Low-Power Wireless Sensor Node with Edge Computing for Pig Behavior Classifications

Yuezhong Xu

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Dong S. Ha, Chair

Sook S. Ha

Mantu K. Hudait

April 5, 2024

Blacksburg, Virginia

Keywords: Wireless Sensor Node (WSN), Edge Computing, Random Forest Classifier (RFC), Bluetooth Low Energy (BLE), Bio-sensing.

Low-Power Wireless Sensor Node with Edge Computing for Pig Behavior Classifications

Yuezhong Xu

(ABSTRACT)

A wireless sensor node (WSN) system, capable of sensing animal motion and transmitting motion data wirelessly, is an effective and efficient way to monitor pigs' activity. However, the raw sensor data sampling and transmission consumes lots of power such that WSNs' battery have to be frequently charged or replaced. The proposed work solves this issue through WSN edge computing solution, in which a Random Forest Classifier (RFC) is trained and implemented into WSNs. The implementation of RFC on WSNs does not save power, but the RFC predicts animal behavior such that WSNs can adaptively adjust the data sampling frequency to reduce power consumption. In addition, WSNs can transmit less data by sending RFC predictions instead of raw sensor data to save power. The proposed RFC classifies common animal activities: eating, drinking, laying, standing, and walking with a F-1 score of 93%. The WSN power consumption is reduced by 25% with edge computing intelligence, compare to WSN power that samples and transmits raw sensor data periodically at 10 Hz.

Low-Power Wireless Sensor Node with Edge Computing for Pig Behavior Classifications

Yuezhong Xu

(GENERAL AUDIENCE ABSTRACT)

A wireless sensor node (WSN) system that detects animal movement and wirelessly transmits this data is a valuable tool for monitoring pigs' activity. However, the process of sampling and transmitting raw sensor data consumes a significant amount of power, leading to frequent recharging or replacement of WSN batteries. To address this issue, our proposed solution integrates edge computing into WSNs, utilizing a Random Forest Classifier (RFC). The RFC is trained and deployed within the WSNs to predict animal behavior, allowing for adaptive adjustment of data sampling frequency to reduce power consumption. Additionally, by transmitting RFC predictions instead of raw sensor data, WSNs can conserve power by transmitting less data. Our RFC can accurately classify common animal activities, such as eating, drinking, laying, standing, and walking, achieving an F-1 score of 93%. With the integration of edge computing intelligence, WSN power consumption is reduced by 25% compared to traditional WSNs that periodically sample and transmit raw sensor data at 10 Hz.

Contents

List of Figures **vi**

List of Tables **viii**

1 Introduction **1**

1.1 Contributions 2

2 Preliminaries **4**

2.1 Traditional Computing vs. Edge Computing 4

2.2 Random Forest Classifier 5

3 Proposed WSN System with Edge Computing **7**

3.1 Proposed WSN System 7

3.2 Proposed Edge Computing through RFC 9

3.3 Implementation of Edge Computing 10

3.4 Power Dissipation after Implementation of Edge Computing 13

4 Power Saving Methods **15**

4.1 Adaptive Data Sampling 15

4.2 Reducd Data Transmission 18

4.3 Power Saving Methods Combined 20

5 Measurement Results **21**

5.1 Classification Performance 21

5.2 Power Analysis 22

6 Conclusion **25**

Bibliography **26**

List of Figures

2.1 General Architecture of Edge Computing 4

2.2 General Architecture of Traditional Computing 5

2.3 Architecture of Random Forest Classifier 6

3.1 WSN System Block Diagram 8

3.2 Custom PCB Design with External Flash Memory 8

3.3 Proposed Edge Computing Block Diagram 9

3.4 Summary of Pigs' Activity Labels 10

3.5 F-1 Score from Ten-fold Cross Verification vs. Window Size Chosen 11

3.6 Ten-fold Cross-Verification Results for each Individual Time Domain Features 12

3.7 Initial Implementation of Edge Computing 13

3.8 Single-Cycle Dissipation Comparison between with and without Implementation of Edge Computing 14

4.1 The Software Flow Chart for Implementation of Adaptive Data Sampling . . 17

4.2 Energy Consumed during Data Transmission vs. Size of Data Package Transmitted 18

4.3 The Software Flow Chart for Implementation of Reduced Data Transmission 19

4.4 Embedded Software Flow with Edge Computing 20

5.1 WSN Power Measurement Result with Sleep Mode, Data Processing, and Data Transmittion 22

5.2 WSN Energy Dissipation for One-Time Data Processing and Data Transmission 23

List of Tables

2.1 Caption 6

4.1 Pigs' Activity Duration Summary 16

5.1 Performance Summary of Proposed RFC through Ten-Fold Cross Verification 21

5.2 WSN Power Dissipation Summary under Different Operation Modes 22

5.3 Daily WSN Energy Dissipation Summary and Comparison 24

Chapter 1

Introduction

The surge in global animal product consumption, expected to rise by 70% by 2050, will worsen the issue of human hunger [5]. Factors like population growth and the spread of cultures promoting the intake of meat also play a role in aggravating the problem. To ensure the meat production efficiency, frequent visits to farms or manual reviews of recorded videos is required [9, 16, 21]. However, these approaches are laborious and inefficient that could negatively impact farms' profit. Additionally, smaller animal activities changes due to sickness may not be effectively caught by farmers' observation.

An effective approach to this issue is using WSNs with motion sensors, such as accelerometers and gyroscopes, enabling simultaneous monitoring of all animals to reduce labor costs [6, 10]. However, the substantial amount of energy needed for transmitting data wirelessly is the major challenge for WSNs [14]. For example, Cheung et al. introduced a low-power WSN system designed to gather motion data from pigs using accelerometers and gyroscopes [9]. The system transmits sensor data to a host machine via BLE, albeit at a power consumption approximately seven times higher than during normal operation [9]. In addition, Antoine-Santoni et al. introduced a wireless tracking and sensor system utilizing Long Range (LoRa) communication, where the current consumption during transmission is roughly nine times greater than that of other data sensing and processing [2]. More over, Hussain et al. introduced a WSN based on Raspberry Pi, capable of real-time sensing of animals' motion [13]. The system transmits both data and images via WiFi, with the WiFi's power consumption potentially reaching 5.3 W, in contrast to the typical power consumption of microcontrollers, which operates below 100 mW [12, 13, 28].

1.1 Contributions

Due to high power consumption during data transmission, it is crucial to optimize data transmission to further save the power dissipation of WSNs. In this paper, we applied edge computing to help reduce data transmissions, thus saving WSN power dissipation. The main contributions are summarized as follows:

- We redesigned the WSN from prior work referenced in [9], incorporating a larger flash memory capacity to enhance edge computing capabilities.
- We trained a compact but accurate RFC using the data gathered from the prior work referenced in [9]. Subsequently, we optimized the model's size and integrated it into the microcontroller, enabling the WSN to effectively predict pigs' activity.
- With the RFC running in the WSN, we programmed it to adaptively adjust data sensing frequency based on predictions of pigs' activity, thereby allowing the sensors to sleep for longer periods and saving power.
- We proposed a more efficient approach to data transmission to reduce power: sending pigs' activity predictions rather than complete sensor data and transmitting larger data packages to reduce the number of transmissions.

The rest of the thesis is organized as follows. Section 2 reviews the WSN system set-up and random forest classifier. Section 3 presents the proposed WSN design, the proposed edge computing through RFC, and implementation of edge computing. Section 4 presents the power optimization methods from edge computing. Section 5 presents the edge computing classification and power measurement results, and Section 6 draws the conclusion.

Chapter 2

Preliminaries

2.1 Traditional Computing vs. Edge Computing

Edge computing is a decentralized architecture for data processing, wherein data processing shifts from a host machine to WSNs [19, 30], as shown in fig. 2.1.

