

Faisal Alhusaini^{1*}Syamsuri Yaakob²

Fakhrul Zaman

Rokhani³Faisul Arif Ahmad⁴

A Hybrid 1D CNN-GOA Approach for Fault Detection in Feed water Pumps: Case Study of Al-Sabiya Steam Power Plant



Abstract: - Feed water pump fault detection is important to ensure the reliability and efficiency of the steam power plant. Fault detection based on threshold monitoring and rule-based systems cannot handle complex operational anomalies. To enhance fault detection accuracy whilst retaining computational efficiency, this study presents a hybrid 1D Convolutional Neural Network (CNN) and Grasshopper Optimization Algorithm (GOA) approach. GOA optimizes hyperparameters to improve the model's predictive performance in tandem with the 1D CNN to extract temporal features from time-series sensor data. In this methodology, data pre-processing, model training on Google Colab using TensorFlow, then deployment on an ESP32 microcontroller using TensorFlow Lite (TFLite). Accuracy, precision, recall, F1-score, confusion matrices, ROC curves, AUC score and inference latency are used to evaluate model's performance from precision. The results show significant better classification accuracy with 99.5% using the pro-posed hybrid model, compared to traditional machine learning techniques. Addition-ally, 97% accuracy on inference time 9 milliseconds is obtained on ESP32 deployment, making it fit for real time industrial applications. Validation of the efficiency of deep learning and metaheuristic optimization combination in predictive maintenance is shown by the findings. This technique enables online, at the edge fault detection in power plants with reduced downtime and improved operational reliability.

Keywords: Fault Detection; Feed water Pumps; Steam Power Plants; 1D Convolutional Neural Network (1D CNN); Grasshopper Optimization Algorithm (GOA); Predictive Maintenance; TinyML; Embedded Systems; Real-Time Monitoring; Deep Learning; Optimization; Edge AI; Industrial IoT (IIoT)

I. INTRODUCTION

Modern power generation is one of the most important infrastructures in the modern world, the uninterrupted power supply is very necessary for industries, commercial buildings and households. Steam power plants having Rankine cycle are also one of the various power generation technologies. Boiler feedwater pumps are needed to provide a steady supply of water at the pressure and rate needed for steady generation of steam in these plants [1]. Severe consequences from feedwater pump failure are unplanned shutdowns, reduced efficiency, and financial losses. As a result, such a fault detection mechanism should be put in place to ensure the predictive maintenance as well as system longevity.

Changing from the traditional rule-based approach to the machine learning and artificial approach in the fault detection of industrial system. Diagnosis with conventional approaches requires expert knowledge and predefined threshold limits, which are not flexible to (or adaptive in) complex and dynamic operational environments. Deep learning has drawn recent interest for its capability to derive complex information from data and has been successfully used for real time and highly accurate fault detection [2,3]. Specifically, Convolutional Neural Networks (CNNs) have shown great promise in industrial settings. Nevertheless, deploying standard CNN models on em-bedded systems with limited resources is quite challenging because of their high computational cost [4].

This study seeks to overcome the high computational costs and the need to detect faults in one-dimension Convolutional Neural Networks (1D CNN) and the Grass-hopper Optimization Algorithm (GOA). The choice of the 1D CNN is motivated by its opportunity to work well with time series data of handling sensor data from the power plant equipment. GOA integration improves the model's performance by improving the hyperparameters

^{1*}Corresponding author: Ph.D. Research Scholar , WIPNET, Computer and Communication System Engineering Department, Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400 , f.alowaid2015@gmail.com

²Associate Professor, WIPNET, Computer and Communication System Engineering Department, Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400

³Associate Professor, WIPNET, Computer and Communication System Engineering Department, Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400

⁴Senior Lecturer, WIPNET, Computer and Communication System Engineering Department, Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400

and weights, while reducing the overall complexity of model with high detection accuracy [5]. With this hybrid approach, the design is lightweight and according to TinyML frameworks to be deployable in real time to embedded devices like ESP32.

Traditional machine learning models and even deep learning approaches often rely on cloud computing for data processing and inference. While CloudML offers high computational power and accessibility, it introduces latency issues, dependency on an internet connection, and potential cybersecurity vulnerabilities. On the other hand, MobileML provides a balance by executing models on mobile devices with reasonable processing power, but it still consumes significant energy and may not be optimal for industrial applications [6]. TinyML, which focuses on deploying machine learning models on microcontrollers and edge devices, is a more viable alternative for real-time predictive maintenance in industrial settings. TinyML allows models to run locally on embedded systems with minimal power consumption, making it ideal for power plants that require continuous and autonomous monitoring [7].

The primary objective of this research is to develop an optimized fault detection system that enhances the reliability of feedwater pumps in steam power plants. Specifically, this study aims:

RQ1: To investigate the effectiveness of 1D CNNs in analyzing sensor data from power plant equipment.

