International Journal of Scientific Research & Engineering Trends
Volume 11, Issue 6, Nov-Dec-2025, ISSN (Online): 2395-566X

A Low-Cost Self-Healing Smart Grid Prototype Using Embedded Random Forest Classification and ESP-NOW Wireless Coordination

Angel Lalu[1], Dr Prakash R[2], Shreyas Sunil[3], Nandhakumar S[4], Divya Bharti[5]
[2]Professor, Department of Electrical and Electronics Engineering
Acharya Institute of Technology, Bengaluru, India
[1,3,4]Undergraduate Students, Department of Electrical and Electronics Engineering
Acharya Institute of Technology, Bengaluru, India
[5]Undergraduate Student, Department of Computer Science and Engineering
Acharya Institute of Technology, Bengaluru, India.

Abstract- Self-healing distribution systems are one of the foundational requirements for future smart grids that are built to withstand disturbances, accommodate bidirectional power flow, and also maintain reliability despite the threat of in- creasing renewable penetration. Traditional FLISR (Fault Location, Isolation, and Service Restoration) solutions used currently depend mostly on SCADA, PMUs, and other high- cost protection relays. This infrastructure is usually not un- available in low-voltage networks, microgrids, and academic environments for teaching purposes. Our work proposes a novel low-cost, microcontroller-based self-healing grid pro- totype that uses ACS712 current sensors, ESP32/ESP8266 wireless sensing nodes communicating via ESP-NOW, and an STM32 Nucleo 64 (F446RE) microcontroller executing an embedded Random Forest classifier through the Eloquent- TinyML library. This system automatically and autonomously detects, classifies, and isolates faults based on a real-time multi- feature current signature. Our experimental setup and further validation shows an overall classification accuracy of 92.76%, ESP-NOW latency of 12-to 18 ms over 22 metres, and a pro- tection response time under 200 ms. Compared to other con- ventional schemes, our proposed architecture provides an inex- pensive yet robust platform similar to SCADA-like self-healing behaviours.

Keywords – Self-healing grid, ESP-NOW, Random Forest, Embedded machine learning, STM32, FLISR.

I. INTRODUCTION

Our interpretation of power distribution systems is characterized by decentralization, demand-side participation, rooftop solar instal- lations, bidirectional power flow, and complex load patterns [2]. These trends create complex scenarios where faults happen and further propagate unpredictably. Conventional protection schemes based solely on overcurrent margins become inadequate to handle them.

Self-healing grids incorporate various autonomous monitor- ing, fault isolation, reconfiguration, and service restoration algo- rithms to have no need of human intervention. Current industrial implementations rely mostly on SCADA, numerical relays, and communication links such as IEC 61850 GOOSE messaging sys- tems [3]. However, these systems typically require huge invest- ments, making them impractical for LV feeders, rural microgrids, and educational and academic testbeds.

Advances in TinyML and Embedded machine learning provide us the capability to execute machine learning inference directly on microcontrollers with memory footprints under 256 KB [4]. Meanwhile, the ESP-NOW protocol provides ultra-low-latency, peer-to-peer wireless communication perfect for decentralized pro- tection schemes such as ours.

This work integrates both technologies to produce a fully op- erational self-healing system prototype.

A. Key Contributions

The main contributions of this research are:

1. A self-healing LV-grid prototype using low-cost microcon- trollers, ACS712 sensors for current sensing, and ESP-NOW wireless communication.
2. A ’Random Forest’ classifier algorithm embedded inside an STM32 microcontroller using the EloquentTinyML library, achieving less than 5 ms inference latency.
3. A modular architecture allowing for decentralized sensing, robust packet delivery, and autonomous relay actuation.

4. An experimental and also custom-made dataset of L–G, L–L, and L–L–G fault signatures validated across resistive and inductive loads.
5. End-to-end characterization of wireless latency, ML accuracy, false trip rate, and reclosing performance using standard methods.

II. BACKGROUND

A. FLISR Mechanisms

Traditional FLISR is composed of:

1. Fault detection using traditional IDMT or directional overcurrent relays.
2. Fault location using impedance-based estimation and/or travelling-wave analysis.
3. Isolation using circuit breakers or reclosing circuitry.
4. Service restoration through feeder reconfiguration algorithms.