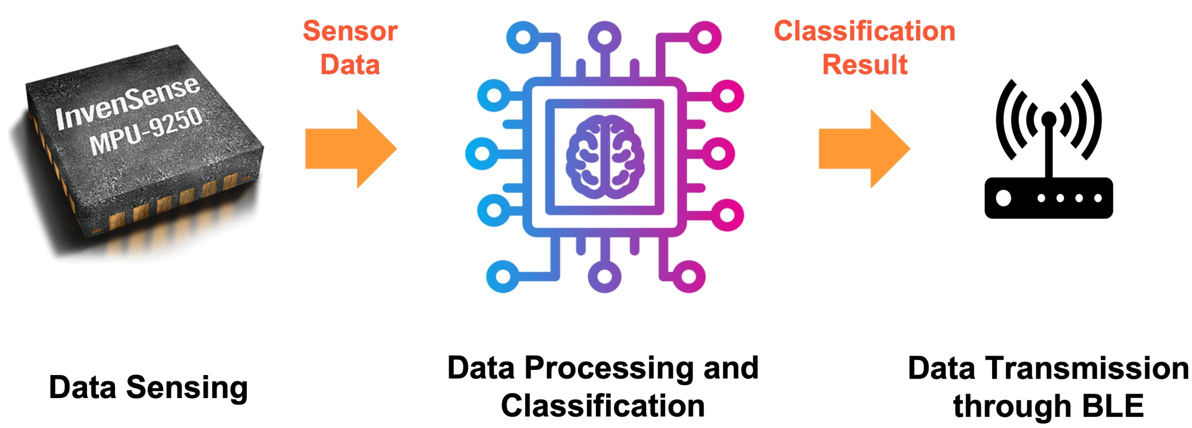

Figure 2.1: General Architecture of Edge Computing

Unlike traditional centralized computing models, as shown in fig. 2.2, which emphasize central computing power and data transmission bandwidth, edge computing redistributes computational tasks to WSNs, thereby reducing the data transmission needs between the host machine and WSNs, as well as the data processing requirements at the host machine [8, 11, 23]. Through the utilization of a distributed network comprising interconnected devices, edge computing maximizes resource usage, leading to improved real-time decision-making abilities, and minimized data transmission load [3, 17, 25].

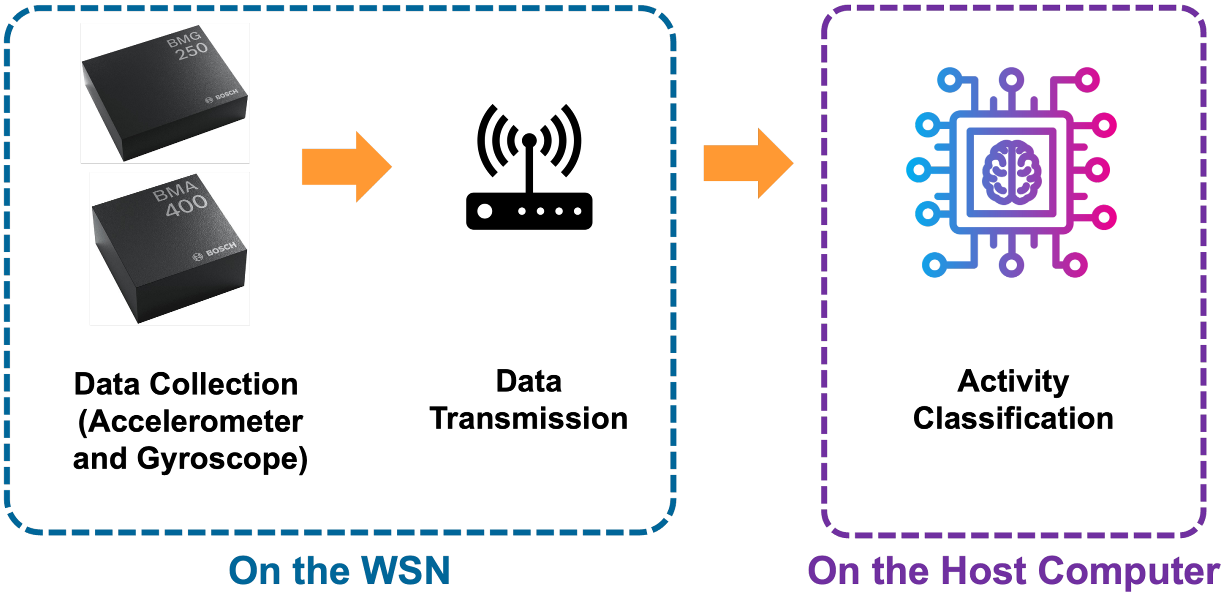

Figure 2.2: General Architecture of Traditional Computing

However, by redistributing computational tasks, edge computing significantly increases the processing power and space requirements on WSNs, potentially leading to programmability issues due to shortages in memory space and computing power [24]. In addition, ensuring the security of WSNs and the quality of results produced by them are also major challenges for applications of edge computing [7, 20, 29].

2.2 Random Forest Classifier

The RFC is a supervised machine learning algorithm for regression and classification, which operates by constructing multiple decision trees during training and combining their predictions through a voting mechanism, as shown in fig. 2.3 [15, 18]. Each decision tree in the RFC is trained independently on a random subset of the dataset and features selected in parallel, which reduces overfitting and enhances the capability of handling large datasets.

Additionally, RFC aggregates the predictions of multiple trees, making it less susceptible to the influence of individual noisy data points or outliers.

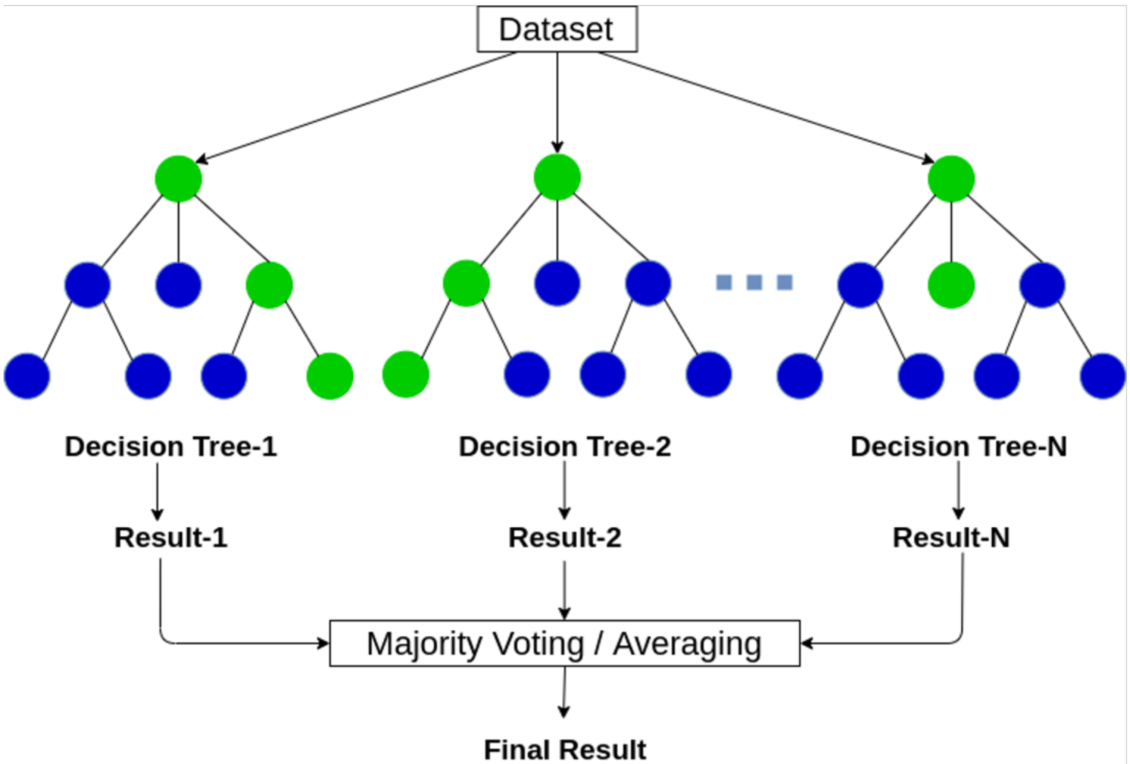

Figure 2.3: Architecture of Random Forest Classifier

RFC is popular in remote sensing due to its high accuracy in this application [1, 4, 22]. Additionally, RFC is implemented with conditional statements that save computing and memory resources on a WSN, resulting in reduced power consumption during data processing.