RQ2: To optimize model parameters using GOA to improve detection efficiency.

RQ3: To deploy the optimized model on ESP32 using TensorFlow Lite to assess its re-al-time performance.

RQ4: To evaluate the model's accuracy, latency, and computational efficiency in a TinyML environment.

Unlike prior studies which often focus on cloud-based fault detection models or rely heavily on computationally expensive solutions unsuitable for embedded deployment, this research distinguishes itself by targeting resource-constrained TinyML platforms. Specifically, it demonstrates the real-time deployment of an optimized 1D CNN-GOA model on an ESP32 microcontroller — a rarely explored combination in industrial predictive maintenance, especially for feedwater pumps. While the monitoring of feedwater pumps might initially appear as a relatively simple application, it presents significant technical challenges when constrained by the limited memory, computational power, and real-time requirements of microcontroller-based systems. Thus, the hybrid 1D CNN-GOA approach proposed in this paper represents a critical innovation, offering a scalable, lightweight, and highly accurate solution for predictive maintenance. This novel integration not only addresses a crucial gap in the literature but also opens pathways for real-world deployment of intelligent fault detection systems within the Industrial Internet of Things (IIoT) and edge AI frameworks.

II. RELATED WORKS

Predictive maintenance has steadily become the keystone of industry operations, ensuring the dependability and productivity of critical machinery. Facial recognition is one of the various deep learning techniques that were successfully used for predictive maintenance in which Convolutional Neural Networks (CNNs) are uniquely successful in fault detection and diagnosis. Still, the appropriate CNN architecture needs to be chosen because variations of the CNN's 1D, 2D and 3D CNNs deliver varying degrees of accuracy, efficiency and computational complexity [8]. It also opens up new possibilities for improving performance while also satisfying the requirement of deployment feasibility in resource constrained environment such as embedded systems through the optimization of these networks with nature inspired algorithms, for example the Grasshopper Optimization Algorithm (GOA).

2.1 *Comparison of 1D, 2D, and 3D CNNs for Fault Detection in Industrial Applications*

The CNN architectures are dependent on the input data structure and mechanism of feature extraction. 2D and 3D CNNs are applied to image and volumetric data analysis while 1D CNNs are used for sequential data, e.g., sensor readings. Selection of the fault detection model that is the most efficient is more dependent on this understanding.

2.1.1 1D CNNs for Time-Series Data Processing

In recent years, 1D CNNs have been proven to be an extremely powerful technique used for analyzing time series sensor data in industrial applications. Unlike traditional machine learning techniques that depend on large quantities of feature engineering, 1D CNNs can automatically learn spatial and temporal dependencies in sequential data. In particular, 1D CNNs were shown to be very effective in rotating machinery fault detection in Junior et al. [4] that demonstrated great accuracy and little preprocessing required compared to traditional statistical approaches. However, their study demonstrated that 1D CNNs were capable of distinguishing subtle differences in vibration signals in order to discover early faults on the motors and pumps [4]. In addition, Fukuoka et al. [9] also compared the performance of 1D CNN with Long Short-Term Memory (LSTM) network in predictive maintenance of wind turbines. They found that 1D CNNs are computationally more efficient for real time monitoring on embedded systems and provide faster inference time than LSTM networks, but they are not good at capturing long term dependencies [9]. Due to the fact that machinery in steam power plants tend to employ continuous sensor data for predictive maintenance, solving the problem with 1D CNNs is the optimal way to extract useful features directly on time series without converting to other forms.

2.1.2 2D CNNs for Fault Detection

2D CNNs, on the other hand, are generally built for image processing but can be applied to time series by transformation, for instance through spectrogram conversion. In the study of Nayak et al. [10] the electrical fault detection in industrial grids were studied using 2D CNNs on dataset represented in heatmaps. 2D CNNs were able to achieve high classification accuracy in their research; however, the inclusion of such 2D CNNs introduced additional computational complexity and were memory intensive [10]. 2D CNNs are less efficient for direct sensor based predictive maintenance because of their reliance on image transformations. In a similar manner, the Xu et al. [11] tries to achieve efficient 2D CNNs by applying lightweight architecture, however they still consumed considerably more computational resources than the 1D CNNs [11]. Therefore, for the real time deployment on the embedded environments, the 1D CNNs became the preferred choice, since it is able to realize high accuracy under the intermediate data transformations but avoids the data transformations.

2.1.3. 3D CNNs for Volumetric Data

3D CNNs are capable of generalizing the convolutional operation to 3 dimensions and hence they provide a way to process volumetric data like medical imaging and video streams. As demonstrated by the study of Roy et al. [39], 3D CNNs for hyperspectral image classification offer better feature extraction powers than 2D models. In contrast, 3D CNNs are also shown by their study to have large computational burden whereby, as the model gets more complex, inference times become longer, and memory consumption is increased [39]. The accuracy of 3D CNNs is improved by spatial richness of data; however, whose deployment in industrial fault detection can be practically impossible, without volumetric sensor data. Again, the hardware constraints of embedded systems such as ESP32 do not allow real-time 3D CNN implementation. Based on these factors, 1D CNNs are going to be the best suited method for fault detection in feed water pump given that sequential sensor data will be the major input.

