



Optimizing Edge Computing and IoT for Affordable and Portable Vibration-Based Machinery Condition Monitoring Solutions in Sri Lankan SMEs

R. P. Samaraweera

Reg. No.:MS23059986

A THESIS

SUBMITTED TO

SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

SPECIALIZATION IN ENTERPRISE APPLICATION DEVELOPMENT

December 2024

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

.

Ms. Shashika Lokuliyana

Approved for MSc. Research Project:

MSc. Programme Co-ordinator, SLIIT

Approved for MSc:

Head of Graduate Studies, FoC, SLIIT

DECLARATION

This is to certify that the work is entirely my own and not of any other person, unless explicitly acknowledged (including citation of published and unpublished sources). The work has not previously been submitted in any form to the Sri Lanka Institute of Information Technology or to any other institution for assessment for any other purpose.



Sign:

R. P. Samaraweera

Date: 28/ 01/ 2025

ABSTRACT

This research focuses on developing an affordable and portable system for monitoring the condition of machinery in Sri Lankan SMEs, utilizing Edge Computing and IoT technologies. The study is conducted in three stages. First, vibration data is collected from sensors attached to gearboxes to monitor for anomalies. In the second stage, the collected signals are processed using wavelet transform to extract relevant features from the data. Finally, machine learning classifiers are employed to identify anomalies, with a comparison of models including Convolutional Neural Networks (CNN), Random Forest (RF), and Autoencoders (AE). The goal was to create an effective solution for early detection of machinery issues, reducing unexpected maintenance costs, and improving operational efficiency in SMEs. This research aims to support SMEs in Sri Lanka by offering a cost-effective method to prevent machinery failures and enhance business modernization. Using IoT, signal processing, and machine learning models combination for gearbox fault detection along with the Python GUI interface called Gearbox Monitoring System (GMS) significantly improves predictive maintenance in the industrial sector. Provide reliable anomaly detection by implementing the RF model in the application, which can help prevent costly downtimes and improve the longevity of machinery.

Ranga Prabhashee Samaraweera

MSc. in Information Technology Specialization in Enterprise Application Development

Supervisor: Ms. Shashika Lokuliyana

December 2024

ACKNOWLEDGMENT

“I would like to thank Ms. Shashika Lokuliyana for her guidance and support throughout my research. Her advice was crucial in helping me plan and prepare my work. I also appreciate all the help and support from my other lecturers and academic staff.”

TABLE OF CONTENT

DECLARATION	ii
ABSTRACT	iii
ACKNOWLEDGMENT	iv
TABLE OF CONTENT	v
LIST OF FIGURES.....	vii
LIST OF TABLES	viii
Chapter 1 - Introduction	1
1.1 Background	1
1.2 Research Problem.....	3
1.3 Research Questions	4
1.4 Objectives.....	4
1.5 Significance of the Study	5
Chapter 2 - Literature Review	6
Chapter 3 - Research Methodology	10
3.1 Research Design.....	10
3.2 Conceptual Framework	12
3.3 Systematic Machinery Condition Identification Design	14
3.3.1 Stage 1: Data Collection.....	14
3.3.2 Stage 2: Signal Processing	14
3.3.3 Stage 3: Anomaly Detection.....	15
3.4 Data Collection Methods.....	17
3.5 Data Analysis Methods.....	18
3.6 Ethical Guidelines	19
3.7 Hardware and Software Requirements	19
3.7.1 Hardware Requirements	19
3.7.2 Software Requirements	21
Chapter 4 - Implementation.....	25
4.1 Stage 1: Data Collection.....	27
4.1.1 Gearbox Prototype.....	27
4.1.2 Vibration Sensors Unit	29