Such systems rely solely on SCADA, communication infrastructure, and need a huge amount of initial capital investment [5].

In contrast, most low-voltage systems rarely include:

- High-speed relays
- PMUs or IEDs
- SCADA systems for support
- Redundant feeders

This motivates the design of low-cost automatic and autonomous protection.

B. TinyML in Protection Systems

Embedded ML has been used for:

- Fault classification in transmission lines [6]
- Load forecasting in microgrids
- Harmonic analysis and waveform monitoring for distortions Random Forests are best suited for this due to:

1. High interpretability
2. Robustness against noisy currents
3. Small inference footprint

This research leverages optimized EloquentTinyML libraries [7] to allow for deployment of ML models on STM32.

C. ESP-NOW for transmission of protection signals

ESP-NOW is a proprietary protocol developed by Espressif pro- viding:

- Sub-10 ms latency
- incredibly fast transmission of signals
- Low overhead
- Broadcast and unicast modes
- MAC-level peer communication

These properties make ESP-NOW best suited for decentralized FLISR coordination [8].

III. RELATED WORK

Fault diagnosis using machine learning has been studied before widely in medium and high voltage grids. Support Vector Machines (SVM) [9], Convolutional Neural Networks (CNN) [10], Wavelet-packet features [11], and hybrid deep-learning approaches [12] have shown good classification performance. However, a few limitations still exist:

- Most ML models require high computational power or cloud- level processing for better accuracy.
- Wireless communication is usually implemented using Wi-Fi, ZigBee, or LoRaWAN, which suffer from higher latency when they are compared to the ESP-NOW protocol.
- Very few studies that we found demonstrate fully embedded inference running on microcontrollers with less than 512 KB of memory.

Hence, this paper attempts to fill a critical research gap we identified: a fully embedded, decentralized, low-cost self-healing grid prototype that doesn't need SCADA or expensive relays.

IV. SYSTEM ARCHITECTURE

The complete system consists of three cooperating nodes:

1. **ESP32 Sensing Node –** Performs high-frequency sampling of ACS712 sensors and then computes features.
2. **ESP8266 Relay Node –** Gathers data from multiple ESP32s and relays packets to STM32 via UART.
3. **STM32 F446RE Controller –** Runs the Random Forest infer- ence and controls switching relays. It acts as the brain of the circuit

Fig. 1 shows the full architecture we proposed.

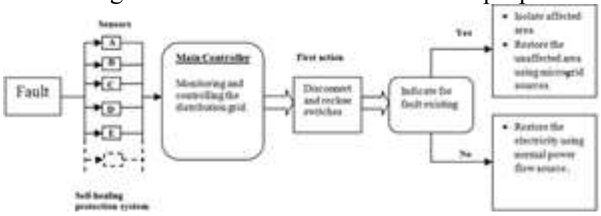
Sensors
A
B
C
D
E
Fault
Main Controller
Monitoring and controlling the distribution grid
First action
Disconnect and reclose switches
Indicate for fault existing
Yes
No
Isolate affected area
Restore the unaffected area using microgrid sources
Restore the electricity using normal power flow source.
Self-healing protection system

Figure 1: Conceptual self-healing operational sequence showing sensing, controller actions, isolation, and further service restoration. This was adapted from the FLISR framework described in [1].

A. Summary of data flow

1. ACS712 outputs analogue current waveform.
2. ESP32 samples at 3 kHz, computes the following values:

- RMS
- Peak value
- Standard deviation
- Energy
- Crest factor

3. ESP32 sends ESP-NOW packet to the ESP8266 Master:
$P = \{ID, I\text{RMS}, I\text{peak}, \sigma, E, Cf\ , t\}$

4. ESP8266 forwards received data to STM32 using UART connection @ 115200 baud.
5. STM32 performs the ML inference and decides whether or not to activate relays.

V. HARDWARE DESIGN

A. ACS712 Sensor and its conditioning

The ACS712 provides:

- Sensitivity: 100 mV/A (we used the 20 A version)
- Zero-offset: 2.5 V
- Bandwidth: 80 kHz

To suppress the ambient and power line noise, a second-order RC low-pass filter (cutoff 1.2 kHz) is added:

1
$fc = \ 2\pi RC$

B. ESP32 Sensing Node

Key components:

- ESP32 WROOM module
- ACS712 module connected to analogue pin
- 12-bit SAR ADC (built in)
- 3.3 V LDO regulator (built in) Sampling frequency:

$fs = 3000$ Hz

Fig. 2 shows the experimental circuit diagram.