Table 2.1: Caption

Chapter 3

Proposed WSN System with Edge Computing

3.1 Proposed WSN System

The proposed WSN system collects 3-axial linear acceleration and angular velocity data from pigs, then process and transmit to the data to a nearby PC, as shown in fig. 3.1. The system consists an MPU9250 IMU module [26] to collect the pigs' motion data, and a TI-CC2640R2F micro-controller unit (MCU) [27] to process and transmit the data. The TI-CC2640R2F microcontroller consists a ARM Cortex M3 low-power computing unit and a BLE unit that satisfies the computation and data transmission needs [27].

To enhance the edge computing capabilities, a custom PCB, with a dimensions of 89.45 $\times$ 51.95 mm, is designed as shown in fig. 3.2. Besides the MCU and IMU, it also includes an 8-Mbit external flash memory, increasing the total flash memory size from 128 KB to 1152 KB. This increase in memory size enables edge computing capabilities on the system.

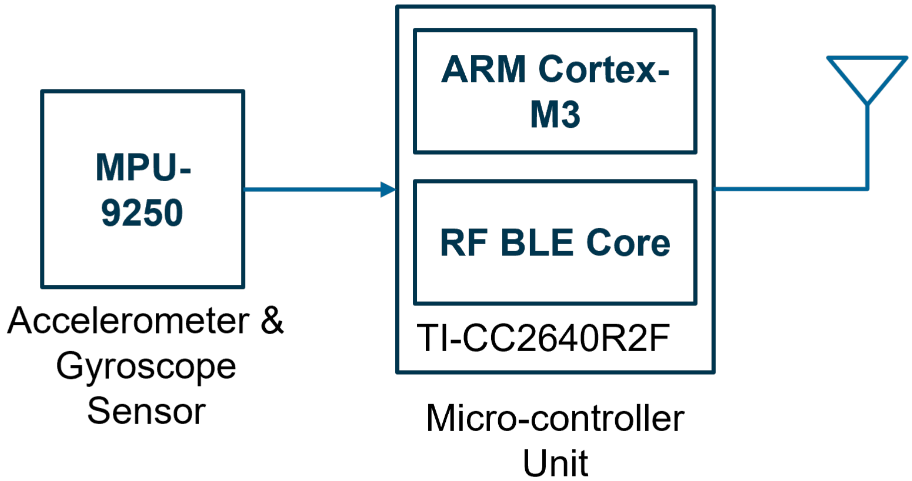

Figure 3.1: WSN System Block Diagram

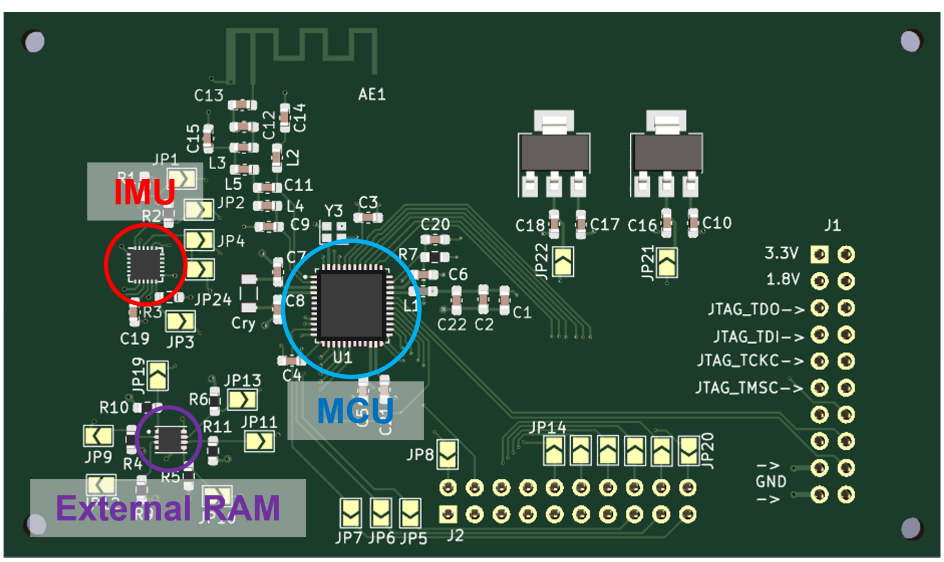

Figure 3.2: Custom PCB Design with External Flash Memory

3.2 Proposed Edge Computing through RFC

In addition to data sensing and transmitting from the MCU, a RFC is trained and implemented on the MCU to enable edge computing. Instead of collecting sensor data and transmitting it to a nearby host computer instantaneously, the sensor data is fed into the RFC to predict the current pigs' activity before transmitting the data, as shown in fig. 3.3.

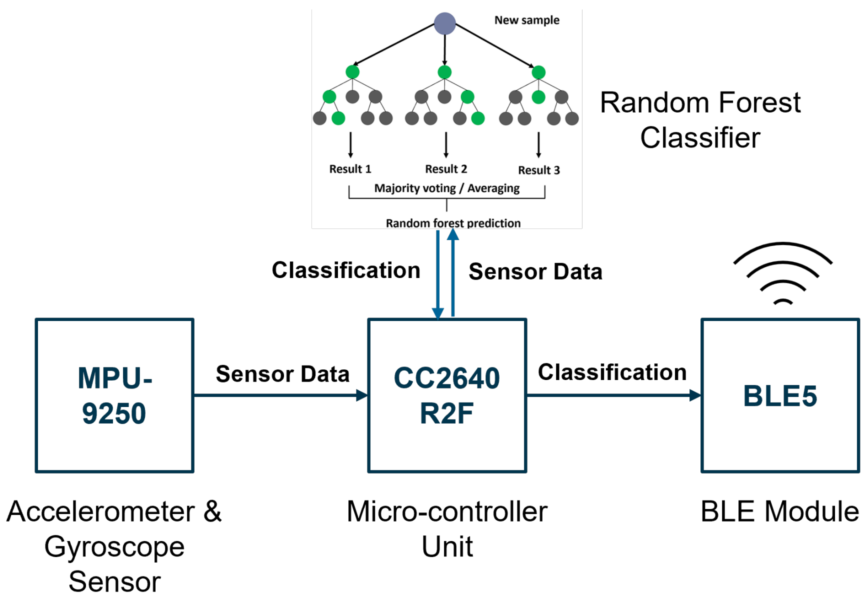

Figure 3.3: Proposed Edge Computing Block Diagram

The RFC is trained using the data collected from our prior work referenced in [9]. The 131-hour collected and manually labeled data show that the pigs spent 99% of their time on activities necessary for living, excluding unknown activities, as shown in fig. 3.4. Therefore, the RFC is trained to precisely predict the five most common activities we captured from the data collection: eating, laying, walking, standing, and drinking.

With the RFC implemented on the MCU, the WSN system becomes a smart system that recognizes pigs' activities without transmitting the raw data to the host machine. It can also adaptively adjust its transmission data rate and sensing frequency based on the activity

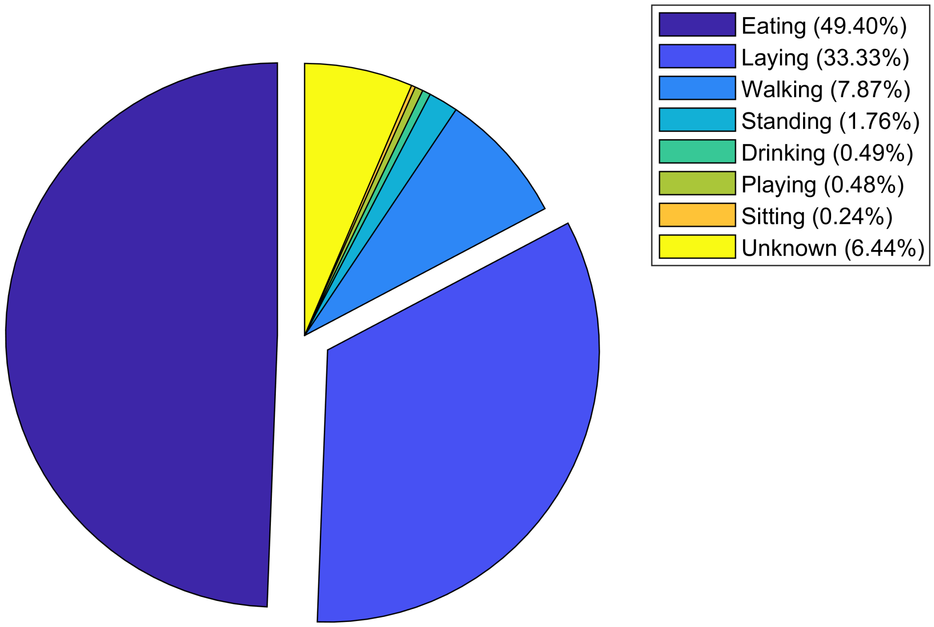

Figure 3.4: Summary of Pigs' Activity Labels

prediction, with the new embedded software flow shown in fig. 3.3.

