AUT Journal of Electrical Engineering
10.22060/EEJ.2026.26338.6107

A LoRa and TinyML Integrated Framework for Operation, Protection and Predictive Maintenance of an Agricultural Three-Phase Induction Motor

Sirivanth Jayachamaraj[1] and Mahadev Halipeth Sidram[1*]

[1]Department of Electrical and Electronics Engineering, Sri Jayachamarajendra College of Engineering, JSS STU, Mysuru, Karnataka, India

[*]Corresponding author: drmhs@jssstuniv.in

Abstract: Agriculture is a vital economic sector in many regions and depends heavily on three-phase induction motors for irrigation pumps, threshers and other farm machinery, making their reliable and uninterrupted operation critical to crop yield and farmer livelihoods. Three-phase induction motors are used widely in industrial and agricultural applications. But these motors are prone to electrical faults like overcurrent, undervoltage and single-phasing. This causes insulation deterioration and unplanned shutdown. Conventional protection depends on fixed threshold relays with no remote supervision or prediction capabilities while cloud dependent Internet of Things systems are impractical in the rural environments without internet infrastructure. This paper proposes a LoRa and TinyML integrated framework which provides real time monitoring, protection, remote control and predictive maintenance of three-phase induction motors in constrained infrastructure environments. The system employs an ESP32 microcontroller, PZEM-004T voltage and current monitoring module, current transformer coil sensors, Reyax RYLR896 LoRa transceivers at 867 MHz and a relay-contactor protection mechanism. The entire process from detecting faults to isolating the motor and TinyML inference thus executes on local edge nodes. The ESP32 on the motor end acts as the Wi-Fi Access Point that supports an operator dashboard on a Local Area Network. The proposed system was validated through MATLAB/Simulink simulation and experimental implementation on a three phase, 2 HP, 415 V, 3.3 A, 50 Hz induction motor.

Keywords: Long-Range (LoRa) communication, Tiny Machine Learning (TinyML), Local Area Network (LAN), Fault detection, ESP32.

10.22060/EEJ.2026.26338.6107

1. INTRODUCTION

Three-phase induction motors are the most widely used electrical machines in industrial, commercial and agricultural sectors. Their popularity stems from the simplicity of construction, high efficiency, mechanical robustness and low maintenance [1,2]. These are used as prime movers in irrigation, pumps, compressors, conveyors, fans and drive systems in the industries where continuous and reliable operation is necessary [3,4]. Agriculture depends primarily on natural rainwater but that can be unreliable and insufficient for optimum crop production throughout the year. Hence, traditional irrigation systems are widely used to artificially supply water to the fields and maintain soil moisture levels. Today, the success of modern farming is highly dependent on the reliability of artificial watering systems to compensate seasonal rains. Induction motors drive water pumps to facilitate large scale irrigation in agricultural environments. Continuous operation of these pumps is essential for enhancing crop productivity and rural livelihood. The characteristic of no brushes, self-starting, speed governing and less manufacturing cost compared to other motors create their large use [5,6]. Induction motors are exposed continuously to electrical stress like voltage supply disturbance, mechanical overload and supply asymmetry which deteriorate motor insulation and mechanical components during their operational lifetime [7].

Three-phase induction motors are prone to electrical faults including overcurrent, undervoltage, phase imbalance and single phasing [1,2,7]. These conditions produce excessive thermal rise, insulation degradation and early failure of the motor making monitoring and protection system necessary [1,2]. Traditional motor protection uses electromechanical relays and locally installed monitoring devices governed by fixed threshold settings, which are sufficient for basic fault isolation but provide no remote supervision, historical data analysis or predictive maintenance capabilities [8,9]. IoT platforms development has led to such remote and real time monitoring system [3-6]; however, the fact that they work only or primarily through networks Wi-Fi or continuous Internet access makes these unsuitable for deployment in agricultural and rural settings where network access is never guaranteed or consistent [10-12]. At the same time, predictive maintenance techniques based on cloud-dependent Machine Learning (ML) approaches [13-18] require expensive high bandwidth data pipes and remote computing resources which cannot be done on cost constrained embedded platforms deployed in infrastructure limited settings [19,20]. The communication barrier is tackled by using a Low-

10.22060/EEJ.2026.26338.6107

Power Wide Area Network (LPWAN) technology like Long-Range (LoRa) Communication that enables reliable multi-kilometer wireless links without the need for the internet infrastructure [10-12,21]. Additionally, algorithmic solutions to deal with microcontroller limitations are addressed by using Tiny Machine Learning (TinyML). This enables quantized, compressed machine learning inference on microcontrollers with power consumption at the milliwatt level [19,20,22]. Despite growing research interest in each of these technologies independently, no prior work has integrated real time fault protection, LoRa communication and edge TinyML predictive maintenance within a single system for remote agricultural induction motors [23].

This paper presents a LoRa and TinyML integrated framework for real time monitoring, control, protection and predictive maintenance of three-phase induction motors, specifically designed for remote agricultural environments lacking conventional communication infrastructure. The proposed system comprises an ESP32 microcontroller, Current Transformer (CT) coil sensors, a PZEM-004T voltage and current monitoring module, Reyax RYLR896 LoRa transceivers operating at 867 MHz and a relay-contactor protection mechanism into a single cloud independent edge architecture. Features like fault detection, motor isolation and TinyML based predictive maintenance execute entirely on the local edge nodes without cloud connectivity. The motor side ESP32 operates as a standalone Wi-Fi Access Point (AP) and embedded web server enabling operator to access real time monitoring dashboard via local IP address over a Local Area Network (LAN) with no dependence on external internet infrastructure. A fully connected neural network trained on the Edge Impulse platform and quantized to INT8 is deployed on the ESP32 for five-class fault classification. The system was validated through MATLAB/Simulink simulation and experimental implementation on a 2 HP, 415 V, 3.3 A, 50 Hz, 1415 RPM three-phase induction motor achieving 98.6% TinyML classification accuracy under laboratory testing and LoRa telemetry upto 2 km in open-field conditions and fault detection response times between 0.5 s to 1 s.

In order to provide a clear overview of the proposed study, the following sections are organized as follows. Section 2 presents a structured literature review covering motor protection and IoT based monitoring, LoRa communication for remote industrial applications and TinyML approaches for edge-based predictive maintenance, identifying the research gap addressed by the present work. Section 3 describes the proposed system architecture, LoRa and TinyML framework design, MATLAB/Simulink motor modelling with the protection subsystem and the fault detection and protection strategy. Section 4 presents the experimental

10.22060/EEJ.2026.26338.6107

results and discussion, covering hardware implementation, simulation and protection performance, TinyML classification accuracy with the full confusion matrix, LoRa communication performance across multiple distances, web-based dashboard operation with fault alert functionality and a comparative analysis with existing works. Section 5 concludes the paper and outlines directions for future work.

2. LITERATURE REVIEW

2.1 Motor Protection and IoT based Monitoring

Three phase induction motors are used widely in industrial and agricultural applications. Their performance can be significantly affected by electrical faults including overcurrent, undervoltage and single-phasing [1,2,7]. Umahon et al. [7] developed a microcontroller-based protection scheme which could identify anomalous conditions and isolate the motor from the AC Supply. Halmare et al. [1] built an Arduino based monitoring and protection scheme which used voltage and current sensing circuits to diagnose faults and trip the motor. While these proposed schemes performed fault protection well, they were intended for local control and did not include remote monitoring and intelligent fault diagnosis. Dawood et al. [2] proposed an efficient protection scheme against single-phasing for three phase induction motor, reporting rapid fault isolation with detection windows consistent with industrial thermal protection requirements.

