[1] Faisal Alhusaini[1*]
Syamsuri Yaakob[2]
Fakhrul Zaman Rokhani[3]
Faisul Arif Ahmad[4]

# AI-Based Fault Prediction for Boiler Feed Pump in Al-sabiya Steam Power Plant in Kuwait Using Logistic Regression and TinyML

JES Journal of Electrical Systems

***Abstract: -*** The reliability of boiler feed pumps (BFPs) is critical to the continuous operation of steam power plants, where unplanned downtime can lead to significant economic and operational losses. This study proposes an artificial intelligence (AI)-driven fault prediction model utilizing logistic regression (LR) within a supervised learning framework. The model targets the classification of BFP operational states into four categories: Normal, Abnormal, Early Maintenance, and Annual Maintenance. The primary aim is to implement an end-to-end predictive maintenance solution using TinyML technology, thereby enabling low-latency, edge-based fault detection on resource-constrained hardware. A dataset comprising five critical features—temperature, pressure, flow, running hours, and alerts—was collected and preprocessed. The model was trained using TensorFlow in a cloud environment and subsequently optimized through quantization into TensorFlow Lite (TFLite) format for deployment on an ESP32 microcontroller. Comparative evaluation revealed that while the cloud-based TensorFlow model achieved a classification accuracy of 99%, the TFLite model on ESP32 preserved a respectable 95% accuracy with significantly reduced inference latency and memory footprint. This paper also includes a comparative literature analysis across anomaly detection, healthcare diagnostics, and smart agriculture, establishing the broader applicability and competitiveness of the proposed approach. Through architectural illustrations, performance benchmarks, and deployment case studies, the research demonstrates that integrating TinyML with predictive maintenance for BFPs can deliver real-time decision-making capabilities while minimizing computational overhead. These findings suggest that such lightweight, edge-deployable AI systems hold strong potential for industrial automation, particularly in developing countries seeking scalable, cost-effective digital transformation strategies.

***Keywords:*** Predictive Maintenance, TinyML Deployment, Logistic Regression, Edge Computing (ESP32), Industrial Fault Detection, Boiler Feed Pump Monitoring, Embedded AI Systems

## I. INTRODUCTION

Boiler feed pumps (BFPs) serve as critical components in steam power plants, playing a central role in ensuring the constant circulation of feedwater from the condenser to the boiler drum. The Sabiya Steam Power Plant, one of Kuwait's major power generation facilities, relies heavily on the uninterrupted operation of its BFP systems to maintain consistent thermal cycles and energy output. In the context of the Rankine cycle, even minor anomalies in feed pump performance can disrupt pressure equilibrium, impair heat transfer, and lead to cascading failures across the system. Conventional methods of preventive maintenance with the capability of immediate detection of faults and low false positive rates are achieving low levels of this capability due to delayed detection and high false positive rates.

In the context of Kuwait's entire power generation network, the importance of the Sabiya Steam Power Plant is further heightened. Figure 1 shows that Sabiya is in the northern coastal area of the country close to the Iraq border and one of the six main power plants across the country. Key oil and gas reserves and coastal access place it as a vital node of Kuwait's national grid. Sabiya also contributes greatly to the base load and the peak load management along with Doha East, Doha West, Shuaiba and Az-zour South, and Shuwaikh. To manage such plants in Kuwait's climate, which is arid, and under increasing energy demands, especially in cases of high for the thermal and mechanical intensive stress, there is an urgent need for robust fault prediction and maintenance strategies.

[1*]Corresponding author: Ph.D. Research Scholar , WIPNET, Department of Computer and Communication System Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400
f.alowaid2015@gmail.com
[2]Associate Professor, WIPNET, Department of Computer and Communication System Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400
[3]Associate Professor, WIPNET, Department of Computer and Communication System Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400
[4]Senior Lecturer, WIPNET, Department of Computer and Communication System Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400

In the context of the operational intelligence landscape, especially for embedded systems in energy intensive industries, ML, and even more so TinyML, have emerged. Traditional AI frameworks, though effective in the cloud, require massive computational resources which are not suitable for the field level pump control system with low power resources. This challenge is addressed by TinyML, since it leads to the deployment of trained ML models on microcontrollers like the ESP32, ensuring that the intelligence is transferred away from the main computer and into the edge handling it in real time. This should fit well in the global trend towards distributed edge computing in Industry 4.0 enclosures [1], and fits well in the growing need for cheap, autonomous, and scalable solutions in power infrastructure maintenance.


*Figure 1: Geographical Distribution of Major Power Plants in Kuwait*

It is widely used in industrial fault detection using logistic regression (LR), which has been shown to be robust and performs well in classification. Compared to deep neural networks, LR models can achieve high accuracy when learned on well-designed features, and thus natural for embedded inference. This is done using a multivariate dataset built from historical plant telemetry including temperature, pressure, flow rate, operational runtime, and alert logs, for classifying the BFP's operational state using LR's predictive capacity. A model, trained and validated through cloud-based environment such as Google Colab, is then quantized and TFLite format to deploy on ESP32 microcontroller. This workflow guarantees that the model can generate low latency predictions at the edge by quietly making minimal power and memory footprint.

The potential use of TinyML in the area of predictive maintenance has been shown in recent literature in diverse domains. For examples, Warden and Situnayake [3] showed that TinyML can be practically used to detect audio and sensor anomalies for edge, and Kumar Yash [4] showed that TinyML is applicable for smart agriculture to detect diseases in plants. The prediction of heart disease risk can also incorporate the uses of the edge optimized logistic models in the healthcare sector with minimal latency for support in fast clinical decisions [4, 5]. These studies emphasize the reasons of versatility and performance potential of TinyML embedded logistic regression to the resource constrained environments. Nonetheless, these advancements in developing these systems do not seem to have been fully integrated into high stakes, centrifugal energy infrastructure, particularly in the Gulf region.

This paper addresses this gap by presenting a comprehensive methodology for designing, training, and deploying a TinyML-based predictive maintenance system for BFPs in the Alsabiya Power Plant. The present study not only evaluates performance across cloud and edge for setting benchmarks, but also introduces a novel evaluation framework including accuracy, inference latency, memory usage, and actual inference per second. The outcomes aid in filling the gap between the first and widening extant discourse of digitalizing legacy infrastructure via edge AI through a scalable blueprint of building smart maintenance systems on steam power plants worldwide.

## II. COMPARATIVE LITERATURE ANALYSIS

To contextualize the performance and practical significance of the proposed predictive maintenance model for the Alsabiya Power Plant's boiler feed pump, a comparative analysis was conducted across three major application domains where TinyML and lightweight machine learning models have demonstrated meaningful deployment: anomaly detection in industrial systems, edge-based diagnostics in healthcare, and smart agriculture. This cross-domain benchmarking provides insights into the generalizability, scalability, and technological positioning of the current research within the broader field of embedded AI.