4.1.3	Arduino Control	32
4.1.4	Data Communication through MQTT Broker.....	34
4.2	Stage 2: Data Preparation	38
4.2.1	Visualize Real-Time Data on Node-RED Dashboard	38
4.2.2	Saving Data to CSV files.....	41
4.2.3	Pre-process Data.....	42
4.2.4	Wavelet Transform for Signal Processing and Feature Extraction.....	43
4.3	Stage 3: Anomaly Detection.....	46
4.3.1	Anomaly Detection Using ML Classifiers	46
4.3.2	Practical Solution for the System Using CNN and RF	56
4.3.3	Display Machine Condition on the System GUI.....	58
4.3.4	Real-Time Data Storage	65
4.4	Test Cases of the GMS System	66
4.4.1	When Data Collecting	66
4.4.2	When Identify Abnormality Condition	68
Chapter 5 -	Discussion	71
Chapter 6 -	Conclusion and Future Works.....	74
6.1	Conclusion.....	74
6.2	Future Works.....	75
References		77
Appendix		81
Appendix 1 -	Steps of the Experiment for Practical Usage.....	81

LIST OF FIGURES

Figure 3.1 - Conceptual Design of The System	12
Figure 3.2 - Steps of Systematic Machinery Condition Identification Process.....	15
Figure 3.3 - Worm Gearbox Figure 3.4 - Worm Gearbox	21
Figure 4.1 - Implementation Diagram.....	26
Figure 4.2 - Gearbox Prototype.....	28
Figure 4.3 - Vibration Sensor Unit	30
Figure 4.4 - Vibration Sensor Unit Structure	31
Figure 4.5 - Arduino IDE Coding	32
Figure 4.6 - Starting and Stopping the EMQX Server	34
Figure 4.7 - Checking Status of the EMQX Server.....	34
Figure 4.8 - EMXQ Dashboard.....	35
Figure 4.9 - Connected Client to the EMQX Server	35
Figure 4.10 - Topics Handle Across the EMQX Server	36
Figure 4.11 - Node-RED Flow Design.....	39
Figure 4.12 - Accelerometer Orientation Detection Readings	39
Figure 4.13 - Acceleration Data Reading	40
Figure 4.14 - Saved Acceleration Based Data Snapshot	41
Figure 4.15 - Check for Null Values	42
Figure 4.16 - Sample Signal for the Selected Window - x-axis	44
Figure 4.17 - Decomposed Components of the Sample Signal - x-axis.....	44
Figure 4.18 - Evaluation Matrix (Classification Report) – Random Forest Model	46
Figure 4.19 - Confusion Matrix - Random Forest Model	47
Figure 4.20 - Evaluation Matrix (Classification Report) - CNN Model	50
Figure 4.21 - Confusion Matrix - CNN Model	50
Figure 4.22 - Evaluation Matrix (Classification Report) – Autoencoders Model	52
Figure 4.23 - Confusion Matrix – Autoencoders Model	53
Figure 4.24 - Confusion Matrix comparison of RN, CNN, AE	55
Figure 4.25 - System readings using CNN model.....	56
Figure 4.26 - User Login of GMS.....	59
Figure 4.27 - Starting Custom Application GMS.....	61
Figure 4.28 - Started Prototype Operation in Custom Application GMS	62
Figure 4.29 - Normal Operation Detection in Custom Application GMS.....	62
Figure 4.30 - Abnormal Operation Detection in Custom Application GMS.....	63
Figure 4.31 - Neutral State Detection in Custom Application GMS.....	64
Figure 4.32 - Store Realtime Data on CSV Files	65

LIST OF TABLES

Table 4.1 - Precision, Recall, F1-Score, and Accuracy – Random Forest Model	47
Table 4.2 - Precision, Recall, F1-Score, and Accuracy - CNN Model	50
Table 4.3 - Precision, Recall, F1-Score, and Accuracy - Autoencoders Model	53
Table 4.4 - Models Comparison Table	54
Table 4.5 - Used Python Modules and Description of Login Section.....	58
Table 4.6 - Python Modules Used to Develop GSM Interface.....	61
Table 4.7 - Test Cases Table when Data Collection Python Script Running.....	67
Table 4.8 - Test Cases Table When Identify Abnormality Condition - GMS System's Login Section.....	68
Table 4.9 - Test Cases Table When Identify Abnormality Condition - GMS System	70
Table 5.1 - Budget for System.....	73