The advancement of IoT has allowed smart motor monitoring systems to be accessed remotely. Santhosh et al. [5] designed a smart control and monitoring system for a three-phase induction motor which allowed the system to be monitored remotely via a central application. An ESP32 based IoT system was developed by Roslan et al. [6] allowing for the remote control and monitoring of a three-phase induction motor via a cloud-based infrastructure. An IoT enabled three-phase motor protection framework with improved fault response time was suggested by Jadhav et al. [8]. A hardware-based induction motor protection scheme with a variety of faults being protected against was implemented by Prathibanandhi et al. [9]. A hardware-based induction motor monitoring system based on IoT was implemented by Syawali and Meliala [24]. Dehbashi et al. [25] have developed a condition monitoring system using IoT which controls the induction motor management by employing a Raspberry Pi. This system makes use of the cloud so as to monitor motor parameters, increasing the overall visibility but dependent on network connection and are no suitable for rural environment such as agriculture [10,12].

2.2 LoRa Communication for Remote Industrial Applications

LoRa-based communication has been significant research interest for remote monitoring under infrastructural scarcity. Ali et al. [10] provided smart monitoring and control of agricultural pumps using LoRa IoT technology, thereby confirming the ability of this technology for usage in rural deployment environments where Wi-Fi and cellular access is not feasible. Sakthi Kumar et al. [11] provided an IoT-based LoRa motor speed control and monitoring system for industries. Ndukwe et al. [12] developed a LoRa based communication for micro-grids, obtaining more than 90% of packet delivery rate at the range of 4 km for urban tests, with no cloud service. Thus, a baseline is set for using LoRa for private server offline implementation in systems similar to the present work. Gore et al. [21] proposed an intelligent farm monitoring system using LoRa IoT for agriculture applications, also reinforcing the use of LoRa in off-site monitoring systems. It has been verified that LoRa communication using the 867 MHz ISM band and suitable spreading factors enables robust low-power data communication at distant places without reliable network availability for power system monitoring [10-12].

2.3 TinyML and Edge AI for predictive maintenance

TinyML offers a new dimension for bringing machine learning intelligence to the microcontrollers, thereby eliminating the dependence on cloud. Lamba et al. [19] present an overview on TinyML that proved the suitability of frameworks like TensorFlow Lite for microcontrollers, Edge Impulse and TVM for production grade TinyML deployments on the ARM Cortex-M and RISC-V microcontrollers with a memory footprint less than 1 MB of flash and 256 kB of RAM, detailing application cases of TinyML in areas of health care, industrial predictive maintenance and smart agriculture and it showed that INT8 quantization only results in small degradation in accuracy with significant reduction in model size.

ML based predictive maintenance methods applied to induction motors have further validated its capability. Yousuf et al. [13] developed an IoT based framework to monitor induction motors health with sensor data analysis at the cloud to assist with predictive maintenance. Abdulkareem et al. [14] presented research where machine learning algorithms were utilized for prediction of induction motor faults with good results in classification performance. Kavana and Neethi [15] applied a neural network for predictive maintenance of induction motors for fault analysis. Rohouma et al. [16] developed a machine learning based framework for detection of bearing faults in three-phase induction motors. Chevtchenko et al. [17] presented a predictive maintenance model by detecting anomaly with the help of real time IoT data. Though these studies prove the feasibility of ML based diagnostics, their approaches often use cloud computation and do not make it possible for edge only operation.

2.4 Research Gap and Contribution

Based on the literature reviewed, studies have only focused on the motor protection, monitoring based on IoT, LoRa communication and TinyML/edge AI. There are no existing studies that integrate real time protection, internet independent long range wireless communication infrastructure, remote monitoring, TinyML edge based predictive maintenance and intelligent decision making, offline for agriculture induction motors. This research will cover the above research gaps by designing a LoRa and TinyML integrated framework which involves internet independent long-range wireless communication and machine learning at the embedded level, which is reliable for remote agriculture application and allows for real time protection and predictive maintenance.

3. METHODOLOGY

3.1 Proposed System Architecture

The proposed system includes monitoring, fault protection, long range wireless communication and predictive maintenance into an independent from cloud-based edge architecture for three-phase induction motor. Motor node ESP32 controller, PZEM-004T phase voltage and current monitoring module, Reyax RYLR896 LoRa transceiver and relay-contactor protection unit comprise the proposed system at motor side. The ESP32 at the motor node collects and analyses motor electrical parameters continuously, implements the logic for detection motor faults, operates the relay-contactor mechanism when a fault like overcurrent, undervoltage and single phasing is detected and makes prediction on predictive maintenance of the motor over TinyML approach and broadcasts the system information via LoRa RF communication to operator side. Also, the ESP32 at the motor end runs an offline Access Point (AP) mode with web server on board which hosts operator interface at local IP address, so the operator is able to check the monitoring page on a device connected to the local hotspot, without relying on external infrastructure or services like the internet or cloud. This design gives an Offline Local Area Network (LAN) system architecture since all the operations like motor control, fault isolation and TinyML prediction is being performed on the edge and local

10.22060/EEJ.2026.26338.6107

Wi-Fi AP provides monitoring page to the operator and logging functionality on the local LAN network. Block Diagram of the proposed system is shown in below Fig. 1.

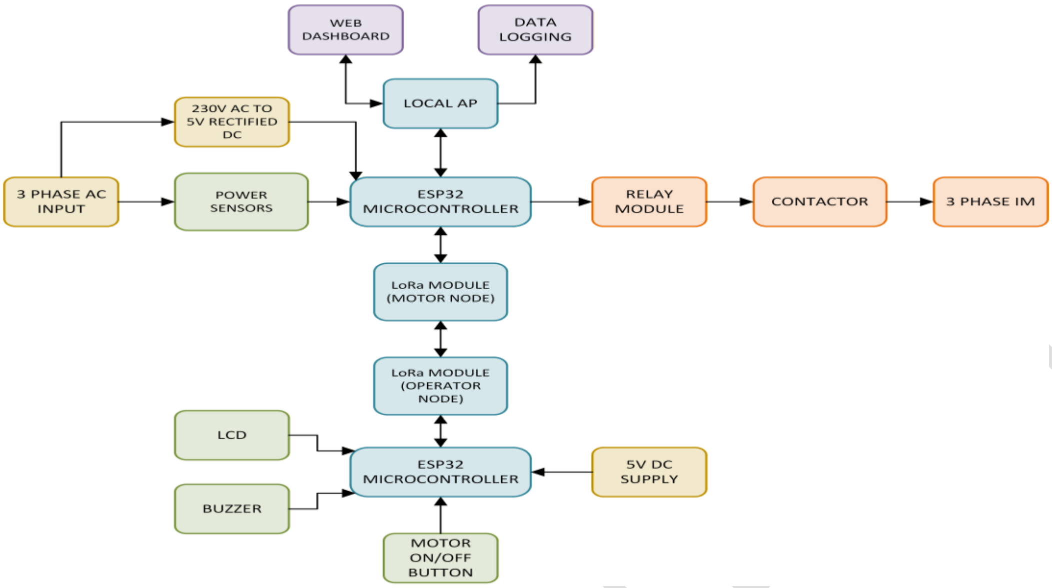

Fig. 1. Block Diagram of the Proposed System

3.2 LoRa and TinyML Predictive Maintenance Framework

A direct point-to-point connection is established between motor and operator nodes by the Reyax RYLR896 LoRa transceivers which operate in the 867 MHz ISM band. This point-to-point link is set with LoRa parameters of Spreading Factor 12, bandwidth 125 kHz, coding rate 4/5 and transmit power 15 dBm. LoRa was chosen for its low power, multi-kilometer communication range and complete lack of infrastructure over and above the two transceivers. There is no Wi-Fi, cellular, internet or cloud link present in the LoRa path so deployment and operation cost are extremely low compared to typical agricultural IoT deployments.