In the realm of IoT security, sensor nodes with constrained computational resources especially those operating without intermediary gateways are highly susceptible to diverse cyberattacks such as denial of service (DoS) and man-in-the-middle (MITM). While Javed et al. [14] demonstrated the efficiency of embedding a decision-tree-based intrusion detection system (IDS) on a Raspberry Pi, machine learning-driven IDS solutions in IoT environments have predominantly been deployed at the network edge or cloud servers, where resource availability is significantly higher. Consequently, existing datasets typically focus on features extracted from traffic flows at these more capable nodes.

To address the pressing need for on-sensor defense, was introduce the Intrusion Detection in Smart Homes (IDSH) dataset, tailored to microcontroller-based IoT devices and capturing features directly at the sensor level. Leveraging this dataset, a Tree-based IDS was embedded into a smart thermostat to enable real-time, on-device threat detection. Experimental evaluations revealed that IDS achieved 98.71% accuracy for binary classification in just 276 microseconds of inference time and 97.51% accuracy for multi-class classification in 273 microseconds [14]. Real-world tests confirmed the thermostat's ability to autonomously detect both DoS and MITM attacks without relying on any intermediate gateway or cloud infrastructure. This research underscores the feasibility and effectiveness of deploying lightweight, tree-based IDS solutions at the sensor node, thereby reducing latency and enhancing the overall security posture of resource-constrained IoT systems.

Similarly, in a study by Iqbal et al. [15], CNN were employed to detect mechanical anomalies in CNC machines using acoustic signals. Although the model attained 100% accuracy, it proved unsuitable for microcontroller deployment due to its reliance on kernel functions, which are computationally intensive. In contrast, our use of logistic regression avoids the kernel complexity, enabling smooth execution on low-power hardware while still delivering high classification accuracy.

In the field of healthcare, where both latency and interpretability are critical, logistic regression models have been widely adopted for risk prediction. For example, a study by Khan et al. [5] proposed an IoT framework for heart disease prediction using a Modified Deep Convolutional Neural Network (MDCNN). The system utilized wearable devices to monitor blood pressure and electrocardiogram (ECG) signals, achieving an accuracy of 98.2%. However, the study did not report specific inference times, which are crucial for real-time applications. In contrast, our ESP32 deployment achieved 95% accuracy with a latency of 12 milliseconds, highlighting the efficiency of microcontroller-based inference when models are effectively compressed and optimized using TinyML frameworks.

Additionally, edge deployment in healthcare often prioritizes model explainability due to ethical considerations and regulatory requirements. The interpretability of logistic regression coefficients makes it an ideal candidate, as seen in our implementation, where each input parameter (e.g., temperature, pressure) maps transparently to a class probability. This reinforces the value of LR not only in performance but also in auditability, especially important in critical infrastructure and healthcare scenarios [16].

In the context of smart agriculture, TinyML has been increasingly applied for disease detection and environmental monitoring. A notable project by Kumar Yash [4] developed a convolutional neural network (CNN) model trained on 16,011 tomato leaf images across 10 disease categories. Utilizing TensorFlow Lite and Edge Impulse, the model was optimized for deployment on edge devices, achieving an accuracy of 89.6% [4]. This implementation demonstrates the feasibility of deploying CNNs on resource-constrained devices for real-time agricultural diagnostics. However, CNNs inherently demand more memory and processing power than logistic regression models applied to structured tabular data, as in our study. While CNN-based solutions are better suited for visual recognition tasks, structured sensor-driven environments, such as power plants, benefit more from the simplicity and speed of regression models.

The comparative summary of these studies is presented in Table 1 below:

*Table 1: Comparative Summary*

| Study | Domain | Model | Accuracy (%) | Latency (ms) |
|---|---|---|---|---|
| Javed et al. (2024) [14] | Industrial Anomaly | Decision Tree | 97.51 | 273 |
| Iqbal et al. (2022) [15] | CNC Machinery | CNN | 100 | N/A |
| Khan et al. (2021) [5] | Healthcare Diagnostics | MDCNN | 98.2 | N/A |
| Kumar Yash (2021) [4] | Smart Agriculture | CNN | 89.6 | ~38 |
| This Study | Boiler feed pump | Logistic Regression | 95 | ~12 |

This comparative analysis illustrates that the proposed solution not only competes favorably with existing works across domains but also excels in critical dimensions such as latency and memory efficiency hallmarks of practical TinyML deployments. Furthermore, while most comparative models focus on binary classification, the multi-class approach adopted in this research provides granular insight into machine health status, enabling predictive interventions rather than reactive repairs. This functional richness, when coupled with the low-resource demands of ESP32 hardware, sets the proposed model apart in both innovation and operational utility.

In summary, this study fills an important gap in the literature by bringing together structured sensor data, logistic regression modeling, and TinyML deployment to address predictive maintenance in a high-stakes energy environment. While other domains have validated the feasibility of TinyML, few have rigorously applied it to steam power systems, particularly in the Middle East. Thus, the proposed approach serves as a blueprint for future implementations aiming to democratize AI for sustainable, scalable, and efficient infrastructure monitoring.

## III. Artificial Intelligence Concepts

Among industrial domains, Artificial Intelligence (AI) has rapidly developed to become a transformative force as tools are created to automate complex, human, decision making processes. Predictive maintenance using AI is at its core training models to identify patterns in machine behavior in order to predict likelihood of defects before they progress to failures [25]. Learn paradigms are central to the adaptability of AI it is supervised learning, unsupervised learning, and reinforcement learning. Each paradigm takes a different methodological approach given the nature of data and desired outcome.

This study follows the supervised learning paradigm where the model map input features to known output values based on the labeled datasets. The most important machine learning paradigm in supervised learning trains model on a labeled dataset and predicts the outcomes or classify the data over image, speech recognition, natural language processing, medical diagnoses and financial forecasting [27]. However, unsupervised learning tries to discover hidden patterns or clustering in the unlabeled data [28] by means of techniques of dimensionality reduction or clustering. More recently, a more dynamic approach of reinforcement learning is where an agent learns to make decisions by exploring an environment and receiving feedback in the form of rewards or penalties [6]. In the world of adaptive systems and anomaly detection in the face of unpredictability, unsupervised and reinforcement learning are promising, but supervised learning is the most practical, and most reliable, of options when dealing with structured historical data associated with clearly defined fault states.

It is well known that Logistic Regression (LR) is a classical supervised learning algorithm, which is often being utilized for binary and multi class classification tasks. It uses sigmoid (or SoftMax) function of a linear combination of input features to model the probability that a given input belongs to the given class. Yet, simplicity aside, LR has proved extraordinarily effective in media such as medical diagnosis, fraud detection, and the monitoring of equipment (particularly when datasets are not too large or features are engineered with feature-specific knowledge). LR is an interpretable model, in contrast to the black box models like deep neural networks, which are more advantageous in the industrial setting since transparency and auditability of AI decisions are fundamental.