3.3 Implementation of Edge Computing

It is essential to reduce data computation in edge computing to save data processing power and fit the RFC model into the limited memory space on the WSN. Our approach to reducing power dissipation and memory space occupancy is to find efficient window sizes and use the most effective features for edge computing.

Optimizing the window size for RFC data processing in edge computing is important. Therefore, we performed ten-fold cross-validation with window sizes ranging from 2 to 15, and the F-1 scores are shown in fig. 3.5. The cross-validation results show that a higher window size leads to more accurate activity prediction. However, increasing the window size will significantly increase MCU computation load, resulting in higher power dissipation and memory occupancy. We chose a window size of 6 to implement edge computing because this perfectly balances the RFC's accuracy and computation load.

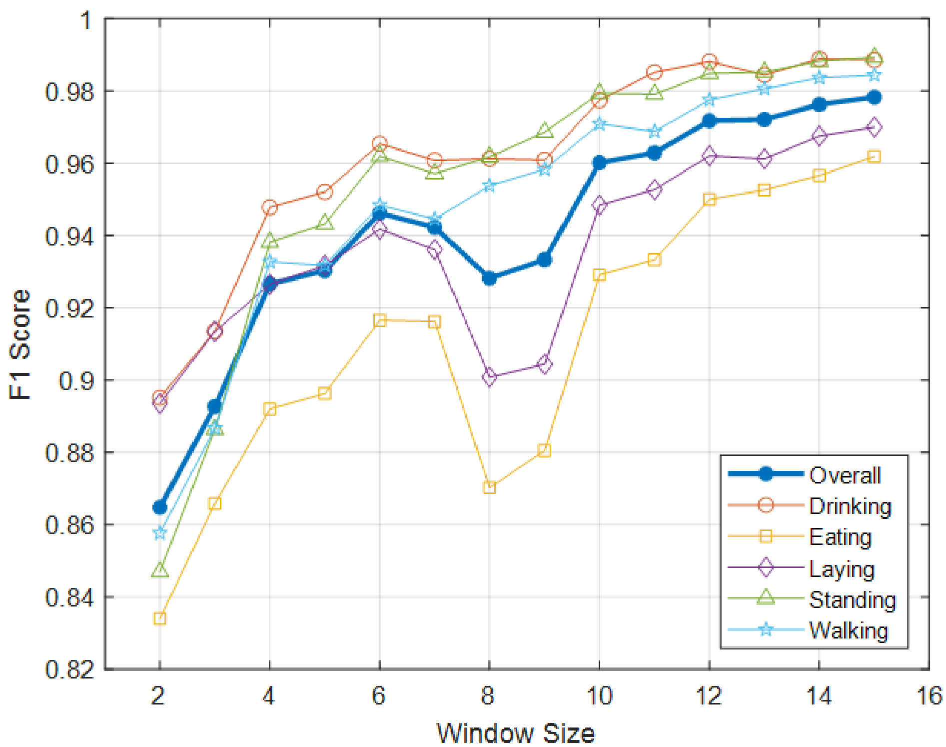

Figure 3.5: F-1 Score from Ten-fold Cross Verification vs. Window Size Chosen

Implementing every conceivable time-domain feature extraction method in the RFC for edge computing is impractical due to limited computational power and memory space. Therefore, selecting the most effective features is crucial for reducing the MCU's computation and memory space occupancy, resulting in power savings. We evaluated the effectiveness of each time domain feature individually through ten-fold cross-validation, with the F-1 scores for each feature shown in Fig. 3.6. Both the 'average' and 'median' features achieved an F-1 score of over 0.9, and combining them resulted in an F-1 score of 0.93. Additional features led to only a marginal increase in F-1 score but a significantly higher computational load. Consequently, we opted to retain only these two features to strike a balance between RFC accuracy and computation load.

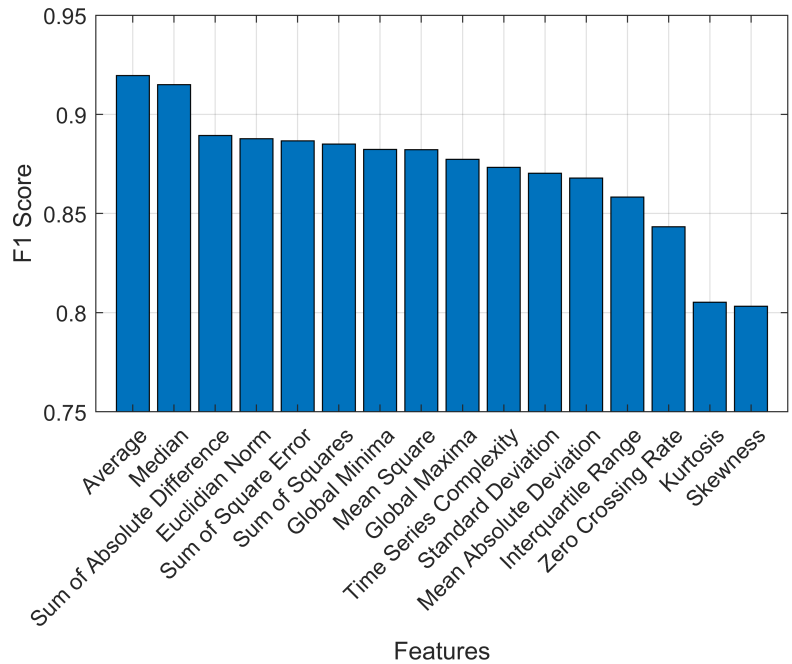

Figure 3.6: Ten-fold Cross-Verification Results for each Individual Time Domain Features

The initial implementation of edge computing keeps the same data sensing and transmission frequency, but an RFC is running on the WSN such that activity prediction is transmitted to the receiver, as shown in Fig. 3.7. This implementation increases the workload for data processing on the microcontroller but reduces the workload for data transmission. Since the microcontroller consumes more power during data transmission than data processing, a power reduction is expected.

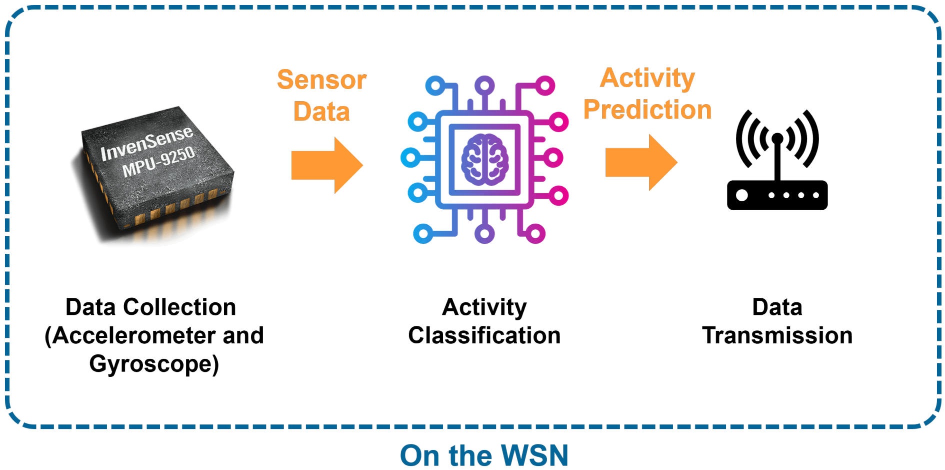

Figure 3.7: Initial Implementation of Edge Computing

3.4 Power Dissipation after Implementation of Edge Computing

Fig. 3.8 shows the comparison of power dissipation of the wireless sensor node under sleep mode, data processing mode, and data transmission mode after the initial implementation of the edge computing. The power dissipation during sleep mode is similar between with and without edge computing due to the operating conditions being the same. However, the power dissipation for data processing and transmission remains similar between with and without edge computing, telling us that the computing and transmission power is not really sensitive to the workload. With an edge computing implemented, it allows us to make the wireless sensor node running more power efficient with methods in Chapter 4.