The ESP32 motor node has TinyML application, for predictive maintenance, running on the edge that requires no cloud computing resources. Motor line-to-line voltage, phase current and power factor were chosen as features because they are already known to be sensitive to the fault classes of interest. Healthy operation data, together with data representative of overcurrent, undervoltage and single-phasing conditions were recorded along with some synthesized winding degradation samples, which represented progressive failure of the motor's performance. All end to end modelling including spectral features, training and INT8 quantization was performed using the Edge Impulse platform.

Data were collected via a sliding-window protocol, each sample consisted of 500 ms of sequential sensor readings collected at 100 Hz, resulting in 91 spectral features per sample on 7 input channels ($V_{\mathrm{r}}, V_{\mathrm{y}}, V_{\mathrm{b}}, I_{\mathrm{r}}, I_{\mathrm{y}}, I_{\mathrm{b}}$ and Power Factor (PF)). This windowing method allows the capture of transient signatures such as the onset of asymmetry in the currents during single-phasing and modulation of the currents during voltage drop conditions, which is not possible using single point measurement. The dataset was composed as detailed below in Table 1. A fully connected neural network was trained with 5 classes, 91 features and 20 nodes on the hidden dense layer. The full pipeline model was quantized to 8-bit integer, used 16.1 kB of flash memory and 1.5 kB of RAM on the ESP32, with a total inference latency of 14 ms per cycle. Real time inference executes locally on the ESP32 edge node without any cloud interaction. The 91 spectral features per sample are extracted by the Edge Impulse Spectral Analysis block, which applies a Fast Fourier Transform (FFT) over each 500 ms window and computes per-channel spectral power values, RMS, skewness, kurtosis and peak frequency statistics. For each of the 7 input channels ($V_{\mathrm{r}}, V_{\mathrm{y}}, V_{\mathrm{b}}, I_{\mathrm{r}}, I_{\mathrm{y}}, I_{\mathrm{b}}$, PF), 13 features are extracted (7 x 13 = 91), capturing both frequency-domain and statistical time-domain characteristics that discriminate among the five fault classes.

Table 1. TinyML Dataset Description

Operation Condition	Total Samples	Train (80%)	Validation (20%)	Input Features	F1 Score
Healthy Operation	500	400	100	Voltage, Current, PF (500 ms @ 100 Hz)	0.97
Overcurrent Fault	500	400	100	Voltage, Current, PF (500 ms @ 100 Hz)	1.00
Undervoltage Fault	500	400	100	Voltage, Current, PF (500 ms @ 100 Hz)	1.00
Single Phasing Fault	500	400	100	Voltage, Current, PF (500 ms @ 100 Hz)	1.00
Winding Degradation	500	400	100	Voltage, Current, PF (500 ms @ 100 Hz)	0.96

3.3 Motor Modelling and Fault Analysis

The 3 phase, 2 HP, 415 V, 3.3 A, 50 Hz, 1415 RPM induction motor was modelled in MATLAB/Simulink to test the proposed framework on healthy and fault conditions before applying it to the hardware. The model contained the following blocks: Three phase source block, three-phase V-I measurement block, custom protection system, three-phase breaker block and a squirrel cage induction motor block. The protection system computed each the three stator current signals on a per phase RMS basis by squaring, low-pass filtering and taking the square root of each signal. Any phase RMS current above a defined overcurrent trip value would initiate an overcurrent (OC) signal trip any phase current below a minimum detection threshold for 500 ms or longer would initiate a Single-Phasing (SP) signal trip. SR latch logic generated latched OC signal and SP signal trip outputs fed through an OR gate to the relay. Faults were induced via three Step input blocks that varied the mechanical load torque and opened individual phase breakers at specified simulation times. Stator current, rotor current, electromagnetic torque and rotor speed outputs of the motor were observed through Bus Selector and Mux blocks, connected to scope displays.

The three-phase active power absorbed by the motor is described by Eq. (1)

$$P = \sqrt{3}\, V_L I_L \cos\phi \tag{1}$$

where P is the three phase active power (W), V_L is the line-to-line voltage (V), I_L is the line current (A) and $\cos\phi$ is the power factor. The line current is derived as Eq. (2)

$$I_L = \frac{P}{(\sqrt{3}\, V_L \cos\phi)} \tag{2}$$

The synchronous speed and per-unit slip of the motor is shown in Eqs. (3) and (4)

$$N_s = \frac{120f}{P} \tag{3}$$

$$s = \frac{(N_s - N_r)}{N_s} \tag{4}$$

where N_s is the synchronous speed (rpm), f is the supply frequency (Hz), P is the number of poles, N_r is the rotor speed (rpm) and s is the per-unit slip. The electromagnetic torque developed by the motor is shown below in Eq. (5)

$$T = \frac{(9550 \times P_{kW})}{N_r} \tag{5}$$

10.22060/EEJ.2026.26338.6107

where T is the electromagnetic torque (Nm). The mechanical power balance in the motor is shown below in Eq. (6)

$$P_{\mathrm{mech}} = \omega T = \frac{2\pi N_r T}{60} \tag{6}$$

Eqs. (1) to (6) provide the nominal operating point and also define the limits for parameterization of the Simulink model. Test for mechanical overload, causing overcurrent and phase opening to induce single-phasing were then applied sequentially in simulation to observe the threshold operation and relay trip response before attempting to implement on the hardware.

3.4 Fault Detection and Protection Strategy

The proposed approach is hardware-based fault detection and isolation, completely performed on the ESP32 motor side node. The motor electrical parameters are continuously sampled and compared against the protection threshold values defined in below Table 2. Overcurrent fault is triggered when any phase current value goes above maximum limit, undervoltage fault when any phase voltage is below minimum limit and single phasing fault when any phase voltage or current value remains below detection limit for more than 25 consecutive cycles of the 50Hz supply. When a fault is triggered, the ESP32 controls the relay control output to de-energize the contactor and disconnect the motor from the three-phase power supply. While triggering the protection function, the transmitted data containing fault type, time and measured parameter value are also sent through the LoRa link to operator node for visualizing on the operator display at web interface and the data is logged to google sheets for future references.

Table 2. Fault Detection Threshold (Simulation derived thresholds; for hardware implementation, the overcurrent threshold was reduced to 3 A, which is shown in Table 3)

Fault	Threshold	Protection Action
Overcurrent	Phase Current > 10 A	Motor Trip (Contactor Open)
Undervoltage	Phase Voltage < 180 V	Motor Trip (Contactor Open)
Single Phasing	Any Phase V < 10 V or I < 0.3 A	Motor Trip (Contactor Open)

4. RESULTS AND DISCUSSION

4.1 Hardware Implementation

The proposed framework was physically implemented using ESP32 microcontroller, CT coils, PZEM-004T (power monitoring module), Reyax RYLR896 (LoRa transceivers) and a relay module controlled by an ESP32 and a Schneider TeSys D LC1D12R7 contactor with rating of 440 V, 12 A. This complete hardware was interconnected with a 3 phase, 2 HP, 415 V, 3.3 A, 50Hz, 1415 RPM squirrel-cage induction motor in order to verify its monitoring, communication, protection and prediction maintenance operations in real time. Also, an autotransformer was used to control the voltage supplied to the experimental system as shown below in Fig. 2.