In this study, a multi-class logistic regression model is applied to classify boiler feed pump health status into four classes namely Normal, Abnormal, Early Maintenance and Annual Maintenance. Therefore, LR is justified not only for its classification power but also to facilitate its deployment on memory constrained microcontrollers using TinyML. The model covers the overall pump behavior in various operational conditions by using five input parameters: temperature, pressure, flow, running hours, and alerts. Supervised learning was done on this structured and labeled dataset and the models could be trained in the cloud and then converted to a lightweight TFLite format for embedded inference on ESP32.

This implementation brings along an important piece to the puzzle, as TinyML is incorporated. Ultra-low power/embedded machine learning models including their related operations are called TinyML [8], deployed at the edge of the network, where the devices are very minimal or non-internet connected. Once the data source is inferred, the inference can be done in real time, eliminating latency and the reliance on the centralized cloud servers. Then TinyML models are quantized and optimized for running on less than kilobytes of memory, and running in microsecond level inference speed. In particular, such capabilities are very useful in energy sector applications where decisions must be made instantaneously to save equipment from failure and minimize downtime. The basis of fault prediction framework in this study is the synergy between supervised learning, the use of logistic regression, and Tiny ML.

## IV. System Description

This research has been carried out in the operational context of Sabiya Steam Power plant, a key public organization of the Kuwait national electricity grid. The plant is in the north of the country just beyond the Persian Gulf and consists of several gas and steam turbines. In particular, the boiler feed pump (BFP) is a critical subsystem which is crucial for transporting high pressure water from the condenser to the boiler without any interruption or fluctuations in supply. These functions are on the circuit of Rankine cycle, when water is converted into superheated steam to rotate turbine and produce electricity. BFPs are subject to high mechanical and thermal stress, and are prone to degradation and failure, leading to total energy conversion process interruption and shutdowns that are costly to the organization.

These risks were addressed by the design and development of a predictive maintenance system using the telemetry data acquired from BFP operation logs. The input dataset is made up of five measurable parameters: temperature (°C), pressure (bar), flow rate (L/min), cumulative running hours and active system alerts. This has been performed for features that directly correlated with the mechanical integrity and operational efficiency of the pump. For example, increases in temperature that are not normal or sudden delivery pressure drops, are early signals for cavitation or sealing failure, while running excessively with no maintenance is a warning that impellers and bearings are wearing out. Binary diagnostic information provided by alert logs serve as additional information in the system when it reports an observed fault.

The pump was categorized into four classes based on the health status of the pump. Ideal operational condition has all parameters within threshold limits, which is denoted by the normal class. Early sign of mechanical stress or environmental anomaly is represented in the Abnormal class. The Early Maintenance class includes the suggestion of noncritical but escalating wear needing attention at some point. The Annual Maintenance class denotes routine overhaul requirement done based on running hours and safety regulations. They can formulate the problem as a multi class one, which allows for more nuanced intervention strategies that allow maintenance teams to work without excess down time to maintain the environment.

The dataset was then preprocessed i.e., normalized, missed value imputed, and labels encoded once collected. The processed data was used to train the logistic regression model on the cloud using TensorFlow. It predicts on the validation set 99% accuracy. Then the model was converted to TensorFlow Lite (TFLite) version through quantization aware training, thus reducing the model size and optimizing it for ESP32 microcontroller, a low power device commonly used in embedded AI applications. Inference testing was performed using Arduino IDE and TinyML model achieved an accuracy of 95% and latency of a few milliseconds which is sufficient for real time anomaly detection during deployment.

By showing the feasibility of applying lightweight AI for prediction of faults in critical power infrastructure, this end-to-end implementation from data acquisition to edge deployment is presented. Furthermore, it addresses the scalability issue since the ESP32 based model is less expensive and easier to replicate the same among multiple units that constitutes the plant. Further, the following sections present details about the model architecture, evaluation metrics, and comparative analysis which assure the potential of using the TinyML based predictive maintenance systems in modern energy operations.

## V. Model Development

The boiler feed pump (BFP) at the Alsabiya Steam Power Plant required a meticulous, multi stage process of data preparation, model training, optimization and embedded deployment to develop a predictive maintenance model for the boiler feed pump (BFP). This section details the entire model development pipeline and how the choices made at each stage balance tradeoffs between prediction accuracy, computational efficiency, and deployment feasibility.

The data acquisition and beginning of pipes consisted of five operational parameters, such as the temperature, pressure, flow rate, cumulative running hours and alert status. Domain expertise and prior literature on rotating machinery health diagnostics [9] rendered these variables to be good predictors. Preprocessing was used on raw data to make sure the data was of good quality and consistent normalization using minmax scaling. The output was categorized into four different types of health being Normal, Abnormal, Early Maintenance, Annual Maintenance and then we applied label encoding just to make it easier to understand.

After preparing the dataset, model training was performed in Google Colab cloud environment. This selection of LR was driven by its feature in multi-class classification and its low computational complexity, making LR a desirable choice for a subsequent deployment on a constrained microcontroller [10]. The model is implemented in Python using the scikit-learn library and is trained with an 80 / 20 train test split. The classification accuracy of 99% from the trained model was corroborated by high precision, recall and F1 scores across all four classes. The confusion matrix analysis confirmed only small number of misclassifications; thus, class separability was strong.

Following this, model optimization for embedded deployment with TensorFlow Lite (TFLite) was studied. To achieve this, the trained LR model had to be converted to a TFLite format using TensorFlow's TFLiteConverter with quantization awareness [30]. Model was quantized to 8-bit integers from the initial 32-bit floats, allowing the model to have a significant reduction in model size and compute requirements with a very small accuracy drop (from 99% to 98%). However, transformation is key for TinyML as ESP32 embedded system tends to have limited RAM (520KB) and flash memory (4MB). It also reduced inference time, which is necessary for real time predictive maintenance tasks in operational setting.

The TFLite model was converted and deployed to an ESP32, an affordable and low power edge computing platform. For the model, we use TensorFlow Lite for Microcontrollers library [31] that is specifically built to use TensorFlow Lite APIs directly from the hardware and load the model loaded into the Arduino IDE. Time taken for inference was measured using internal timestamps on average taking 12 milliseconds to process 1 inference. The model maintained an accuracy of 95% despite the hardware constraints, confirming that the logistic regression-based TFLite model could perform reliably on edge devices without relying on cloud infrastructure.