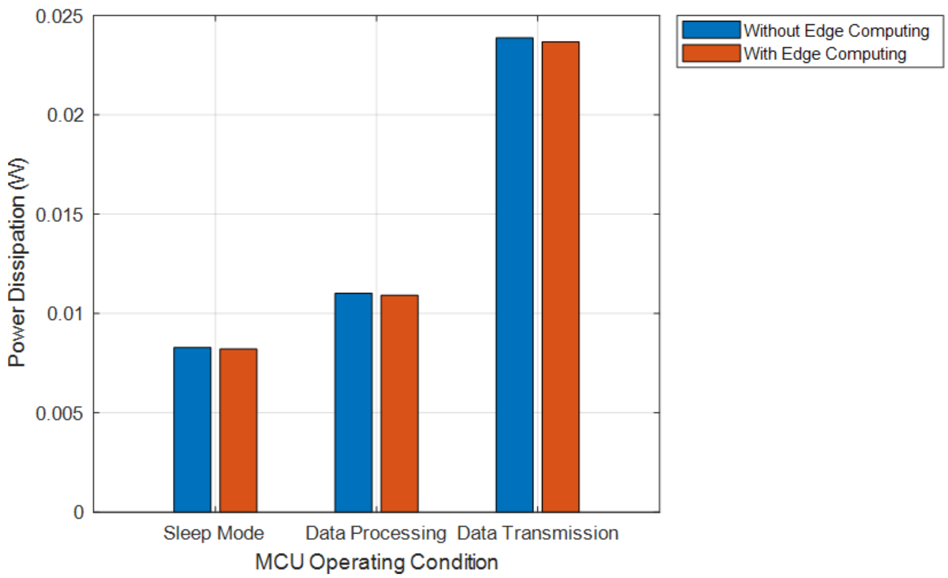

Figure 3.8: Single-Cycle Dissipation Comparison between with and without Implementation of Edge Computing

Chapter 4

Power Saving Methods

WSN consumes a lot of energy during the data processing and transmission. The typical power dissipation is 23.9 mW for data transmission and 11 mW for data processing, which is significantly higher than the typical sleep power dissipation of 8.3 mW, as shown in fig. 3.8. After adding edge computing to WSNs that predict and transmit a 4-bit activity code every 0.1 second, the WSNs' power dissipation remains similar, which still results in wasted power. However, the implemented edge computing enables us to reduce data sampling and transmission, ultimately saving power.

4.1 Adaptive Data Sampling

A lower data sampling rate will help save power, but it also reduces the resolution of data in normal operation, leading to misrecognizing pigs' activity. However, a reduced sampling rate model may apply when edge computing is implemented, because edge computing enables the WSNs to understand pigs' activity. To reduce the power dissipation from data sampling, we reduced the data sampling rate by extending the WSN's time in sleep mode. The time in sleep mode is determined through pigs' activity duration analysis using data collected and manually labeled in [9], with the results shown in table 4.1.

Pigs' activity duration varies due to a number of factors, such as the time of day and other pigs' activities. For example, pigs can lay down for as short as 0.8 seconds when other

Table 4.1: Pigs' Activity Duration Summary

	Duration of Activity (s)				
Activity	Min	25th Quartile	Median	75th Quartile	Max
Drinking	0.5	3.8	6.1	9.2	59.8
Eating	0.4	8.6	19.2	43.8	787.6
Laying	0.8	13.1	36.9	151.1	7009.1
Standing	0.3	2.2	3.9	7.4	104.2
Walking	0.7	3.0	4.6	7.5	139.2

pigs are active, and they can also keep laying down for about 2 hours in the evening. To achieve a better balance between accurate activity recognition and low power dissipation, we decided to put the WSN to sleep for {6.1, 19.2, 36.9, 3.9, 4.6} seconds after the WSN predicts {drinking, eating, laying, standing, walking} activities, with the flow chart shown in Fig. 4.1. Because the WSN sleep mode period is significantly longer than 0.1 second with this approach, it leads to a significantly lower data sampling rate and significantly lower power dissipation from data sampling.

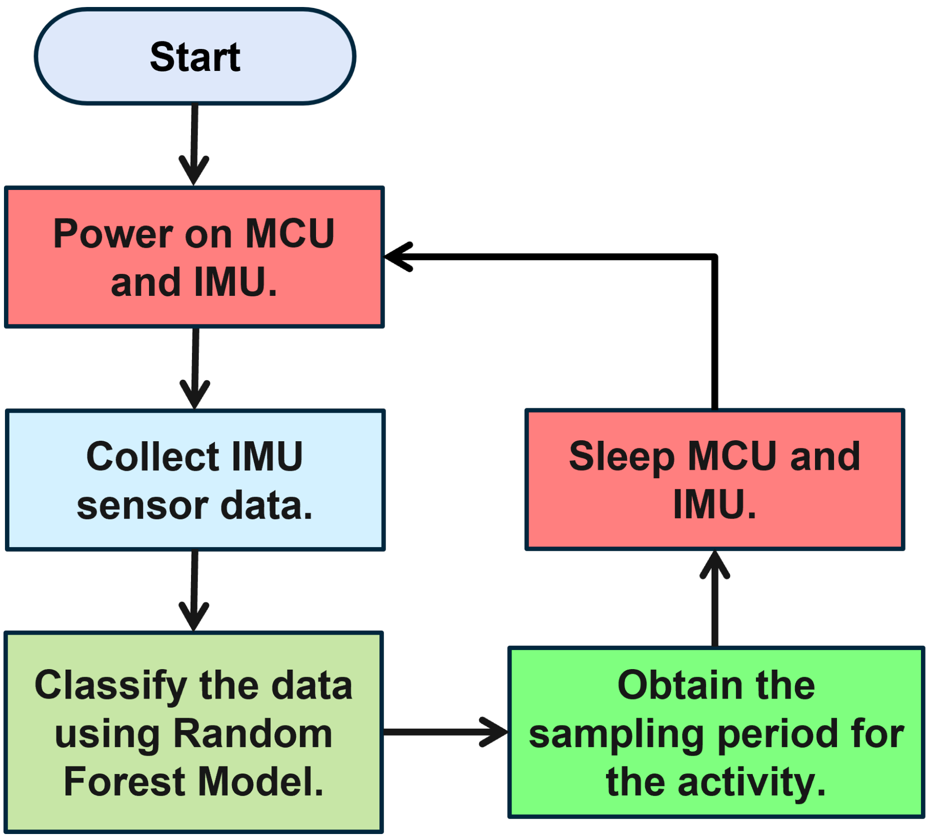

Figure 4.1: The Software Flow Chart for Implementation of Adaptive Data Sampling

4.2 Reducd Data Transmission

Data transmission consumes the most power while the WSN is operating, so reducing data transmission will significantly reduce power dissipation. Each data transmission cycle involves activating the BLE, sampling starting bits before transmitting data, and sampling end bits before deactivating the BLE. The power dissipation associated with these operations remains constant. Thus, transmitting larger data packages with fewer transmissions is more efficient. To demonstrate this, we transmitted random data packages with sizes of {64, 128, 256, 512} bits and measured the energy consumption. We found that the energy consumed for data transmission was similar regardless of the data package size, as shown in Fig. 4.2.