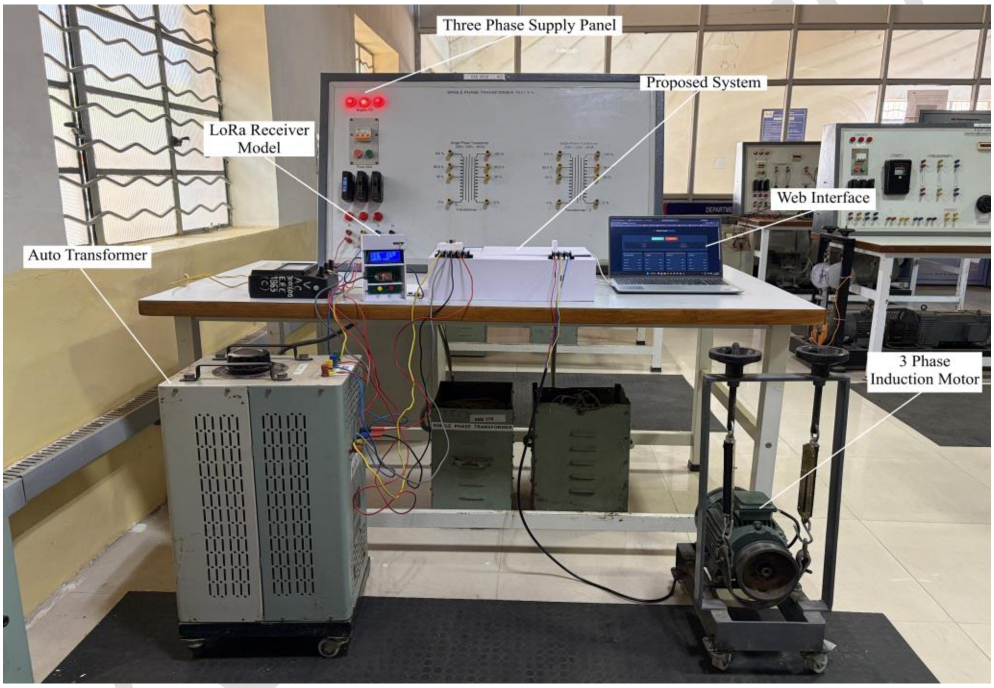

Fig. 2. Proposed System Experimental Setup

The system successfully monitored the voltage and current parameters of the motor at a real time, transmitted its operation data via LoRa link and initiated the fault protection process with relay-contactor disconnections as well as the TinyML inference was operated at local ESP device. The operator dashboard on the local Wi-Fi AP at motor side ESP32, which was displayed from any device that is connected by using standard browser, ensured that the whole system was functioning in LAN configuration without any internet access.

4.2 Simulation Results

10.22060/EEJ.2026.26338.6107

Before testing on the hardware, the motor model was simulated in MATLAB/Simulink to study its behaviour under normal operation, fault conditions and protection action. Under normal operating condition with no load till 0.5 s and rated load from 0.5 s to 7s, the motor draws a high starting transient current before settling into a steady, balanced oscillation around 2.57 A rms, as shown in below Fig. 3. This confirms stable operation once the motor reaches steady state.

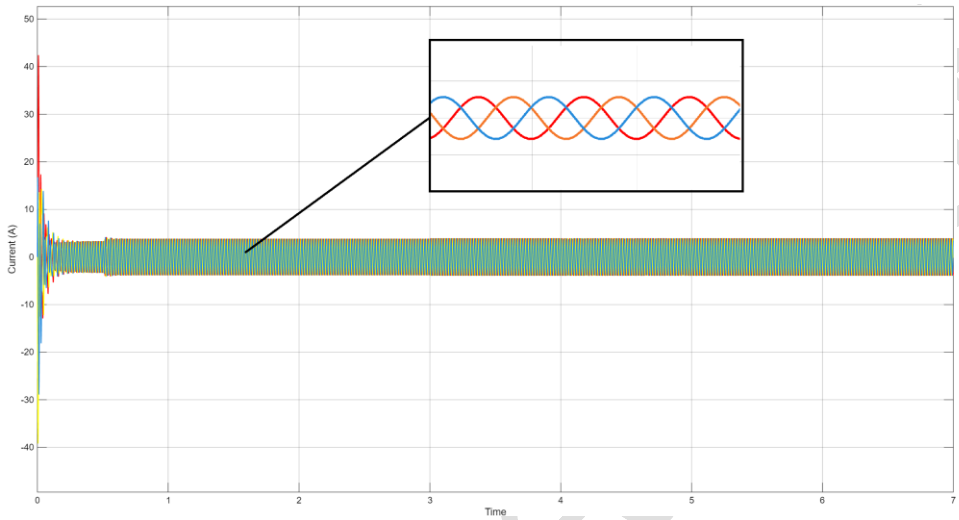

Fig. 3. Stator current at Normal operating condition of the motor

In order to simulate fault conditions, the motor was run at a rated mechanical load of 10 N-m from 0.5 s to 3 s. From 3 s to 4 s, an additional 10 N-m was applied, taking the total load to 20 N-m and driving the phase current up to approximately 10 A, matching the overcurrent threshold defined in Table 2 above. The load was then brought back to 10 N-m from 4 s to 5 s, after which one phase was opened from 5 s to 7 s to introduce a single-phasing condition, seen as the remaining phase current rising while the open phase drops to zero. Fig. 4 below shows both fault conditions captured in sequence.

Fig. 5 below shows the protection action in simulation. As the fault current persists and crosses the defined threshold, the protection logic generates a trip signal and the relay/breaker block disconnects the motor from the supply, bringing the current down to zero. This confirmed that the protection logic worked correctly in simulation before it was implemented on the hardware.