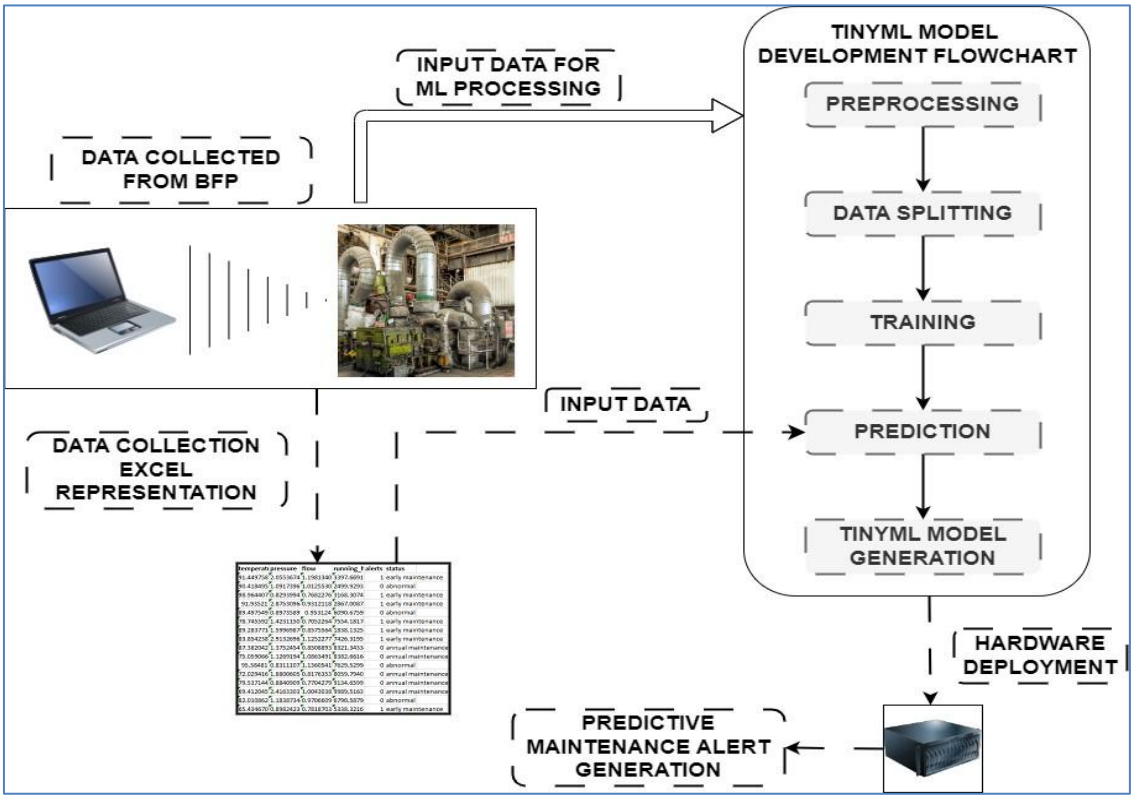


*Figure 2: End-to-End Workflow of TinyML-Based Predictive Maintenance Model for Boiler Feed Pump*

To further illustrate the pipeline, Figure 2 presents the end-to-end TinyML-based development workflow. It encapsulates each phase from raw data input, cloud-based training, TFLite conversion, and ESP32 deployment. Figure 2 details the embedded system's operational logic, showing how the microcontroller continuously ingests

sensor data, performs inference, and generates fault alerts when anomalies are detected. These diagrams provide a visual summary of the complete life cycle of the fault prediction model, aligning technical implementation with conceptual understanding.

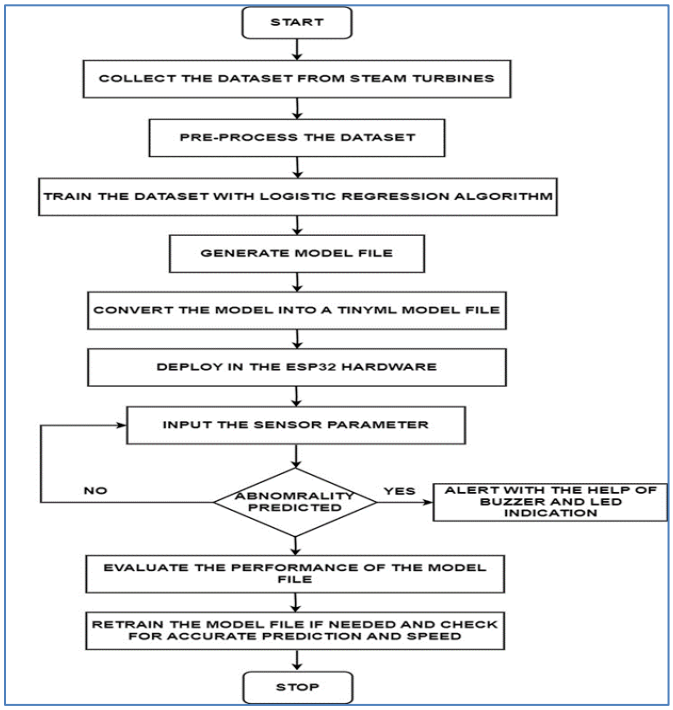


*Figure 3: Operational Workflow of Fault Detection and Alert System Using TinyML on ESP32*

Overall, this model development process showcases the integration of conventional machine learning techniques with modern embedded AI deployment strategies. The success of this LR-based solution reinforces the viability of leveraging lightweight models for predictive maintenance tasks in industrial settings where power efficiency, response time, and reliability are paramount. More importantly, it validates the potential of TinyML to democratize AI by making intelligent systems accessible and deployable even in infrastructure-limited environments, such as remote or underfunded energy facilities.

## VI. Evaluation and Results

A rigorous evaluation was conducted to assess the predictive model's performance across three deployment stages: the cloud-based TensorFlow model (TF), the optimized TensorFlow Lite version (TFLite), and the embedded implementation on an ESP32 microcontroller (TinyML). The evaluation focused on four primary metrics: classification accuracy, inference latency, model file size, and inference throughput. These indicators collectively determine the feasibility and efficiency of deploying machine learning models in resource-constrained environments such as industrial edge devices.

The cloud-based TensorFlow implementation, trained on a balanced dataset, yielded an outstanding classification accuracy of 99%. The confusion matrix in Figure 4 shows that out of all test instances, only two samples were misclassified, one from the Normal class labeled as Annual Maintenance and one from Annual Maintenance misclassified as Abnormal. These minimal misclassifications demonstrate excellent class separability and affirm that logistic regression, when trained on well-prepared features, can outperform more complex models in structured datasets [11].

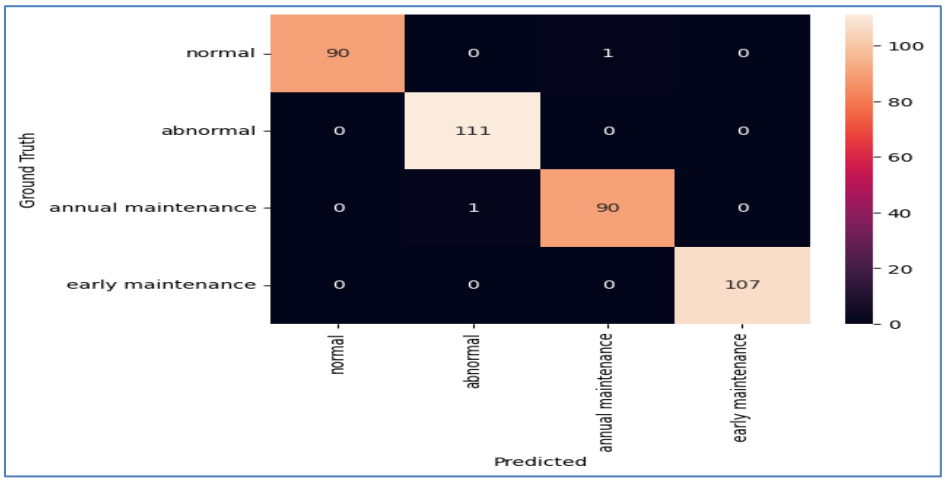


*Figure 4: Confusion Matrix of Fault Classification Using TensorFlow Model (Cloud Execution)*