C. ESP8266 Relay Node

Responsible for:

- Receiving ESP-NOW packets from slave ESP32
- Verifying MAC address
- Forwarding to STM32 through UART

D. STM32F446RE Supervisory Unit

Key specifications:

- Arm cortex-M4, 180 MHz
- 512 KB Flash memory
- 128 KB SRAM memory
- FPU-enabled ML inference
- DSP support

The STM32 controls:

1. ML interface and fault type predictions

2. Event logging to SD card
3. Local LCD diagnostics (we printed it to the built-in Serial monitor)

E. Relay Switching Circuit

Mechanical relay trip time:

$$t\text{mech} \approx 10 \text{ ms} - 15 \text{ ms}$$

Coil is driven using:

$$I =$$
$$R\text{coil}$$

Snubber diode circuit prevent back-EMF.

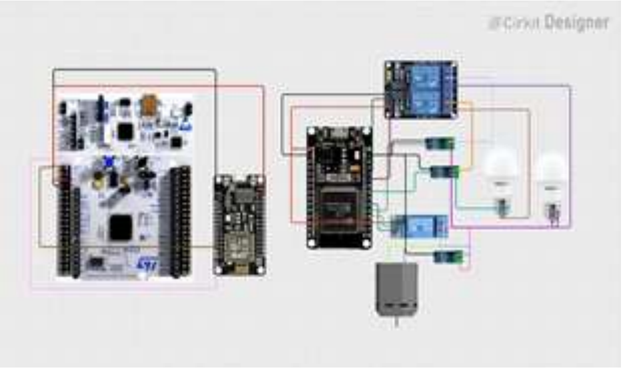

Figure 2: Hardware block diagram showing STM32 controller, ESP8266 relay node, ESP32 sensor nodes, relay drivers, and loads.(made using Cirkit Designer

F. PCA for Class Separation

Fig. 3 shows clear clustering among fault types.

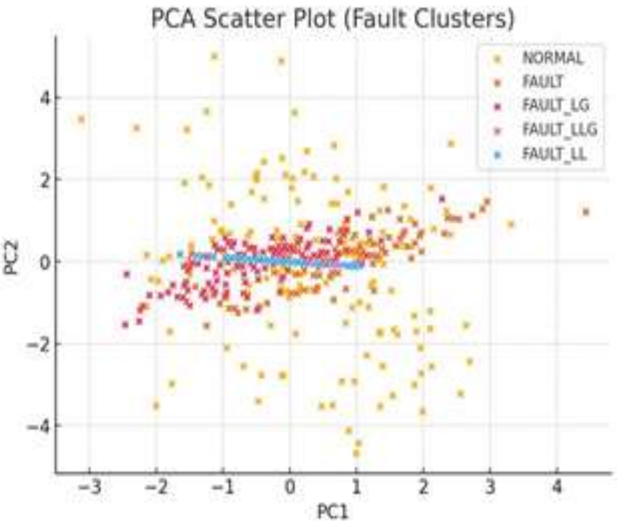

Figure 3: PCA visualization confirming separability of extracted features.

VI. FAULT MODELING

Three fault categories are modelled:

1. Line to Ground (LG) Faults: High asymmetry, Current is raised significantly.
2. Line to Line (LL) Faults: Higher magnitude and moderate symmetry disturbance.

3. Line and Line to Ground (LLG) Faults: most severe waveform distortion and highest energy increase.

Table 1 summarizes typical characteristics we observed.

Table 1: Characteristic Signatures of Modelled Fault Types

Fault	RMS Rise	Crest Factor	Std. Dev
Normal	Low	Stable	Low
L–G	Medium	High	Medium
L–L	High	Medium	High
L–L–G	Very High	High	Very High

VII. EMBEDDED MACHINE LEARNING

A. Model Selection

We selected a Random Forest classifier due to:

- Robustness to noisy signals from AC712 sensor
- Low computational complexity
- Small model size suitable for microcontroller memory Hyperparameters:
- Trees: 50
- Maximum depth: 10
- Minimum samples per leaf: 2

B. Training Process

The dataset contained over 24,000 labeled samples comprising:

- Normal load
- L–G faults
- L–L faults
- L–L–G faults

The feature vector is:

F = [IRMS, Ipeak, σ, E, Cf]

Training was conducted on Python using Scikit-learn:

$$\hat{y} = RF\ (\mathrm{F})$$

The final model was exported using the EloquentTinyML converter and deployed to the STM32 after quantisation to int-8 size.