10.22060/EEJ.2026.26338.6107

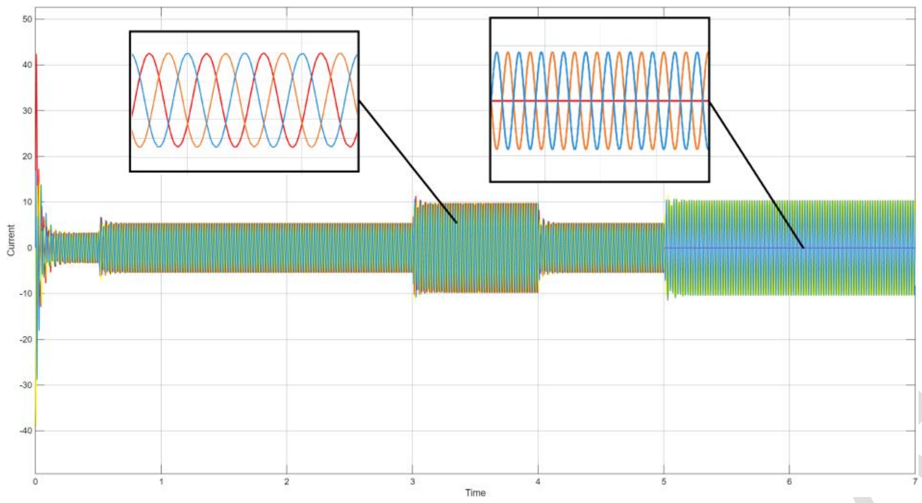

Fig. 4. Stator current at Fault condition of the motor

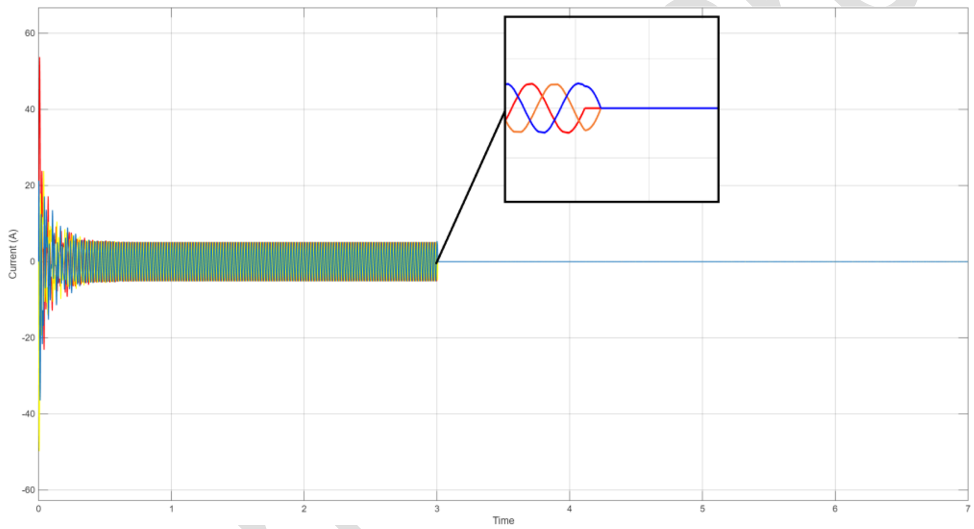

Fig. 5. Stator current at fault condition with protection system.

4.3 Hardware Fault Detection and Response Performance

The fault thresholds in Table 2 were derived from the MATLAB/Simulink model and used as the design reference. On the hardware, the undervoltage and single-phasing thresholds were implemented as designed. For overcurrent, the threshold was adjusted before testing on the actual motor. The simulation threshold of 10 A is nearly three times the rated current of the motor (3.3 A), and repeatedly loading the real motor up to that point for demonstration risked damaging it. The motor's measured no-load current was about 2.07 A, so it was gradually loaded until the current reached 3 A, and the overcurrent threshold was set at this value. Table 3 below summarizes this deviation between the simulation derived and hardware implemented thresholds.

Table 3. Simulation Derived vs. Hardware Implemented Thresholds

Fault Type	Simulation Threshold	Hardware Threshold	Reason for Deviation
Overcurrent	Phase Current > 10 A	Phase Current > 3 A	Motor protected from repeated stress during demonstration. 10 A (~3x rated current) risked damage on repeated trials
Undervoltage	Phase Voltage < 180 V	Phase Voltage < 180 V	No deviation
Single Phasing	Any Phase V < 10 V or I < 0.3 A	Any Phase V < 10 V or I < 0.3 A	No deviation

i. Overcurrent

The motor was loaded using a spring balance and belt arrangement. The belt tension was increased gradually to raise the mechanical braking load on the motor shaft. Table 4 shows the phase current recorded as the load was increased. When the current reached the set threshold of 3 A, the ESP32 detected the overcurrent condition and de-energized the contactor. The motor was then isolated from the connected three phase supply within the response time given in the below Table 4.

Table 4. Current vs. Load (Belt Tension) During Overcurrent Test

Applied Belt Tension (kg)	Phase Current (A)	Contactor Status
0	2.07	Closed (Motor Running)
1.90	2.60	Closed (Motor Running)
2.00	2.80	Closed (Motor Running)
2.10	3.0	Open (Motor Isolated)

ii. Undervoltage

The voltage supplied to the motor was lowered below the set threshold of 180 V by means of a three-phase autotransformer. When the phase voltage was measured below this value, the ESP32 detected the undervoltage condition and tripped the relay and the contactor disconnected the motor within the response time specified in the below Table 5.

10.22060/EEJ.2026.26338.6107

iii. Single Phasing

One of the three supply phases was manually disconnected. The ESP32 detected the resulting near zero voltage and current on the open phase, identified it as a single-phasing condition and de-energized the contactor within the response time given in Table 5 below.

Table 5. Fault response characteristics

Fault Type	Detection time (s)	Protection Action
Overcurrent	0.5-0.8	Contactor de-energizes, Motor Isolated
Undervoltage	0.6-0.9	Contactor de-energizes, Motor Isolated
Single Phasing	0.5-0.7	Contactor de-energizes, Motor Isolated

The detection times account for sensor sampling delay, firmware threshold detection time and relay mechanical switching time. The total detection time of 0.5–0.9 s includes four parts: sensor sampling delay (~100 ms/cycle), TinyML inference time (14 ms) on the ESP32, firmware confirmation cycles depending on fault type, and relay mechanical switching (~30 ms). The pure TinyML processing time is 14 ms per cycle, which is fast enough for continuous real-time motor monitoring. For a 2 HP motor with a thermal time constant of 1800 seconds, these detection times provide a sufficient margin before insulation is damaged.

Each fault condition was tested for 20 different trials to ensure consistency and detection time of the fault. Table 6 below shows the detection time recorded for each trial, along with the computed mean and standard deviation (SD) of the response time.

Table 6. Repeatability of Fault Detection Across 20 Trials

Trial	Overcurrent (s)	Undervoltage (s)	Single Phasing (s)
1	0.7	0.8	0.6
2	0.5	0.9	0.5
3	0.6	0.6	0.6
4	0.5	0.8	0.5
5	0.6	0.6	0.7

10.22060/EEJ.2026.26338.6107

6	0.5	0.6	0.5
7	0.7	0.9	0.5
8	0.5	0.9	0.5
9	0.8	0.8	0.5
10	0.8	0.6	0.6
11	0.6	0.7	0.4
12	0.5	0.7	0.6
13	0.6	0.7	0.5
14	0.7	0.8	0.7
15	0.6	0.7	0.6
16	0.8	0.6	0.7
17	0.5	0.9	0.5
18	0.5	0.6	0.5
19	0.7	0.8	0.5
20	0.6	0.6	0.6
Mean ± SD	0.61 ± 0.109	0.73 ± 0.117	0.55 ± 0.083

4.4 TinyML Classification Performance

2500 samples were used to train the TinyML model across the five fault classes in Edge Impulse (500 samples per class: 400 for training, 100 for validation, an 80/20 split). The features were taken from spectral data collected from the sensors over a sliding window of 500 ms at 100 Hz, producing 91 features for each sample from 7 input channels, with real sensor noise and line parameter variations included during data collection to improve generalization. The resulting neural network used was fully connected with a 91-neuron input, 20-neuron hidden and 5-neuron output layer and achieved 98.6% accuracy on the held-out validation set under controlled lab conditions (loss: 0.11). Healthy, Overcurrent, Single Phasing and

10.22060/EEJ.2026.26338.6107

Undervoltage classes showed perfect F1 scores of 1.00. Winding Degradation achieved an F1 score of 0.96, with 8 of 100 validation samples misclassified as Healthy, reflecting the electrical similarity between early winding degradation and low-load healthy operation. The weighted average precision, recall and F1 score across all classes were 0.99, with an Area under ROC Curve of 1.00. Fig. 6 below shows the full confusion matrix with absolute sample counts across 500 validation samples. The network requires 16.1 kB flash memory and 1.5 kB of RAM after INT8 quantization on the ESP32 and 14 ms per cycle for inference, which is clearly fast enough for continuous motor monitoring.