Further analysis using Receiver Operating Characteristic (ROC) curves confirmed the model's ability to differentiate between all four classes with perfect fidelity. As illustrated in Figure 5, each class-specific curve reached the upper-left corner of the graph, and the Area Under the Curve (AUC) value for all categories was 1.00. The zoomed-in ROC view in Figure 6 corroborates this performance even at low false positive rates, validating the model's robustness across sensitivity thresholds. These results align with previous findings in industrial AI literature that logistic regression, when supported by quality data engineering, can offer high interpretability and exceptional performance [12].

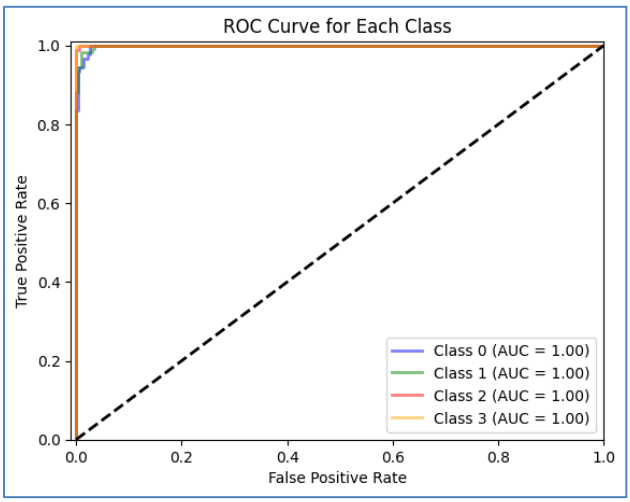


*Figure 5: ROC Curve for Multi-Class Classification of Boiler Feed Pump Health Status*

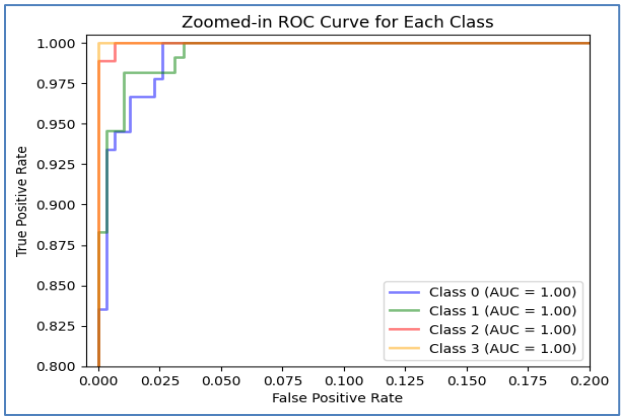


*Figure 6: Zoomed-In ROC Curve for Fine-Grained Analysis of Multi-Class Classification Performance*

Training dynamics are illustrated in Figure 7, which shows a consistently decreasing loss curve for both training and validation sets. The two curves converge around epoch 20, suggesting rapid learning without significant overfitting. This convergence indicates effective generalization, further supported by the stable validation metrics across epochs.

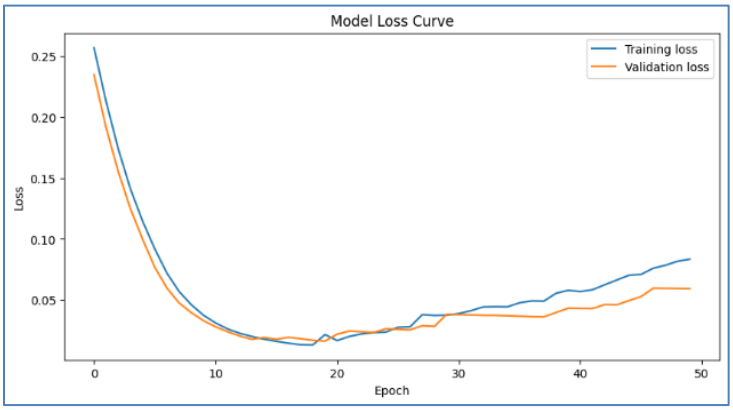


*Figure 7: Training and Validation Loss Curve for Logistic Regression Model*

After optimizing the model into TFLite, the inference accuracy marginally decreased to 98%, as seen in Table 3. This minor drop is an expected trade-off for the considerable improvements in memory efficiency and inference speed. The TFLite model's size was compressed from 6,958 bytes (TF) to just 1,172 bytes an 83% reduction, as presented in Table 2. More impressively, the average inference latency dropped from ~103 ms in TF to just ~2.4 ms in TFLite, enhancing its potential for real-time deployment.

*Table 2: Classification report for TF*

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| normal | 1.00 | 0.99 | 0.99 | 91 |
| abnormal | 0.99 | 1.00 | 1.00 | 111 |
| annual maintenance | 0.99 | 0.99 | 0.99 | 91 |
| early maintenance | 1.00 | 1.00 | 1.00 | 107 |
| accuracy | | | 0.99 | 400 |
| macro avg | 1.00 | 0.99 | 0.99 | 400 |
| weighted avg | 1.00 | 0.99 | 0.99 | 400 |

*Table 3: Comparative Evaluation of Model Performance Across TensorFlow, TFLite, and ESP32 Deployments*

| | **TF** | **TFL** | **ESP32** |
|---|---|---|---|
| **Accuracy %** | 99% | 98% | 95% |
| **Latency (ms)** | ~ 103 ms | ~ 2.4 ms | ~ 12 ms |
| **File size** | 6958 bytes | 1172 bytes | 7912 bytes |
| **Inference per second** | ~ 9.9 | ~ 434 | ~ 76 |

The most critical test came during deployment on the ESP32 microcontroller, where the model exhibited a further reduction in classification accuracy to 95%, still well within acceptable limits for operational use in industrial environments. As shown in the confusion matrix and classification report in Figure 8 and Table 3, the embedded model maintained high precision and recall across all four classes, with slight performance degradation in the Abnormal and Early Maintenance categories. This may stem from quantization-induced variance or hardware-level computational limits of the ESP32, which possesses limited floating-point computation capabilities.