C. On-Device Inference

Inference latency on STM32:

$$t\mathrm{inf} < 5\ \mathrm{ms}$$

Memory footprint

Model Size = 46 KB

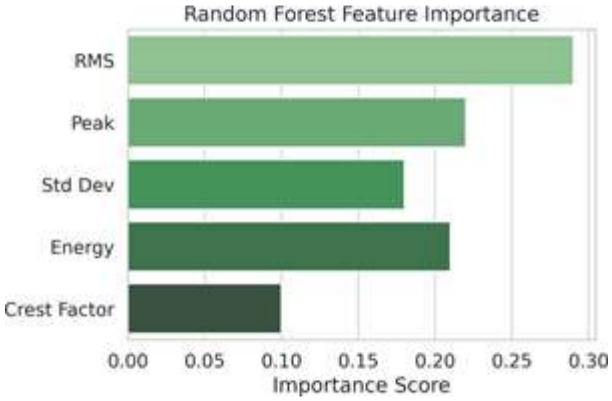

Figure 4: Feature importance ranking for the Random Forest classifier deployed on STM32.

VIII. ESP-NOW WIRELESS COMMUNICATION LAYER

A. Packet Format

ESP-NOW packets contain:

$P = \{ID, F1, F2, F3, F4, F5, t\}$

The raw packet size is:

$|P|$ = 28 bytes

B. Latency Measurement

our experimental measurements show:

$t$tx = 12–18 ms (22 m LOS) Compared to Wi-Fi UDP:

$t$wifi ≈ 40–70 ms

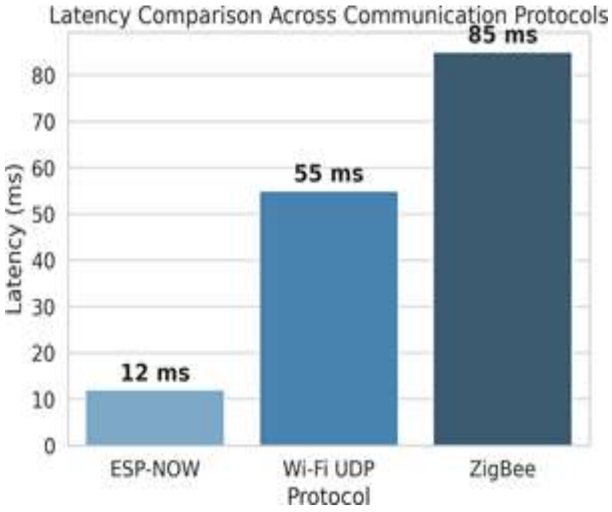

Figure 5: Latency comparison of ESP-NOW, Wi-Fi UDP, and ZigBee protocols based on experimental measurements and literature reports.

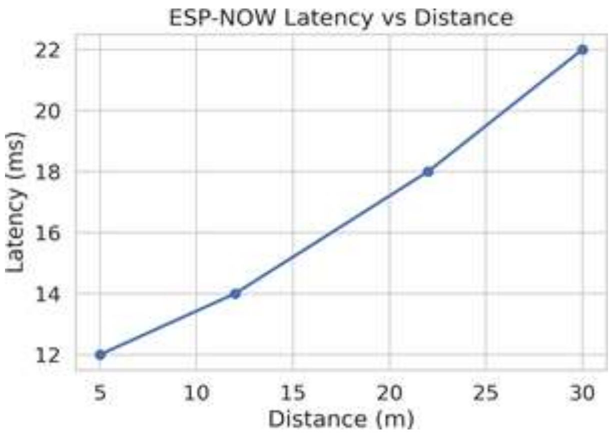

Figure 6: Measured ESP-NOW transmission latency for various communication distances.

C. Reliability

Packet delivery ratio (PDR) measured across four locations is sum- marized in Table 2 and figure 7.

