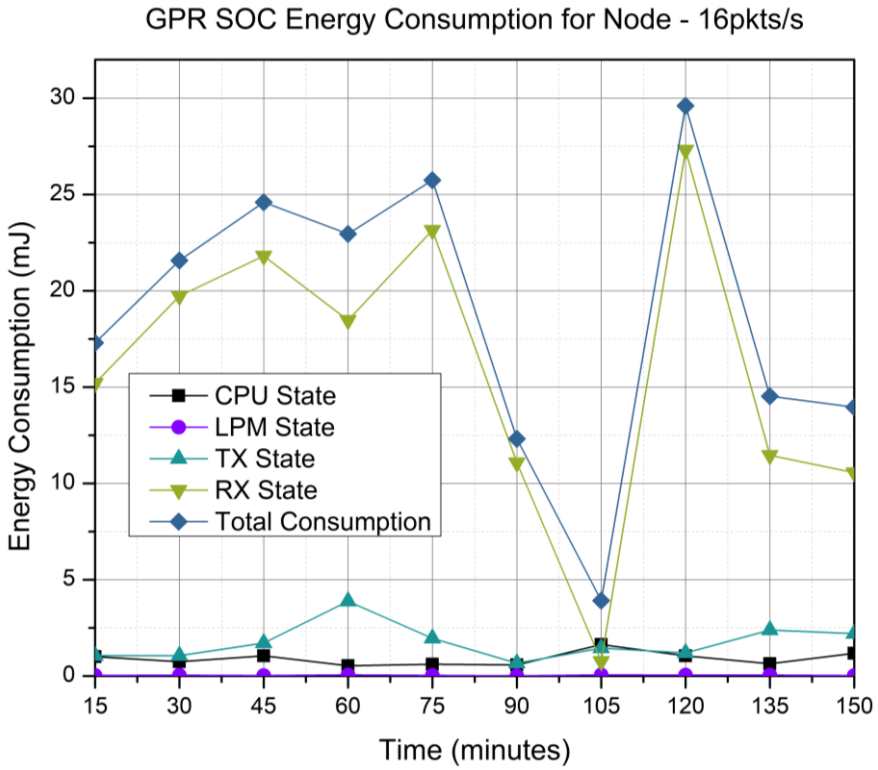



**FIGURE F14: ENERGY CONSUMPTION FOR NODE AT 16PKT/S**

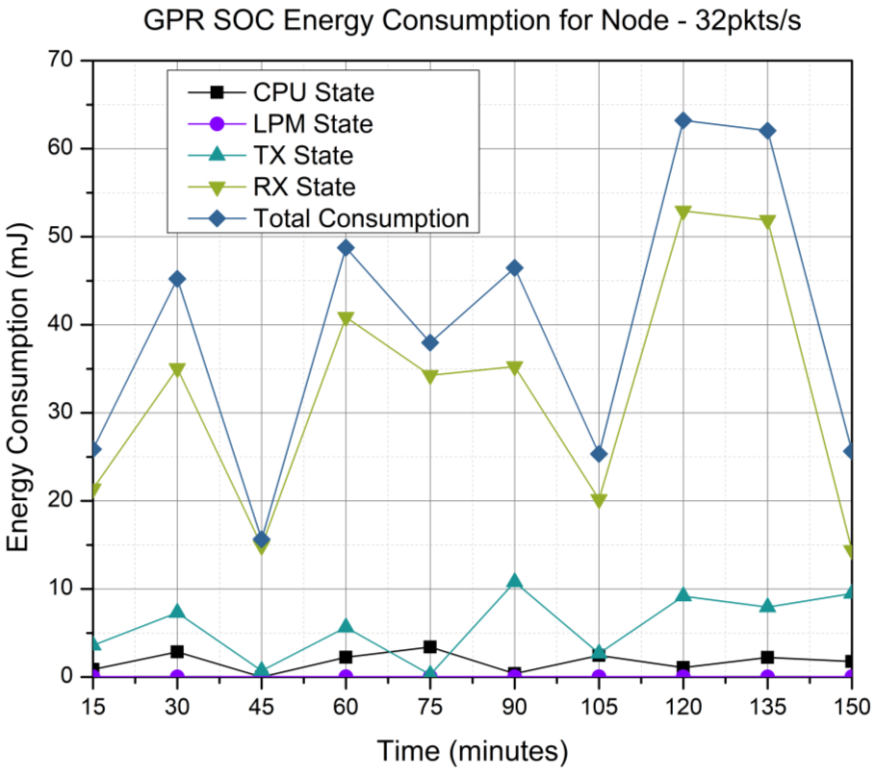



**FIGURE F15: ENERGY CONSUMPTION FOR NODE AT 32PKT/S**

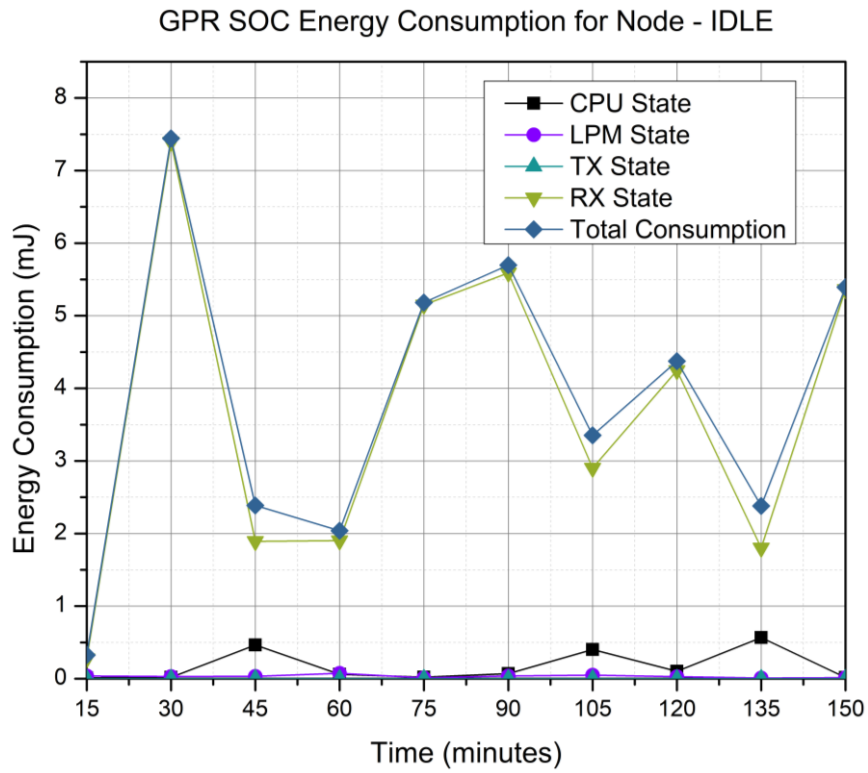



**FIGURE F16: ENERGY CONSUMPTION FOR NODE WHEN IDLE**

## LIST OF PUBLICATIONS

### 1. Journals:

1 Ali Omer, Mohamad Khairi Ishak, Ashraf Bani Ahmad, Mohd Fadzli Mohd Salleh, Chia Ai Ooi, Muhammad Firdaus Akbar Jalaludin Khan, and Imran Khan, "On-line WSN SoC estimation using Gaussian Process Regression: An Adaptive Machine Learning approach", Alexandria Engineering Journal, 61(12), 9831-9848.

(Indexed: ISI, Rank: Q1, Impact Factor: 3.732)

2 Ali, O., Ishak, M. K., Ooi, C. A., & Bhatti, M. K. L. (2022). Battery characterization for wireless sensor network applications to investigate the effect of load on surface temperatures. Royal Society open science, 9(2), 210870.