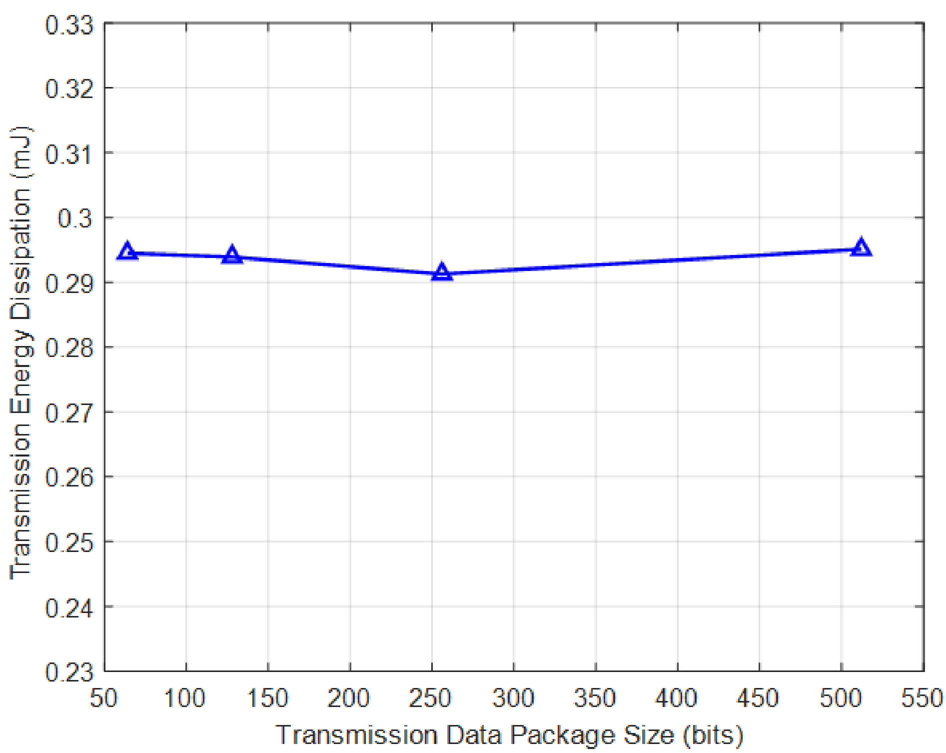

Figure 4.2: Energy Consumed during Data Transmission vs. Size of Data Package Transmitted

As real-time data sensing and transmission become less critical after the implementation of edge computing, we developed an embedded program to store the activity predictions made by the edge computing system, as the software flow chart shown in Fig. 4.3. Data transmission occurs when the data payload reaches its maximum capacity. The maximum data capacity for each transmission is 704 bits, equivalent to 176 prediction results. This approach reduces transmission energy consumption by 176 times compared to real-time data transmission.

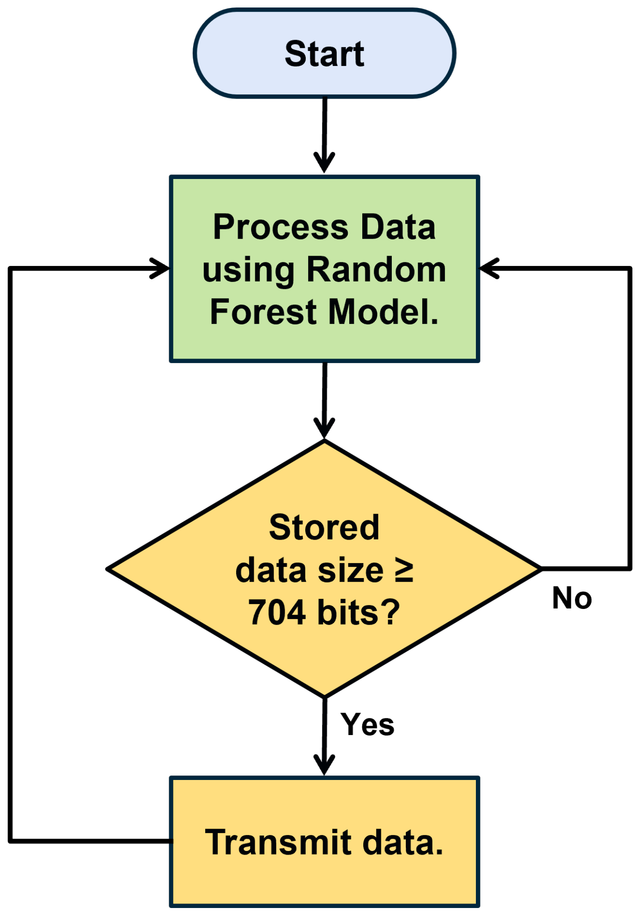

Figure 4.3: The Software Flow Chart for Implementation of Reduced Data Transmission

4.3 Power Saving Methods Combined

By combining the power saving methods mentioned in sections 4.1 and 4.2, the general software flow in the wireless sensor node is shown in Fig. 4.4. The wireless sensor node starts by collecting the sensor data and then processing it through the RFC. Once the classification result is obtained, the wireless sensor node starts by deciding whether to store or transmit data based on the size of the total stored data, then decides the time to sleep itself and wait for the next data collection.

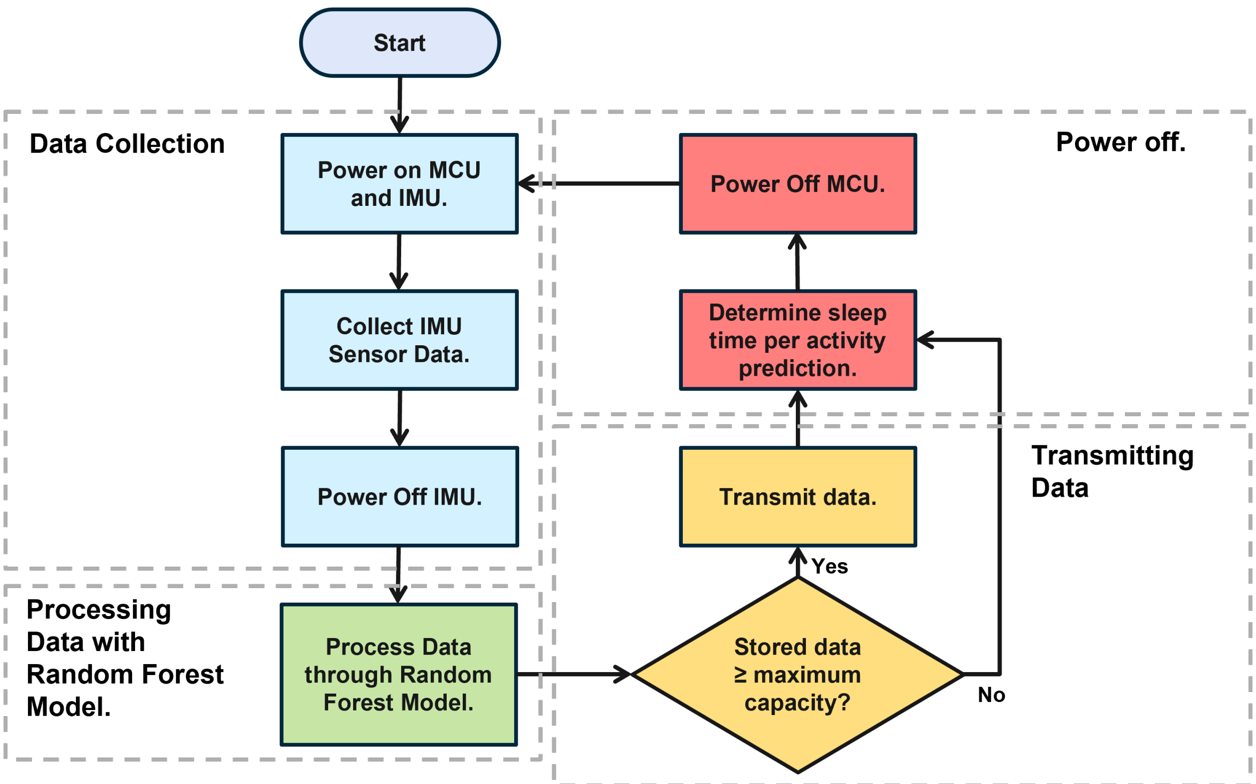

Figure 4.4: Embedded Software Flow with Edge Computing

Chapter 5

Measurement Results

5.1 Classification Performance

The classification performance of the RFC was evaluated using ten-fold cross-validation, with the recall, precision, overall confusion matrix, and F-1 score presented in Table 5.1. The accelerometer sensor readings for 'eating' and 'standing' are similar, leading to some confusion between these activities. Our RFC achieved an overall F-1 score of 93%, demonstrating high effectiveness in predicting 'drinking' and 'laying' activities with high recall and precision, but relatively lower effectiveness in predicting 'eating' activity.