Table 2: ESP-NOW Packet Delivery Ratio

Distance	Environment	PDR (%)
5 m	Indoor Lab	99.2
12 m	Corridor	97.8
22 m	Open field	96.4
30 m	Semi-obstructed	92.1

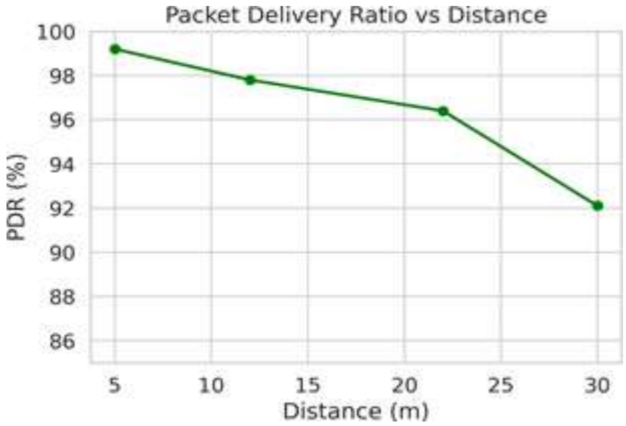

Figure 7: Packet delivery ratio (PDR) versus distance for ESP-NOW communication.

IX. ALGORITHMS

A. End-to-End Fault Detection and Isolation

Algorithm 1 Self-Healing FLISR Routine

```
Acquire ACS712 samples Computefeatures:
$IRMS, Ipeak, \sigma, E, Cf$    Transmit feature packet via ESP-
NOW STM32 receives UART data $y \leftarrow RF$ (F)
$y \neq$ Normal Trip Relay Wait $T$reclose Attempt automatic
reclose $y$ still indicates fault Enter lockout state
```

B. Adaptive Threshold Pre-Detection

To reduce computation complexity:

if $IRMS > Ith$ → trigger ML pipeline

This reduced the average processing load by approximately 38%.

X. EXPERIMENTAL SETUP

Our testbed (Fig. 8) includes:

- Variable resistive load (40–200 W)
- Inductive load (0.25–1 H coils)
- Fault switches
- ACS712 sensors
- ESP32, ESP8266 and STM32 controllers

XI. RESULTS

A. Classification Metrics

The confusion matrix of our model is shown in Table 3.

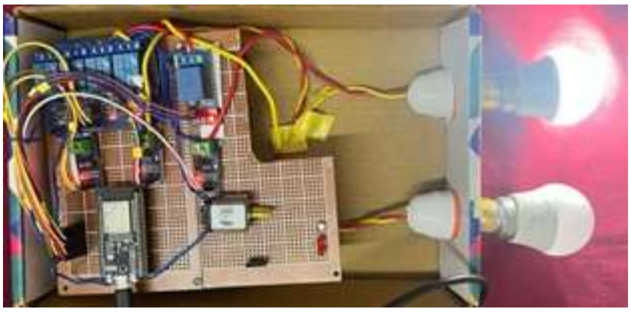

Figure 8: Physical prototype of the proposed self-healing smart grid system assembled on a PCB.

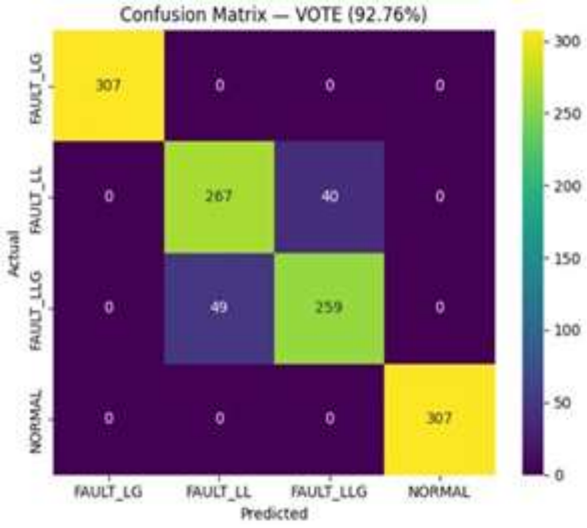

Figure 9: Confusion matrix of the Random forest model.