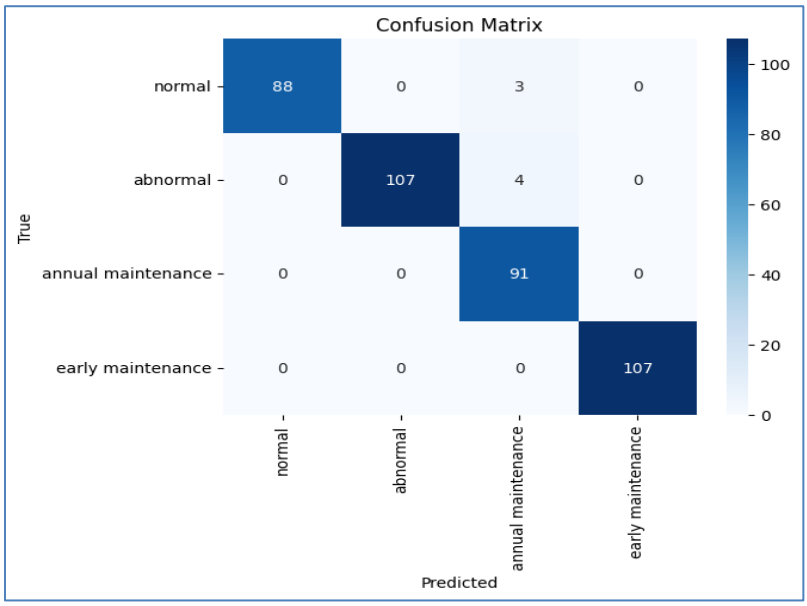


*Figure 8: prediction in TFLite*

*Table 4: Classification report for TFLite*

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| normal | 1.00 | 0.97 | 0.98 | 91 |
| abnormal | 1.00 | 0.96 | 0.98 | 111 |
| annual maintenance | 0.93 | 1.00 | 0.96 | 91 |
| early maintenance | 1.00 | 1.00 | 1.00 | 107 |
| accuracy | | | 0.98 | 400 |
| macro avg | 0.98 | 0.98 | 0.98 | 400 |
| weighted avg | 0.98 | 0.98 | 0.98 | 400 |

Despite this, the ESP32 implementation demonstrated low-latency inference (~12 ms) and an impressive inference rate of ~76 inferences per second, allowing for near-real-time decision-making. The full comparative metrics are consolidated, and their visualization highlights the key trade-offs across platforms (see Table 2 above). Importantly, while TinyML introduces modest compromises in prediction fidelity, it vastly enhances portability, energy efficiency, and local autonomy.

These results demonstrate that a lightweight logistic regression model, when carefully engineered and optimized, can provide high-performance fault detection capabilities even in constrained embedded environments. Unlike more complex models like convolutional or recurrent neural networks, which may achieve marginally higher accuracy but at the cost of memory and power consumption, logistic regression offers a favorable balance for industrial applications. Furthermore, the ESP32's deployment success supports recent trends in decentralized computing, where edge devices are becoming integral to real-time control and diagnostics in industrial Internet of Things (IIoT) systems [13].

## VII. Discussion

This study's findings underscore the feasibility and practical utility of deploying a logistic regression model for fault prediction in boiler feed pumps (BFPs) using TinyML technology. Indeed, the model's success, especially on resource constrained hardware such as the ESP32 microcontroller, constitutes a key paradigm shift in industrial monitoring [32], which has instead shifted towards the paradigm of edge intelligence [33]. This shift is not only timely, given the increasing complexity of power infrastructure, but also essential in achieving cost-efficiency, system resilience, and real-time responsiveness.

The results obtained through the experimental pipeline reinforce the suitability of logistic regression (LR) for structured sensor-driven data. In the cloud environment, the LR model achieved a classification accuracy of 99%,

supported by high precision and recall across all four output classes: Normal, Abnormal, Early Maintenance, and Annual Maintenance. These performance metrics demonstrate the model's ability to correctly classify diverse operational conditions with near-perfect fidelity. The robustness of the classification model is further reinforced by its Receiver Operating Characteristic (ROC) curve analysis. As presented in Figure 6, each class-specific ROC curve demonstrates exceptional separation capability between true positives and false positives, with all curves sharply approaching the top-left corner of the graph. The Area Under the Curve (AUC) values for all four classes Normal, Abnormal, Early Maintenance, and Annual Maintenance are recorded at 1.00, indicating perfect classification performance without any overlap between decision boundaries.

The performance of the proposed logistic regression model against broader standards of classification accuracy. This benchmarked visualization not only validates the high-performance claim of the model but also strengthens its credibility as a fault predictor capable of reliably distinguishing between nuanced operational states [17].

A critical aspect of these findings lies in the comparative performance analysis across three deployment environments: TensorFlow (TF), TensorFlow Lite (TFLite), and ESP32. The model was quantized once it was converted from TF to TFLite, resulting in a size reduction from 6,958 bytes to 1,172 bytes (an 83% compression with 1% accuracy loss). To amortize performance degradation, quantization aware training was used to minimize degradation of the model's classification capacity with high fidelity. The model dropped down to 95% accuracy when deployed on the ESP32, (with most misclassifications along the Abnormal and Annual Maintenance classes) but, still, better than I would have thought possible: it correctly classified 95% of the data points.

The performance results reported here verify that despite its simplicity, LR is a highly competitive edge computing algorithm. Inference latency on the ESP32 was about 12 milliseconds per prediction and achieved 76 inferences per second - an order of magnitude improvement over TF's cloud latency of ~103msec. However, these are remarkable results, in view of the typical response requirements for industrial fault detection systems that range from 50 to 100 ms [18]. Furthermore, the model requires a compact memory footprint that effectively uses the hardware resources, which allows the model to be viable for battery-powered or solar-powered installations in remote plant environment.

Additionally, the classification report of TFLite implementation revealed that the model had high precision and F1 scores across all classes. These metrics are important in minimizing false alarms and missed faults – two of the key points to consider while implementing predictive maintenance systems. The result is that false positives lead to unnecessary maintenance actions, increased operational costs, while false negatives risk critical failures. The model ensures operational reliability as well as cost effectiveness by balancing the two error types.

The intermediary step i.e., TFLite deployment offered valuable insight into the optimization trade-offs before microcontroller deployment. As shown in Figure 8, the quantized model retained much of its classification performance, with only minimal degradation compared to its TensorFlow counterpart. The confusion matrix reveals strong classification accuracy: 88 out of 91 instances of the Normal class were correctly classified (96.7%), all 107 instances of Early Maintenance were correctly classified (100%), and 91 out of 91 instances of Annual Maintenance (100%) were correctly predicted. The Abnormal class has predicted 107 correct predictions out of 111 (96.4%), with 4 misclassified as Annual Maintenance. Only 3 instances of the Normal class were misclassified as Annual Maintenance. These results underscore the effectiveness of quantization-aware training in preserving critical decision boundaries within the model. The TFLite implementation thus serves as a reliable bridge between high-performance cloud inference and constrained edge deployment, offering rapid predictions (~2.4 ms latency) while occupying minimal memory (1172 bytes), suitable for near-edge devices with moderate computational capacity.