To justify the use of a neural network over simpler classifiers, the same 7-feature raw sensor dataset (2000 training, 500 test samples) was evaluated using a Fixed Threshold rule, Decision Tree (max depth 10), SVM with RBF kernel (C=10), and KNN (k=5) implemented in Python (scikit-learn). Results are summarized in Table 7 below. The Fixed Threshold, SVM and KNN classifiers all achieved 80.0% accuracy but completely failed on at least one fault class: Overcurrent was undetectable by SVM and KNN (F1=0.00), while Winding Degradation was undetectable by the Fixed Threshold method (F1=0.00). The Decision Tree performed worst at 38.2%, failing on three of five fault classes. The proposed neural network (TinyML on ESP32) achieved 98.6% accuracy with a weighted F1 of 0.99, correctly classifying all five fault types. Although the neural network inference time of 14 ms per sample on the ESP32 is higher than the software-based classifiers, it remains well within the motor protection time budget and operates entirely on-device without cloud dependency, demonstrating a clear and significant advantage over simpler approaches.

10.22060/EEJ.2026.26338.6107

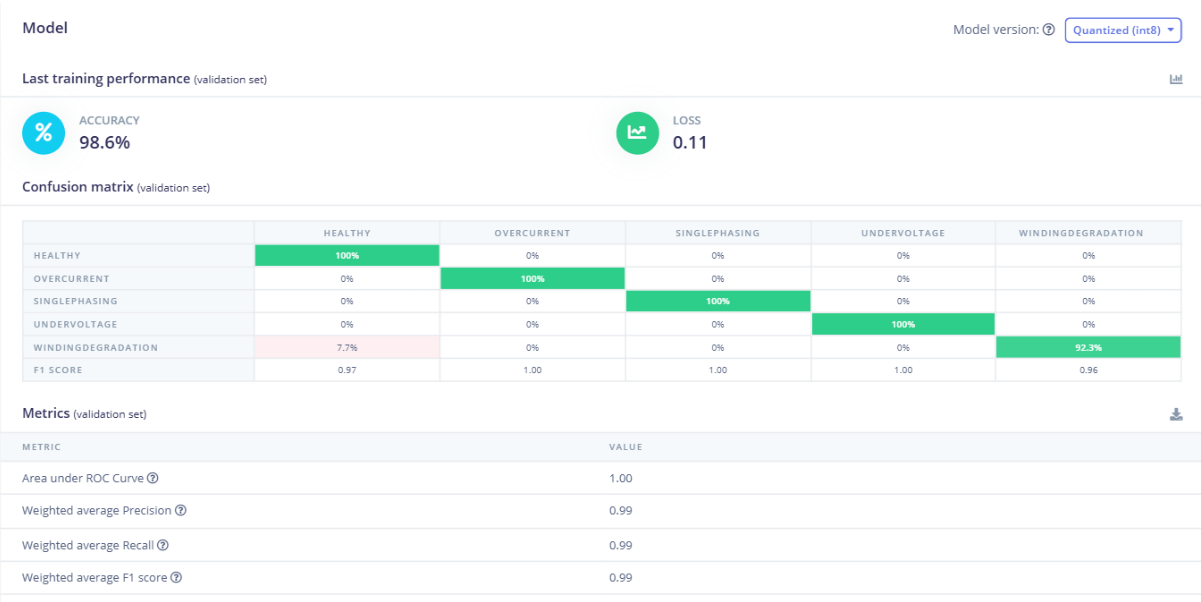

Fig. 6. Edge Impulse TinyML Model Performance – Confusion Matrix and Metrics (Validation Set, 500 Samples, 98.6% Accuracy)

Table 7. Classifier Comparison on Five-Class Motor Fault Dataset (500-Sample Test Set)

Classifier	Accuracy (%)	Wtd. F1	Inference (ms/sample)	Limitations
Fixed Threshold	80.0	0.733	<0.001	Cannot detect Winding Degradation (F1=0.00).
Decision Tree	38.2	0.324	0.001	Fails on Overcurrent, Single Phasing, Undervoltage.
SVM (RBF kernel)	80.0	0.733	0.024	Cannot detect Overcurrent (F1=0.00).
KNN (k=5)	80.0	0.733	0.008	Cannot detect Overcurrent (F1=0.00).
NN-TinyML (Proposed)	98.6	0.99	14.0*	Best overall; *14 ms on ESP32 (hardware constraint).

4.5 LoRa Communication Performance

LoRa communication was tested using the Reyax RYLR896 transceiver pair at 867 MHz with AT+PARAMETERS set to SF=12, BW=125 kHz, CR=4/5 and TX power of 15 dBm, 92.5% of Packet Delivery Ratio (PDR) could be received with 450 m of range in urban environments with high radio interference along with walls and buildings. In an agricultural setting (line-of-sight), the range was extended to almost 2 km while still achieving 89.2% PDR at -124 dBm RSSI. The parameters tested can be found below in Table 8. This indicates that

LoRa is suitable for use in remote agricultural areas where there is neither cell services nor Wi-Fi. The observed 92.5% and 89.2% PDR with SF12 is acceptable with the current LoRa link budget. Increased spreading factor provides greater link margin at the expense of data rate. Temperature variation in real time agricultural environments can affect sensor accuracy and ESP32 performance. The PZEM-004T current and voltage sensors have a specified operating range of 0 to 40 °C with accuracy within ±0.5% over this range. The ESP32 module operates reliably between -40 °C and 85 °C. All hardware experiments were performed within the temperature range of 20 °C to 31 °C, which is in the above range. For deployments in areas with temperatures above 40 °C, sensor recalibration and ESP32 thermal management should be considered to ensure measurement accuracy.

Table 8. LoRa Link Performance vs. Distance

Distance (m)	Environment	RSSI (dBm)	PDR (%)	Link Status
50	Indoor Lab (reinforced concrete building, 2.4/5 GHz Wi-Fi present, ~20 °C)	-72	100	Excellent
150	Indoor Corridor (1 concrete wall, moderate Wi-Fi, ~22 °C)	-88	99.1	Excellent
300	Indoor multi-wall (3 concrete walls, dense Wi-Fi/BT interference, ~24 °C)	-101	96.8	Good
450	Indoor Max Range (4+ walls, high RF interference, ~24 °C)	-112	92.5	Good
800	Open Field (flat agricultural land, clear line-of-sight, ~28 °C, no obstacles)	-116	95.4	Good
1500	Open Field (flat agricultural land, sparse vegetation, ~30 °C)	-120	91.7	Satisfactory
2000	Open Field (max) (flat agricultural	-124	89.2	Satisfactory

10.22060/EEJ.2026.26338.6107

	land, clear sky, ~31 °C, no obstacles)	

4.6 Web-Based Monitoring and Control Dashboard

The operator monitoring and control interface has been designed as a web-based application hosted directly by the motor side ESP32. This ESP32 functions as a standalone Wi-Fi AP (at 2.4 GHz) that operators can connect to with any Wi-Fi device and open up a web browser to view the system's IP address to gain full access without any connection to the outside world, no DNS, no internet and no cloud systems. Operators are able to view current, voltage, power factor, motor status and faults and can also directly command the motor to start and stop as shown below in Fig. 7.