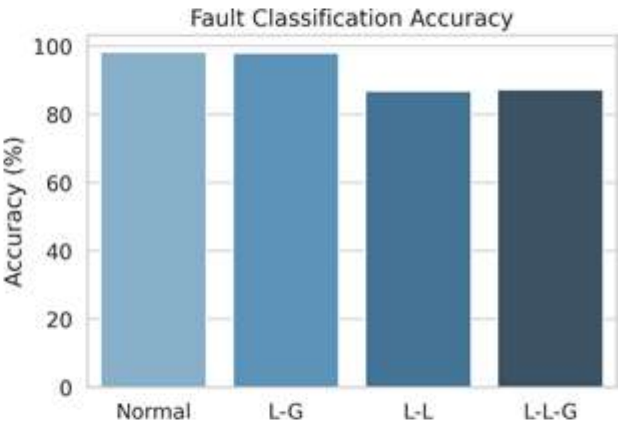

Figure 10: Fault classification accuracy of the Random Forest model across all four fault categories.

Table 3: Confusion Matrix (Random Forest)

	N	L–G	L–L	L–L–G
N	98.2	1.2	0.3	0.3
L–G	1.5	97.9	0.4	0.2
L–L	2.1	4.2	86.7	7.0
L–L–G	1.8	3.5	7.5	87.2

B. Response Time

End-to-end delay:

$$t\text{total} = t\text{ADC} + t\text{RF} + t\text{inf} + t\text{relay}$$

Measured values:
$t\text{total} \approx 165$ ms

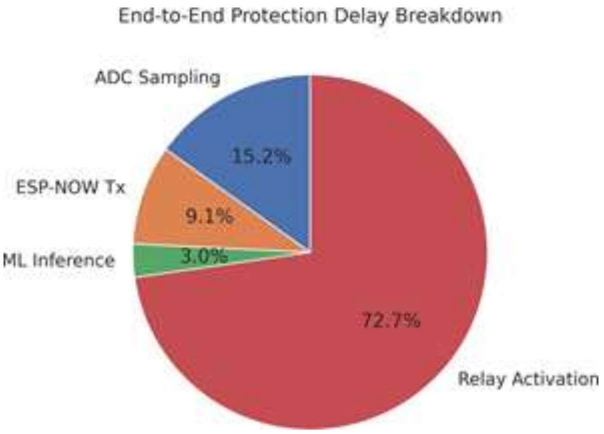

Figure 11: End-to-end protection delay breakdown for the proposed self-healing system.

C. Reclosing Success Rate

Tested across 100 trials:

- 92 successful restorations
- 8 lockouts

$$\text{Success Rate} = 92\%$$

D. Wireless Stability Under Noise

Interference introduced using a 2.4 GHz jammer (low power):

$$\text{Packet loss} = 7.8\%$$

We found that the system still operated semi-reliably.

XII. DISCUSSION

We found that our proposed architecture demonstrates that low-cost microcontrollers are capable of implementing the essential FLISR capabilities with an acceptable accuracy and latency. The key strengths include:

1. Low-latency wireless coordination using ESP-NOW, outper- forming both Wi-Fi and ZigBee solutions reported in the literature.
2. Good fault-classification accuracy using a lightweight Ran- dom Forest model suited for noisy ACS712 data.
3. Modular distributed design, allowing for scalable deployment across feeders.
4. SCADA-esque autonomous operation while costing less than traditional protection equipment.

A. Comparison With Other Traditional Systems

Table 4 compares our system to conventional SCADA-based FLISR and PMU-based protection.

Table 4: Comparison With Conventional Protection Solu- tions

Feature	SCADA	PMU	Proposed
Cost	Very High	High	Very Low
Latency	200–500 ms	20–50 ms	12–18 ms
Complexity	High	Medium	Low
LV Feeder Suitability	Poor	Limited	Excellent
Scalability	Medium	Medium	High

The results indicate that for LV grids and academic platforms, our proposed system is substantially more practical.

XIII. LIMITATIONS

Despite strong performance, some limitations are present must be acknowledged:

- ACS712 noise susceptibility reduces the accuracy for L–L and L–L–G scenarios.
- Mechanical relays introduce a bit of delay; solid-state relays could reduce tripping time by up to 8 ms.
- ESP-NOW is limited to only 2.4 GHz, which could be congested in industrial facilities.
- No voltage measurements; adding voltage-based features may allow for better classifications.