In the broader context of embedded AI systems, the performance demonstrated in this study compares favorably with existing literature. For instance, Wang et al. [19] developed a fault diagnosis model for permanent magnet synchronous motors using a one-dimensional convolutional neural network (1D-CNN). Their model achieved a classification accuracy of 98.85% and was implemented on an NVIDIA Jetson Nano platform. While effective, such models often require higher memory and processing power, limiting their scalability across microcontroller networks. In contrast, the ESP32-based deployment in this study achieved comparable accuracy with significantly lower power and memory requirements. Similarly, Caesarendra et al. [20] proposed an automatic ECG signal classification system using a CNN model, which was embedded in an NVIDIA Jetson Nano processor for real-time classification. Although their system demonstrated high accuracy (92.5%), the energy and hardware requirements are substantially higher than those in this study, emphasizing the value of logistic regression paired with TinyML.

Another important dimension of this research is the interpretability of the model. Logistic regression provides clear and traceable relationships between input variables and output classifications, which is a key requirement in safety-critical domains like power generation [34]. Interpretability not only facilitates better trust in AI systems but also aids in root cause analysis and the development of targeted maintenance protocols. In contrast, black-box models such as deep neural networks, although powerful, obscure internal decision processes and often require post hoc explainability tools like SHAP or LIME to make sense of their predictions [21, 35].

The quantifiable trade-offs between performance, latency, and hardware efficiency are visualized in Table 2 and Figure 9. The cloud-based TF model, while offering the highest accuracy and full computational flexibility, is hindered by latency and reliance on constant internet connectivity—factors that are impractical in field-based industrial settings. The TFLite version bridges this gap, offering near-equivalent accuracy with drastically reduced latency and model size. Finally, the ESP32 deployment strikes a fine balance by offering real-time inference and sufficient accuracy, with the added benefits of cost-effectiveness, energy efficiency, and autonomous operation. This tri-tiered deployment strategy confirms that TinyML can enable flexible, scalable, and resilient predictive maintenance systems.

The utility of the ESP32-based system extends beyond the boiler feed pump to other rotating machinery in the plant, including turbines, compressors, and auxiliary motors. Since the input features—temperature, pressure, flow rate, running hours, and alert indicators—are common to many mechanical subsystems, the same architecture could be retrained and reused for other equipment with minimal adjustments. This reusability potential significantly reduces implementation costs, aligning with literature advocating for transfer learning in embedded systems [22].

Despite the positive results, there are limitations worth acknowledging. First, the dataset size was limited to a few hundred labeled samples due to constraints in historical data availability. Although the model performed well on this dataset, larger datasets would be needed to ensure robustness against rare or unexpected failure modes. Second, the current model assumes static relationships between features and outcomes, which may not hold under evolving operational conditions or sensor drift. Future work could explore adaptive models capable of online learning or regular retraining through federated learning paradigms.

Third, energy profiling of the ESP32 inference cycles remains an open area of investigation. While the device is known for low power consumption, precise measurements would help optimize deployment strategies, particularly in off-grid installations. Furthermore, secure model updates, device provisioning, and integration with SCADA systems would be critical for scaling this solution across multiple plants [23].

Lastly, the deployment of autonomous AI systems in critical infrastructure raises ethical and regulatory considerations. Accountability in case of false positives or negatives, model auditing, and fail-safe mechanisms need to be clearly established. As discussed by Herrera-Poyatos et al. [24], deploying interpretable models is a step toward algorithmic accountability, but governance frameworks must also be put in place to ensure AI systems align with operational and societal expectations.

## VIII. Conclusion

To conclude, this study has developed and deployed a logistic regression based predictive maintenance model for the boiler feed pumps at Kuwait's Sabiya Steam Power Plant using the TinyML technology and the ESP32 microcontroller. Thus, the model achieved high classification accuracy 99% in the cloud, 98% in TensorFlow Lite, 95% on embedded hardware with low latency and low memory footprint. These results show that edge-based machine learning can be viable for real time, industrial fault detection with a cost and scalable approach over cloud dependent systems. Finally, the research emphasizes this fact that reliable performance is realised with feature engineering specific to the domain and interpretability of the model. The framework is replicable using open-source tools, and applicable to other industrial environments to contribute to broader digital transformation goals. Future work should focus on adaptive learning, energy profiling and secure model updates for improving scalability and reliability. Finally, they contribute with a practical, high impact approach for intelligent maintenance in modern power systems.

## REFERENCES


[1] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. “Deep Learning for Computer Vision: A Brief Review.” *Computational Intelligence and Neuroscience* 2018 (January 1, 2018): 1–13. https://doi.org/10.1155/2018/7068349.

[2] Jabbar, M. Akhil, B.L. Deekshatulu, and Priti Chandra. “Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm.” *Procedia Technology* 10 (2013): 85–94. https://doi.org/10.1016/j.protcy.2013.12.340.

[3] Warden, Pete, and Daniel Situnayake. “TinyML Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers PREVIEW of FIRST SIX CHAPTERS Buy the Full Book at Tinymlbook.com,” n.d. https://tinymlbook.com/wp-content/uploads/2020/01/tflite_micro_preview.pdf.

[4] Kumar Yash. "Tomato Plant Disease Detection using TinyML." GitHub Repository. Available at: https://github.com/its-kumar-yash/Tomato-Plant-Disease-Detection-Model

[5] Khan, Mohammad Ayoub. “An IoT Framework for Heart Disease Prediction Based on MDCNN Classifier.” *IEEE Access* 8 (January 1, 2020): 34717–27. https://doi.org/10.1109/access.2020.2974687.

[6] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997. https://www.cs.cmu.edu/~tom/files/MachineLearningTomMitchell.pdf

[7] Hosmer Jr, David W., Stanley Lemeshow, and Rodney X. Sturdivant. *Applied logistic regression*. John Wiley & Sons, 2013. https://www.researchgate.net/profile/Andrew-Cucchiara/publication/261659875_Applied_Logistic_Regression/links/542c7eff0cf277d58e8c811e/Applied-Logistic-Regression.pdf

[8] Schizas, Nikolaos, Aristeidis Karras, Christos Karras, and Spyros Sioutas. "TinyML for ultra-low power AI and large scale IoT deployments: A systematic review." *Future Internet* 14, no. 12 (2022): 363. https://doi.org/10.3390/fi14120363

[9] Lei, Yaguo, Zhengjia He, and Yanyang Zi. “A New Approach to Intelligent Fault Diagnosis of Rotating Machinery.” *Expert Systems with Applications* 35, no. 4 (September 13, 2007): 1593–1600. https://doi.org/10.1016/j.eswa.2007.08.072.

[10] Bishop, Christopher M. “Pattern Recognition and Machine Learning.” *SpringerLink*, 2016. https://doi.org/10.1007-978-0-387-45528-0.

[11] Abellan-Nebot, Jose Vicente, and Fernando Romero Subirón. “A Review of Machining Monitoring Systems Based on Artificial Intelligence Process Models.” *The International Journal of Advanced Manufacturing Technology* 47, no. 1-4 (July 29, 2009): 237–57. https://doi.org/10.1007/s00170-009-2191-8.