Table 5.1: Performance Summary of Proposed RFC through Ten-Fold Cross Verification

Method	RFC					
Activity	Drinking	Eating	Laying	Standing	Walking	*Recall (%)*
Drinking	**1978**	9	7	0	6	*98.9*
Eating	75	**2163**	61	104	97	*86.5*
Laying	55	82	**2311**	27	25	*92.4*
Standing	12	56	36	**2376**	20	*95.0*
Walking	24	69	13	25	**1869**	*93.5*
Precision (%)	*92.3*	*90.9*	*95.2*	*93.8*	*92.7*	***93.0***

5.2 Power Analysis

The power dissipation of the WSN system was measured using an Analog Discovery 2 scope with a shunt resistor, and the resulting waveform is illustrated in Fig. 5.1. The average power dissipation under different operation modes is detailed in Table 5.2. The WSN's maximum power consumption, reaching 27.9 mW, occurs during transmission, with all other operation modes consuming less than 15 mW of power.

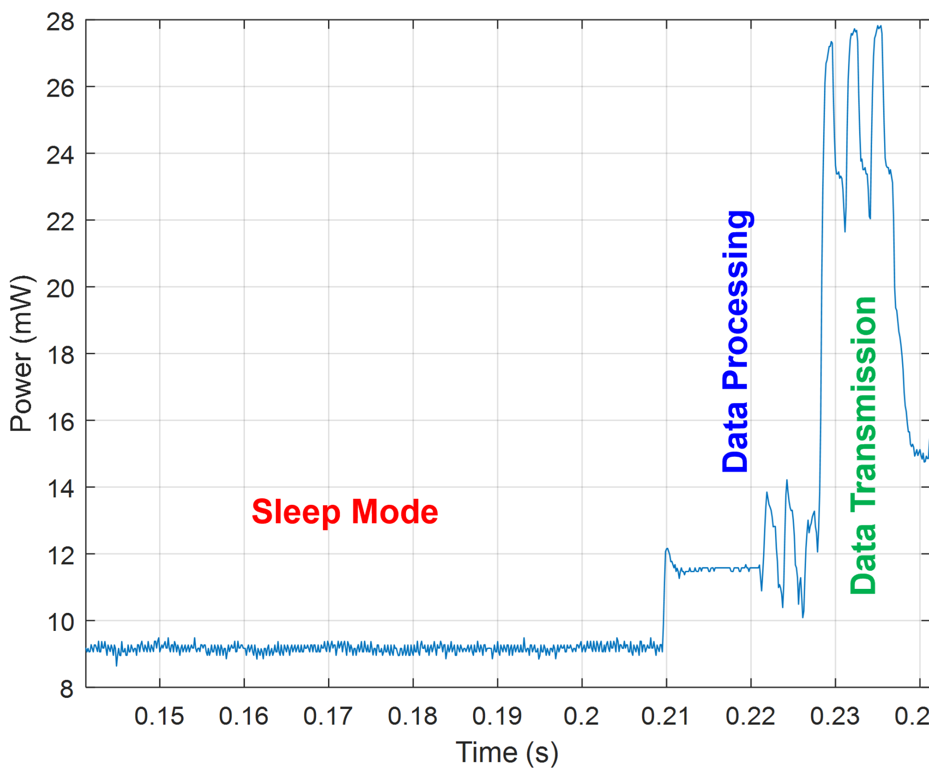

Figure 5.1: WSN Power Measurement Result with Sleep Mode, Data Processing, and Data Transmittion

Table 5.2: WSN Power Dissipation Summary under Different Operation Modes

WSN Operation Mode	Supply Voltage (V)	Avg. Current Dissipation (mA)	Avg. Power Dissipation (mW)
Sleep Mode	3.3	2.52	8.3
Data Processing	3.3	3.33	11.0
Data Transmission	3.3	7.24	23.9

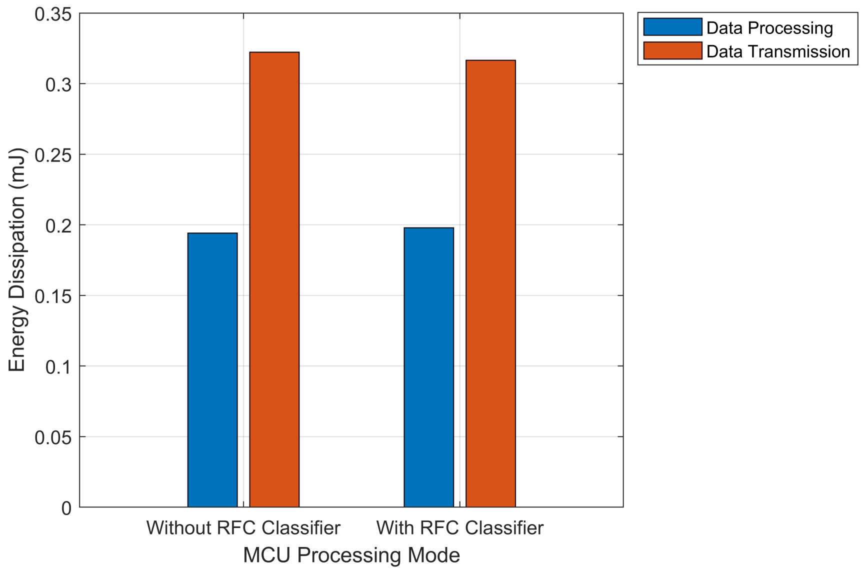

Figure 5.2: WSN Energy Dissipation for One-Time Data Processing and Data Transmission

Table 5.3 presents a summary of the daily energy dissipation in the Wireless Sensor Network (WSN), comparing scenarios with and without the application of edge computing. The calculation of daily energy dissipation takes into account the summary of pigs' activity, as illustrated in Fig. 3.4, the variable sampling rate detailed in Table 4.1, and the power dissipation results depicted in Fig. 5.1.

With the implementation of edge computing in the WSN, the daily energy consumed during data processing decreases from 46.5 mWh to 0.293 mWh, and from 77.3 mWh to 2.7 μWh for data transmission. However, due to the WSN operating in sleep mode for an additional 7.4 hours after the implementation of edge computing, there is a larger daily energy consumption in sleep mode. Nevertheless, the overall daily energy dissipation decreases from 261 mWh to 197 mWh, resulting in a 25% power savings every day.

Table 5.3: Daily WSN Energy Dissipation Summary and Comparison

Activity	Daily Energy Dissipation (mWh)							
	With Edge Computing				Without Edge Computing			
	Sleep Mode	Processing	Transmission	**Total**	Sleep Mode	Processing	Transmission	**Total**
Eating	105	0.131	1.2e-3	105	72.9	24.8	41.2	139
Laying	70.7	0.046	4.2e-4	70.8	49.2	16.7	27.7	93.7
Walking	16.6	0.087	8.0e-4	16.7	11.6	3.94	6.56	22.1
Standing	3.72	0.023	2.1e-4	3.74	2.60	0.88	1.47	4.95
Drinking	1.04	4.1e-3	3.7e-5	1.04	0.72	0.25	0.41	1.38
Total	*197*	*0.293*	*2.7e-3*	**197**	*137*	*46.5*	*77.3*	**261**

Chapter 6

Conclusion

This paper presents a method to reduce power consumption in WSN systems, achieving a 25% reduction compared to the original operation. The proposed edge computing, powered by RFC, predicts pigs' activity in the WSN and adaptively adjusts the data sampling and transmission rates to reduce power consumption. We optimized the RFC by selecting the most efficient window size and effective features, verifying the approach through a 10-fold cross-validation for accurate results, aiming for minimal computing power dissipation and memory occupancy. The application of the proposed edge computing reduces the overall daily energy dissipation from 261 mWh to 197 mWh, resulting in a 25% power saving each day.

Bibliography

[1] Saleh Alghamdi, Zhuqing Zhao, Dong S Ha, Gota Morota, and Sook S Ha. Improved pig behavior analysis by optimizing window sizes for individual behaviors on acceleration and angular velocity data. *Journal of Animal Science*, 100(11):skac293, 09 2022. ISSN 1525-3163. doi: 10.1093/jas/skac293. URL https://doi.org/10.1093/jas/skac293.