(Indexed: ISI, Rank: Q1, Impact Factor: 2.963)

3 Ali, O., Ishak, M. K., Bhatti, M. K. L., Khan, I., & Kim, K. I. (2022). A Comprehensive Review of Internet of Things: Technology Stack, Middlewares, and Fog/Edge Computing Interface. Sensors, 22(3), 995.

(Indexed: ISI, Rank: Q2, Impact Factor: 3.576)

4 Ali, O., & Ishak, M. K. (2021). Energy Efficient Scheme for Wireless Sensor Networks based on CONTIKIMAC Protocol. Journal of Engineering Science and Technology, 16(6), 4666-4686.

(Indexed: Scopus, Rank: Q3)

5 Omer, A., Khairi, M., & Ishak, M. K. L. (2022). A Machine Learning Approach for Early COVID-19 Symptoms Identification. Computers, Materials, & Continua, 3803-3820.

(Indexed: ISI, Rank: Q2, Impact Factor: 3.772)

6 Ali, O., Ishak, M. K., & Bhatti, M. K. L. (2021). Emerging IoT domains, current standings, and open research challenges: a review. PeerJ Computer Science, 7, e659.

(Indexed: ISI, Rank: Q2, Impact Factor: 1.392)

7 Ali, O., Ishak, M. K., & Bhatti, M. K. L. (2021). Early COVID-19 symptoms identification using hybrid unsupervised machine learning techniques. Computers, Materials and Continua, 747-766.

(Indexed: ISI, Rank: Q2, Impact Factor: 3.772)

8 Ali, O., Ishak, M. K., & Bhatti, M. K. L. (2021). Adaptive clear channel assessment (A-CCA): energy efficient method to improve wireless sensor

networks (WSNs) operations. AEU-International Journal of Electronics and Communications, 131, 153603.

(Indexed: ISI, Rank: Q2, Impact Factor: 3.183)

### 2. Conference Proceedings

1 Ali, O., Ishak, M. K., & Bhatti, M. K. L. (2021). Internet of things security: Modelling smart industrial thermostat for threat vectors and common vulnerabilities. In Intelligent Manufacturing and Mechatronics (pp. 175-186). Springer, Singapore.

2 Ali, O., & Ishak, M. K. (2020, April). Bringing intelligence to IoT edge: Machine learning based smart city image classification using Microsoft azure IoT and custom vision. In Journal of Physics: Conference Series (Vol. 1529, No. 4, p. 042076). IOP Publishing.

### 3. Conference Articles

1 Omer Ali, Mohamad Khairi Ishak, Furqan Memon, and Mohd Shahrimie Mohd Asaari, "Estimation of Battery State-of-Charge using Feedforward Neural Networks", ECTICON 2022, Thailand

(Accepted)

2 Ali, O., Ishak, M. K., Zawawi, M. A. M., Seman, M. T. A., Bhatti, M. K. L., & Yusoff, Z. Y. M. (2020, June). A MAC protocol for energy efficient wireless communication leveraging wake-up estimations on sender data. In 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON) (pp. 45-50). IEEE.

3 Ali, O., Ishak, M. K., Wuttisittikulkij, L., & Maung, T. Z. B. (2020, January). IoT Devices and Edge gateway provisioning, realtime analytics for simulated and virtually emulated devices. In 2020 International Conference on Electronics, Information, and Communication (ICEIC) (pp. 1-5). IEEE.

### 4. Awards

1 Best Presenter award (*IoT Devices and Edge gateway provisioning, realtime analytics for simulated and virtually emulated devices*): 1st International Conference on Computing, Information Science and Engineering (ICISE) 2020, 29, 30 January 2020, Kota Bharu, Kelantan, Malaysia.

2 3MT Winner 2019 ($1^{st}$ position Engineering Campus – Universiti Sains Malaysia)

3 3MT Winner 2021 ($3^{rd}$ position Engineering Campus – Universiti Sains Malaysia)