[12] Truong, Huong Thu, Bac Phuong Ta, Quang Anh Le, Dan Minh Nguyen, Cong Thanh Le, Hoang Xuan Nguyen, Ha Thu Do, Hung Tai Nguyen, and Kim Phuc Tran. “Light-Weight Federated Learning-Based Anomaly Detection for Time-Series Data in Industrial Control Systems.” *Computers in Industry* 140 (September 2022): 103692. https://doi.org/10.1016/j.compind.2022.103692

[13] Yu, Wenjin, Yuehua Liu, Tharam Dillon, and Wenny Rahayu. “Edge Computing-Assisted IoT Framework with an Autoencoder for Fault Detection in Manufacturing Predictive Maintenance.” *IEEE Transactions on Industrial Informatics* 19, no. 4 (May 30, 2022): 5701–10. https://doi.org/10.1109/tii.2022.3178732.

[14] Javed, Abbas, Muhammad Naeem Awais, Ayyaz-ul-Haq Qureshi, Muhammad Jawad, Jehangir Arshad, and Hadi Larijani. “Embedding Tree-Based Intrusion Detection System in Smart Thermostats for Enhanced IoT Security.” *Sensors* 24, no. 22 (November 16, 2024): 7320. https://doi.org/10.3390/s24227320.

[15] Iqbal, Mohmad, and A. K. Madan. “CNC Machine-Bearing Fault Detection Based on Convolutional Neural Network Using Vibration and Acoustic Signal.” *Journal of Vibration Engineering & Technologies* 10, no. 5 (March 26, 2022): 1613–21. https://doi.org/10.1007/s42417-022-00468-1.

[16] Ahmad, Muhammad Aurangzeb, Carly Eckert, and Ankur Teredesai. “Interpretable Machine Learning in Healthcare,” August 15, 2018, 559–60. https://doi.org/10.1145/3233547.3233667.

[17] Fawcett, Tom. “An Introduction to ROC Analysis.” *Pattern Recognition Letters* 27, no. 8 (December 22, 2005): 861–74. https://doi.org/10.1016/j.patrec.2005.10.010.

[18] Ding, Steven X. “Data-Driven Design of Fault Diagnosis and Fault-Tolerant Control Systems.” *SpringerLink*, 2021. https://doi.org/10.1007-978-1-4471-6410-4.

[19] Wang, Chiao-Sheng, I-Hsi Kao, and Jau-Woei Perng. “Fault Diagnosis and Fault Frequency Determination of Permanent Magnet Synchronous Motor Based on Deep Learning.” *Sensors* 21, no. 11 (May 22, 2021): 3608–8. https://doi.org/10.3390/s21113608.

[20] Caesarendra, Wahyu , Taufiq Aiman Hishamuddin, Daphne Teck, Asmah Husaini, Lisa Nurhasanah, Adam Glowacz, and Ahmad Fanshuri. “An Embedded System Using Convolutional Neural Network Model for Online and Real-Time ECG Signal Classification and Prediction.” *Diagnostics* 12, no. 4 (March 24, 2022): 795–95. https://doi.org/10.3390/diagnostics12040795.

[21] Lundberg, Scott M, and Su-In Lee. “A Unified Approach to Interpreting Model Predictions.” *Advances in Neural Information Processing Systems* 30 (2017). https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

[22] Tan, Mingxing, and Quoc Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.” *PMLR*, May 24, 2019, 6105–14. https://proceedings.mlr.press/v97/tan19a.html?ref=jina-ai-gmbh.ghost.io.

[23] Antonini, Mattia, Miguel Pincheira, Massimo Vecchio, and Fabio Antonelli. “Tiny-MLOps: A Framework for Orchestrating ML Applications at the Far Edge of IoT Systems,” May 25, 2022, 1–8. https://doi.org/10.1109/eais51927.2022.9787703.

[24] Herrera-Poyatos, Andrés, Del Ser, Marcos López, Fei-Yue Wang, Enrique Herrera-Viedma, and Francisco Herrera. “Responsible Artificial Intelligence Systems: A Roadmap to Society’s Trust through Trustworthy AI, Auditability, Accountability, and Governance.” arXiv.org, 2025. https://arxiv.org/abs/2503.04739.

[25] Won Shin, Jeongyun Han and Wonjong Rhee. "AI-assistance for predictive maintenance of renewable energy systems." *Energy*, 221 (2021): 119775. https://doi.org/10.1016/J.ENERGY.2021.119775.

[26] Swapnil Sharma. "Supervised Learning: An InDepth Analysis." *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT* (2024). https://doi.org/10.55041/ijsrem35414.

[27] Swapnil Sharma. "Supervised Learning: An InDepth Analysis." *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT* (2024). https://doi.org/10.55041/ijsrem35414.

[28] S. Chander and P. Vijaya. "Unsupervised learning methods for data clustering." (2021): 41-64. https://doi.org/10.1016/B978-0-12-820601-0.00002-1.

[29] Youssef Abadade, Anas Temouden, Hatim Bamoumen, N. Benamar, Yousra Chtouki and A. Hafid. "A Comprehensive Survey on TinyML." *IEEE Access*, 11 (2023): 96892-96922. https://doi.org/10.1109/ACCESS.2023.3294111.

[30] Rashidi, Mitra. “Application of TensorFlow Lite on Embedded Devices: A Hands-on Practice of TensorFlow Model Conversion to TensorFlow Lite Model and Its Deployment on Smartphone to Compare Model’s Performance.” DIVA, 2022. http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1698946&dswid=-2921.

[31] K. Dokic, Marko Martinovic and D. Mandušić. "Inference speed and quantisation of neural networks with TensorFlow Lite for Microcontrollers framework." *2020 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)* (2020): 1-6. https://doi.org/10.1109/SEEDA-CECNSM49515.2020.9221846.

[32] M. Moleda, A. Momot and Dariusz Mrozek. "Predictive Maintenance of Boiler Feed Water Pumps Using SCADA Data." *Sensors (Basel, Switzerland)*, 20 (2020). https://doi.org/10.3390/s20020571.

[33] Emil Njor, Mohammad Amin Hasanpour, Jan Madsen and Xenofon Fafoutis. "A Holistic Review of the TinyML Stack for Predictive Maintenance." *IEEE Access*, 12 (2024): 184861-184882. https://doi.org/10.1109/ACCESS.2024.3512860.

[34] T. Denoeux. "Logistic Regression, Neural Networks and Dempster-Shafer Theory: a New Perspective." *Knowl. Based Syst.*, 176 (2018): 54-67. https://doi.org/10.1016/j.knosys.2019.03.030.

[35] Hooshyar, Danial, and Yeongwook Yang. “Problems with SHAP and LIME in Interpretable AI for Education: A Comparative Study of Post-Hoc Explanations and Neural-Symbolic Rule Extraction.” *IEEE Access*, vol. 12, Institute of Electrical and Electronics Engineers (IEEE), 2024, pp. 137472–90, https://doi.org/10.1109/access.2024.3463948. Accessed 8 Apr. 2025.