[2] Thierry Antoine-Santoni, Jean-Sebastien Gualtieri, François-Marie Manicacci, and Antoine Aiello. AMBLoRa: a Wireless Tracking and Sensor System Using Long Range Communication to Monitor Animal Behavior,. In *the Seventh International Conference on Smart Cities, Systems, Devices and Technologies*, Proceedings of the Seventh International Conference on Smart Cities, Systems, Devices and Technologies, pages 35–40, Barcelone, Spain, July 2018. IARIA, IARIA. URL https://hal.science/hal-03028480.

[3] Ahmet Cihat Baktir, Atay Ozgovde, and Cem Ersoy. How can edge computing benefit from software-defined networking: A survey, use cases, and future directions. *IEEE Communications Surveys Tutorials*, 19(4):2359–2391, 2017. doi: 10.1109/COMST.2017.2717482.

[4] Mariana Belgiu and Lucian Drăguţ. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:24–31, 2016. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2016.01.011. URL https://www.sciencedirect.com/science/article/pii/S0924271616000265.

[5] D. Berckmans. General introduction to precision livestock farming. *Animal Frontiers*,

7(1):6–11, 01 2017. ISSN 2160-6056. doi: 10.2527/af.2017.0102. URL https://doi.org/10.2527/af.2017.0102.

[6] M.R. Borchers, Y.M. Chang, I.C. Tsai, B.A. Wadsworth, and J.M. Bewley. A validation of technologies monitoring dairy cow feeding, ruminating, and lying behaviors. *Journal of Dairy Science*, 99(9):7458–7466, 2016. ISSN 0022-0302. doi: https://doi.org/10.3168/jds.2015-10843. URL https://www.sciencedirect.com/science/article/pii/S0022030216304490.

[7] Jie Cao, Quan Zhang, and Weisong Shi. *Challenges and Opportunities in Edge Computing*, pages 59–70. Springer International Publishing, Cham, 2018. ISBN 978-3-030-02083-5. doi: 10.1007/978-3-030-02083-5_5. URL https://doi.org/10.1007/978-3-030-02083-5_5.

[8] Keyan Cao, Yefan Liu, Gongjie Meng, and Qimeng Sun. An overview on edge computing research. *IEEE Access*, 8:85714–85728, 2020. doi: 10.1109/ACCESS.2020.2991734.

[9] Brandon Cheung, Yuezhong Xu, and Dong Sam Ha. Wireless sensor node system to monitor pig activities for behavior classification. In *2023 IEEE 66th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 571–575, 2023. doi: 10.1109/MWSCAS57524.2023.10406003.

[10] Anderson da Silva Santos, Victor Wanderley Costa de Medeiros, and Glauco Estácio Gonçalves. Monitoring and classification of cattle behavior: a survey. *Smart Agricultural Technology*, 3:100091, 2023. ISSN 2772-3755. doi: https://doi.org/10.1016/j.atech.2022.100091. URL https://www.sciencedirect.com/science/article/pii/S2772375522000569.

[11] Abhishek Priyadarshan Damle, Sook Shin Ha, Zhuqing Zhao, Barbara Roqueto Dos Reis, Robin White, and Dong Sam Ha. Power efficient wireless sensor node through

edge intelligence. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1749–1753, 2022. doi: 10.1109/ISCAS48785.2022.9937324.

[12] Karina Gomez, Roberto Riggio, Tinku Rasheed, and Fabrizio Granelli. Analysing the energy consumption behaviour of wifi networks. In *2011 IEEE Online Conference on Green Communications*, pages 98–104, 2011. doi: 10.1109/GreenCom.2011.6082515.

[13] Shaik Javeed Hussain, Samiullah Khan, Raza Hasan, and Shaik Asif Hussain. Design and implementation of animal activity monitoring system using ti sensor tag. In Pradeep Kumar Mallick, Valentina Emilia Balas, Akash Kumar Bhoi, and Gyoo-Soo Chae, editors, *Cognitive Informatics and Soft Computing*, pages 167–175, Singapore, 2020. Springer Singapore. ISBN 978-981-15-1451-7.

[14] Mohamed Rawidean Mohd Kassim. Iot applications in smart agriculture: Issues and challenges. In *2020 IEEE Conference on Open Systems (ICOS)*, pages 19–24, 2020. doi: 10.1109/ICOS50156.2020.9293672.

[15] Vrushali Y Kulkarni and Pradeep K Sinha. Pruning of random forest classifiers: A survey and future directions. In *2012 International Conference on Data Science Engineering (ICDSE)*, pages 64–68, 2012. doi: 10.1109/ICDSE.2012.6282329.

[16] M. L. V. Larsen, H. M.-L. Andersen, and L. J. Pedersen. Changes in activity and object manipulation before tail damage in finisher pigs as an early detector of tail biting. *animal*, 13(5):1037–1044, 2019. doi: 10.1017/S1751731118002689.

[17] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B. Letaief. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys Tutorials*, 19(4):2322–2358, 2017. doi: 10.1109/COMST.2017.2745201.

[18] Aakash Parmar, Rakesh Katariya, and Vatsal Patel. A review on random forest: An

ensemble classifier. In Jude Hemanth, Xavier Fernando, Pavel Lafata, and Zubair Baig, editors, *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018*, pages 758–763, Cham, 2019. Springer International Publishing. ISBN 978-3-030-03146-6.

[19] Gopika Premsankar, Mario Di Francesco, and Tarik Taleb. Edge computing for the internet of things: A case study. *IEEE Internet of Things Journal*, 5(2):1275–1284, 2018. doi: 10.1109/JIOT.2018.2805263.

[20] Tie Qiu, Jiancheng Chi, Xiaobo Zhou, Zhaolong Ning, Mohammed Atiquzzaman, and Dapeng Oliver Wu. Edge computing in industrial internet of things: Architecture, advances and challenges. *IEEE Communications Surveys Tutorials*, 22(4):2462–2488, 2020. doi: 10.1109/COMST.2020.3009103.

[21] Domingo S. Rodriguez-Baena, Francisco A. Gomez-Vela, Miguel García-Torres, Federico Divina, Carlos D. Barranco, Norberto Daz-Diaz, Manuel Jimenez, and Gema Montalvo. Identifying livestock behavior patterns based on accelerometer dataset. *Journal of Computational Science*, 41:101076, 2020. ISSN 1877-7503. doi: https://doi.org/10.1016/j.jocs.2020.101076. URL https://www.sciencedirect.com/science/article/pii/S1877750318311359.

[22] V.F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J.P. Rigol-Sanchez. An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67:93–104, 2012. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2011.11.002. URL https://www.sciencedirect.com/science/article/pii/S0924271611001304.

[23] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017. doi: 10.1109/MC.2017.9.

[24] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016. doi: 10.1109/JIOT.2016.2579198.

[25] Raghubir Singh, Simon Armour, Aftab Khan, Mahesh Sooriyabandara, and George Oikonomou. The advantage of computation offloading in multi-access edge computing. In *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 289–294, 2019. doi: 10.1109/FMEC.2019.8795335.

[26] TDKInvenSense. Tdk-invensense motion sensor universal evaluation board (uevb) user guide. August 2017. URL https://invensense.tdk.com/wp-content/uploads/2017/06/AN-IVS-0001EVB-00-v1.4.pdf.

[27] Texas Instrument. CC2640 Simplelink Bluetooth Wireless MCU Rev. b (SWRS176B). July 2016. URL https://www.ti.com/lit/ds/symlink/cc2640.pdf.

[28] Ioanna Tsekoura, Gregor Rebel, Peter Glösekötter, and Mladen Berekovic. An evaluation of energy efficient microcontrollers. In *2014 9th International Symposium on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC)*, pages 1–5, 2014. doi: 10.1109/ReCoSoC.2014.6861368.

[29] Blesson Varghese, Nan Wang, Sakil Barbhuiya, Peter Kilpatrick, and Dimitrios S. Nikolopoulos. Challenges and opportunities in edge computing. In *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, pages 20–26, 2016. doi: 10.1109/SmartCloud.2016.18.

[30] Wei Yu, Fan Liang, Xiaofei He, William Grant Hatcher, Chao Lu, Jie Lin, and Xinyu Yang. A survey on the edge computing for the internet of things. *IEEE Access*, 6: 6900–6919, 2018. doi: 10.1109/ACCESS.2017.2778504.