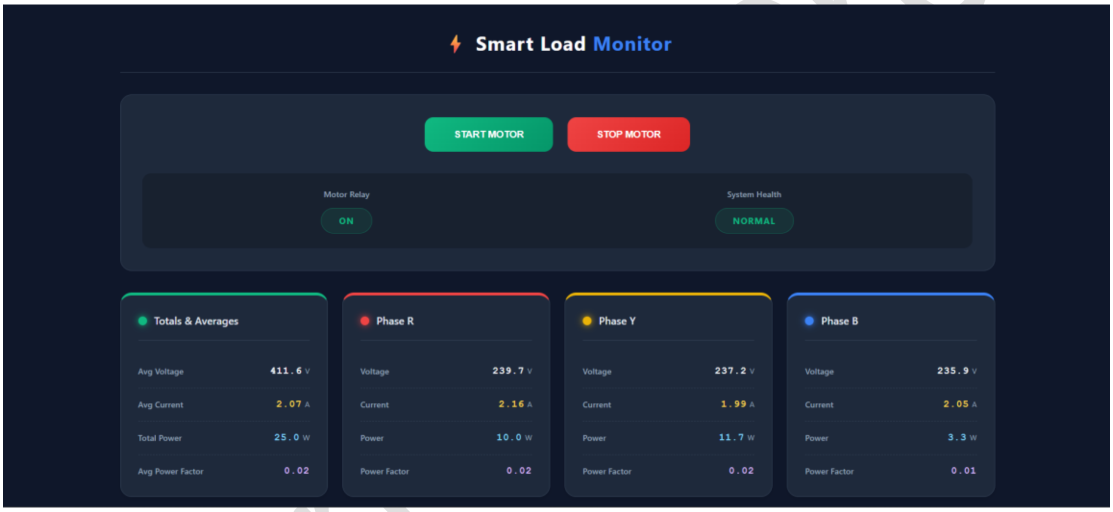

Fig. 7. Web Interface

Control commands are transmitted via LoRa between the receiver ESP32 and motor side ESP32, which directly actuates the relays to start/stop the motor. Alerts can be directly popped up to the operator immediately as fault condition occurs, as illustrated below in Fig. 8 where an undervoltage alert message from the system is displayed to the operator and that can be acknowledged.

10.22060/EEJ.2026.26338.6107

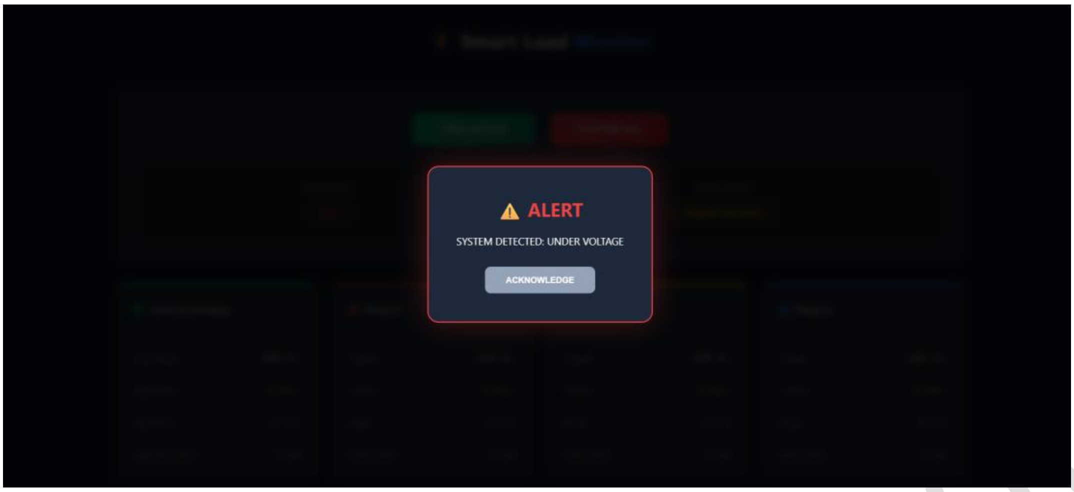

Fig. 8. Alert indication on Web Interface

Motor parameters acquired can be optionally logged directly into Google Sheets as shown in below Fig. 9, if the ESP32 is linked to an operator's device hotspot. All basic functionalities are stand alone.

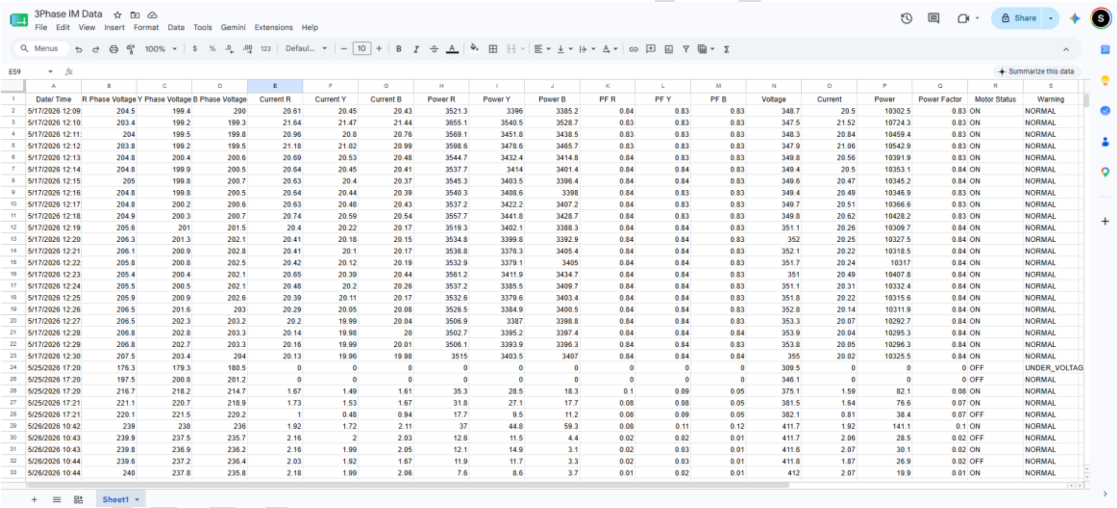

Fig. 9. Data Logging in Google Sheets

4.7 Comparative Analysis with Existing Works

Table 9 below provides a comparative analysis of the developed system and relevant works across nine functionality dimensions. Existing systems individually provide motor protection, IoT-based remote monitoring, LoRa based communication, machine learning based fault analysis and TinyML embedded intelligence. The proposed system is distinguished by four characteristics absent from all prior works. (a) LoRa based long range communication independent of cellular or internet infrastructure, (b) TinyML/edge AI inference deployed directly on the microcontroller without cloud resource, (c) a fully self-contained offline LAN architecture eliminating cloud dependency while preserving operator dashboard access and (d) integration of all capability dimensions within a single low-cost hardware platform.

Table 9. Comparative Analysis with Existing Studies

Feature	[2]	[5]	[6]	[7]	[10]	[12]	[13]	Prop.
Fault Protection	Yes	Yes	Yes	Yes	No	No	No	Yes
Remote Monitoring	No	Yes	Yes	No	Yes	Yes	Yes	Yes
Remote Control	No	Yes	Yes	No	No	No	No	Yes
LoRa Communication	No	No	No	No	Yes	Yes	No	Yes
Data Logging	No	Yes	Yes	No	No	Yes	Yes	Yes
Predictive Maintenance	No	No	No	No	No	No	Yes	Yes
TinyML/Edge AI	No	No	No	No	No	No	No	Yes
Cloud-Independent Oper.	Yes	No	No	Yes	No	Yes	No	Yes

5. CONCLUSION AND FUTURE WORK

This work presented a LoRa and TinyML integrated framework was developed to monitor, control, protect and perform predictive maintenance of three-phase induction motor for remote agricultural applications where conventional communication infrastructure is not available. All the motor protection, isolation and predictive maintenance inferencing are performed at the local edge nodes. There are no cloud server or internet connection requirement. ESP32 at motor side acts as a local Wi-Fi Access Point and an embedded web server that allows operator to access the real-time monitoring dashboard and local logs via self-constrained LAN network.