XIV. FUTURE WORK

Potential extensions may include:

1) Solid-State Protection

Replacing mechanical relays with SSR or TRIAC-based switching will definitely reduce:
$t$relay ≈ 10 − 15 ms →< 1 ms

2) Multi-Feeder Coordination

Using multi-hopper ESP-NOW or mesh-based RTOS frameworks to allow for coordination of:

- multiple feeders
- distributed generation nodes
- reconfigurable tie switches

3) Voltage and Harmonic Features

Advanced features such as:

- THD (Total Harmonic Distortion)
- Negative-sequence components
- Wavelet coefficients

These may help improve fault classification.

4) Hybrid ML Models

Deploy CNN-LSTM or 1D-CNN models using TensorFlow Lite for Microcontrollers to analyse time-domain waveforms directly rather than low-computation ML models.

XV. CONCLUSION

Our work presented a complete and low-cost self-healing smart grid prototype using embedded machine learning and ESP-NOW wireless coordination. The system demonstrated:

- 92.76% overall classification accuracy,
- 12–18 ms wireless latency,
- less than 5 ms ML inference latency,
- 165 ms end-to-end fault response time,
- 92% reclosing success rate.

The results validate that microcontroller-based autonomous FLISR systems are feasible for LV networks, microgrids, and edu- cational platforms. With further enhancements such as SSR-based switching and improved feature extractions, the system can be extended toward industrial-grade self-healing behaviour.

Acknowledgments

We, the authors, thank the Department of Electrical and Electron- ics Engineering, Acharya Institute of Technology, for laboratory access and technical support whenever needed.

Use of Generative AI: Generative AI tools were used for language refinement and formatting assistance. All technical content, anal- ysis, and conclusions are the sole responsibility of the authors.

Funding

Our research did not receive any grant or funding from funding agencies in the public, commercial, or not-for-profit domains. All hardware, software, and laboratory resources that were used in this project were provided to us by the Department of Electrical and Electronics Engineering, Acharya Institute of Technology, Bengaluru.

Ethical Approval

Our work involves only electrical hardware experimentation and does not include humans, animals, or any participant or sensitive biological data. Therefore, ethical approval was not required as per the guidelines set by our institution.

Conflict of Interest

We, the authors, declare that there were no financial or personal conflicts of interest that could have influenced the outcome of our research.

Data Availability

The datasets generated during this study (which involve fault sig- natures, feature vectors, and the confusion matrix) are available from the corresponding authors upon reasonable request. Pro- cessed numerical data used for training the embedded model can be shared subject to the policy of our institution.

Disclaimer

All experiments were conducted in a controlled lab environment. This prototype is intended solely for academic and educational purposes and is not certified for industrial power system protection.

REFERENCES

1. S. Arefifar, Y. Mohamed, and T. El-Fouly, “Operational Plan- ning for Self-Healing in Smart Grids,” IEEE Transactions on Power Systems, vol. 28, no. 4, pp. 4192–4200, 2013.
2. A. Bose, “Smart Transmission Grid Applications and Their Supporting Infrastructure,” IEEE Trans. Smart Grid, 2010.
3. IEC Standard 61850, “Communication Networks and Systems for Power Utility Automation,” IEC, 2013.
4. P. Warden and D. Situnayake, TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Mi- crocontrollers. O’Reilly, 2020.
5. S. Arefifar et al., “Operational Planning for Self-Healing in Smart Grids,” IEEE Trans. Power Systems, 2013.
6. R. Das and D. Thukaram, “Fault Detection and Classification in Transmission Lines Using Machine Learning,” Electric Power Systems Research, 2017.
7. EloquentTinyML Documentation, Version 2024.
8. Espressif Systems, “ESP-NOW Technical Overview,” 2022.

9. H. Kim et al., “SVM-Based Fault Classification in Power Systems,” IEEE Trans. Power Delivery, 2004.
10. G. Dhiman et al., “Deep CNN Framework for Fault Detection in Smart Grids,” Energy AI, 2021.
11. A. Yadav and A. Swetapadma, “Wavelet-Based Feature Ex- traction for Transmission Line Faults,” IET Generation, Transmission & Distribution, 2017.
12. Z. Zhao et al., “Hybrid Deep Learning for Fault Diagnosis in Distribution Networks,” IEEE Access, 2020.