The proposed framework was validated through both MATLAB/Simulink simulation and experimental implementation on a 3 phase, 2 HP, 415 V, 3.3 A, 50 Hz, 1415 RPM induction motor. Relay-contactor based protection successfully identified and isolated overcurrent, undervoltage and single phasing fault between 0.5 s to 1 s. The LoRa link, based on 867MHz with SF 12/BW 125, maintained stable communications up to 450 m indoors and 2 km open field, with packet delivery ratio > 89%. The proposed TinyML model is a fully connected neural network quantized at INT8 and deployed on ESP32 via Edge Impulse, which achieved 98.6% classification accuracy over all 5 fault types under laboratory testing, with per-inference latency of 14 ms, flash memory footprint of 16.1 kB and RAM usage 1.5 kB to perform real-time predictive maintenance without any computation infrastructure.

Compared to all previous work in this domain, the proposed approach has simultaneously introduced LoRa based infrastructure free communication and Edge AI inference, offline LAN based operation and full fault protection which has been unified in a single low-cost hardware. In future work, the proposed method can be extended to LoRa mesh network deployed at large scale agricultural pump stations. In addition, adding sensors such as temperature-based winding degradation could complement the electrical features for predictive maintenance and can also investigate adaptive retraining mechanism for TinyML model to account for motor ageing and operation conditions variability and study on early exit thresholder classification approach to minimize latency during frequent monitoring.

REFERENCE

[1] A.M. Halmare, T. Meshram, N. Narnaware, et al., Monitoring and protection of three-phase induction motor in case of abnormal conditions, International Journal of Engineering Research and Technology, 12(4) (2025) 34–39.

[2] A. Dawood, M.A. Ismeil, H.S. Hussein, et al., An efficient protection scheme against single-phasing fault for three-phase induction motor, IEEE Access, 12 (2024) 6298–6317.

[3] A.A. Musyafiq, M.H. Akbar, Purwiyanto, et al., IoT-based control, monitoring, and protection system for 3-phase induction motors in electric motorcycles, Journal of Power, Energy, and Control, 1(2) (2024) 103–110.

[4] A.M. Rajput, G. Abid, S.A. Shaikh, et al., IoT based smart industrial panel for controlling three-phase induction motor, in: Proceedings of the 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2020, pp. 1–5.

[5] R. Santhosh, Sailakshmi, V.M. Siddharthan, et al., Protection and control of 3 phase induction motors in industrial plants with smart remote monitoring and control on a

centralized system, in: Proceedings of the 6th International Conference on Trends in Electronics and Informatics (ICOEI), 2022, pp. 317–326.

[6] M.A.H. Roslan, J. Sampe, N.H.M. Yunus, et al., Remote controlling of three-phase induction motor based on auto trans starter with IoT application, in: Proceedings of the IEEE 10th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), 2024, pp. 1–6.

[7] O. Umahon, J.D. Keshi, I.S. Ejededawe, Microcontroller based fault detection and protection of three-phase induction motor against abnormal conditions, Global Scientific Journal, 7(2) (2019) 30

[8] N.K. Jadhav, R.S. Kale, V.C. More, et al., The novel IoT-enabled protection for three phase induction motor, International Journal of Creative Research Thoughts, 13(5) (2025) d766–d772.

[9] K. Prathibanandhi, Thejaswarya, V. Malini, et al., Induction motor protection system, in: Proceedings of the International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), 2024, pp. 1–5.

[10] A.H. Ali, R.F. Chisab, M.J. Mnati, A smart monitoring and controlling for agricultural pumps using LoRa IoT technology, Indonesian Journal of Electrical Engineering and Computer Science, 13(1) (2019) 286–292.

[11] B.S. Kumar, S. Amruthavarshini, S. Ramalingam, et al., LoRa-IoT based industrial automation motor speed control monitoring system, in: Proceedings of the International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), 2023, pp. 1–6.

[12] C. Ndukwe, M.T. Iqbal, X. Liang, et al., LoRa-based communication system for data transfer in microgrids, AIMS Electronics and Electrical Engineering, 4(3) (2020) 303–325.

[13] M. Yousuf, T. Alsuwian, A.A. Amin, et al., IoT-based health monitoring and fault detection of industrial AC induction motor for efficient predictive maintenance, Measurement and Control, 57 (2024) 1146–1160.

[14] A. Abdulkareem, T. Anyim, O. Popoola, et al., Prediction of induction motor faults using machine learning, Heliyon, 11(1) (2025) e41493.

[15] V. Kavana, M. Neethi, Fault analysis and predictive maintenance of induction motor using machine learning, arXiv preprint arXiv:2409.09944, (2024).

[16] W. Rohouma, A. Zaitouny, M.F. Wahid, et al., A machine learning framework for bearing fault detection in three-phase induction motors, in: Proceedings of the 4th International Conference on Smart Grid and Renewable Energy (SGRE), 2024, pp. 1–5.

[17] S.F. Chevtchenko, M.C.M. dos Santos, D.M. Vieira, et al., Predictive maintenance model based on anomaly detection in induction motors: A machine learning approach using real-time IoT data, arXiv preprint arXiv:2310.14949, (2023).

[18] T.A. Smrity, M.Z. Muntaqim, H.M. Kafi, et al., Hybrid CNN-BYOL approach for fault detection in induction motors using thermal images, arXiv preprint arXiv:2510.07692, (2025).

[19] A.K. Lamba, D. Pal, S. Koli, et al., TinyML for edge devices: Current state, technological advancements, and future opportunities, in: Proceedings of the 1st International Conference on Advancing Sustainable Solutions through Technology (ICASST), 2026, pp. 1–6.

[20] M. Altayeb, M. Zennaro, P. Manzoni, Edge ML vs. TinyML: A comparative analysis with experimental results on power efficiency and model optimization, in: Proceedings of the IEEE 23rd Consumer Communications & Networking Conference (CCNC), 2026, pp. 1–6.

[21] S. Gore, S. Patil, V. Khalane, Intelligent farm monitoring system using LoRa enabled IoT, in: Proceedings of the IEEE Bombay Section Signature Conference (IBSSC), 2022, pp. 1–5.

[22] T. Puslecki, K. Walkowiak, A novel one-versus-all approach for multiclass classification in TinyML systems, IEEE Embedded Systems Letters, 17(1) (2025) 71–74.

[23] S.J. Suji Prasad, M. Thangatamilan, M. Suresh, et al., An efficient LoRa-based smart agriculture management and monitoring system using wireless sensor networks, International Journal of Ambient Energy, 43(1) (2022) 5447–5450.

10.22060/EEJ.2026.26338.6107

[24] R. Syawali, S. Meliala, IoT-based three-phase induction motor monitoring system, Journal of Renewable Energy, Electrical, and Computer Engineering, 3(1) (2023) 12–18.

[25] N. Dehbashi, M. SeyyedHosseini, A. Yazdian-Varjani, IoT based condition monitoring and control of induction motor using Raspberry Pi, in: Proceedings of the 13th Power Electronics, Drive Systems, and Technologies Conference (PEDSTC), 2022, pp. 1–5.