2.2 *Optimization of CNNs Using the Grasshopper Optimization Algorithm (GOA)*

With resource limited environment, deep learning models' efficiency and accuracy are highly dependent on optimization. Inspired by nature, the Grasshopper Optimization Algorithm (GOA) has become a widely popular metaheuristic algorithm which can be used for fine tuning neural network hyperparameters and reduce computational overhead. Whereas Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) need to balance the exploration versus exploitation in practicing optimization, GOA can do it better, and thus better suit deep learning applications.

2.2.1 GOA in Machine Learning and CNN Optimization:

Saremi et al. [37] introduced GOA as a bio-inspired optimization algorithm inspired from grasshopper swarming behavior. It has been widely applied ever since in engineering and artificial intelligence fields, including

neural network weight optimization [37]. The works of Rahmati et al [13] compare GOA with PSO to optimize neural networks for embedded operation which show that GOA provided faster convergence and lower computational cost while assuring high accuracy. To date, the most recent study by Nabavi et al. [14] used GOA to develop CNN architectures for medical image classification. When they fine tune filter size, learning rate, and dropout rate in their research, they found that GOA significantly reduced model training time with little impact to the classification accuracy as well [14]. With these advantages, the integration of GOA with 1D CNNs for predictive maintenance in industrial applications will be a good choice.

2.2.2 GOA for TinyML and Embedded Systems

For this, CNN models on embedded devices like ESP32 involve one of the prime challenges, which is between model intricacy and computational efficiency. It has been shown that GOA can be harnessed to improve neural networks for deployment on TinyML platforms. Alzubaidi et al. [15] used the GOA to prune the deep learning models and show that the algorithm could truncate the number of parameters without sacrificing accuracy much. This suggests that it is possible to use GOA to fine-tune 1D CNN architectures in order to perform real time fault detection on micro controllers without excessive memory consumption [15].

2.3 Previous Studies on 1D CNNs in Industrial Applications

Many industrial settings have successfully deployed 1D CNNs, from predicting maintenance of manufacturing plants to condition monitoring of power stations. That being said, Zhu et al. [16] conducted a study in which they used 1D CNNs to detect faults in hydraulic systems with 99.43% fault classification accuracy. Based on the research, the outcomes demonstrated that 1D CNNs could differentiate among different failure modes from sensor readings, especially in the feedwater pump monitoring [16]. Similarly, research by Kiranyaz et al. [38] employed 1D CNNs to perform immediate vibration analysis in rotating machinery. The detection accuracy and the response time achieved by their CNN model were better than that from conventional signal processing techniques [38]. These findings aid in proving the applicability of the present study in using 1D CNNs to vibration monitoring since the process is comparable to feedwater pump diagnostics.

2.4 Gaps in Existing Literature and the Need for a Hybrid Approach

Although research on CNN architectures and optimization techniques has been extensively explored, a big gap between such studies and application of 1D CNNs with GOA for predictive maintenance in industrial settings remains. While individual studies show the efficacy of these methodologies on their own, their joint potential has not been explored. In addition, although past research has demonstrated how TinyML benefits from embedded deployments, there has been limited study on the optimization of deployment specific CNN models for ESP32 based fault detection [16]. Building upon these gaps, this study develops a hybrid 1D CNN with GOA approach that trades off accuracy while maintaining fair computational efficiency and deployment feasibility [5]. This research attempts to combine the best features extraction of 1D CNN and optimization of GOA to create a highly accurate fault detection system for feedwater pumps in steam power plants.

Overall, the findings will be useful in continuing the effort to develop intelligent maintenance systems, while illustrating how light-weight AI models can be fielded in resource constrained industrial environments. Finally, while CNNs have become the predominate choice for predictive maintenance, the architecture plays a large role in its efficiency and applicability, and 1D CNNs are the optimal choice for processing time series sensor data with GOA integration delivering a novel way of raising predictive accuracy with the same level of computation overhead. Unlike real world and time critical applications, neither of the previous studies incorporate fault detection techniques and hence this study addresses a critical research gap that no prior studies made them combine with these techniques.

III. METHODOLOGY

3.1 Overview of Research Steps and Flowchart

The structure of the methodology follows a set of steps, as seen in the flowchart, in order to facilitate data processing, model training, and real time deployment of the model. First, data is collected from feedwater pumps

based on readings from some of its sensor parameters (pressure, temperature, flow, running hours, and alert signals). The input of the fault detection model are these readings. Preprocessing of the collected data consists of data normalization and label encoding in order to standardize the dataset and ready it for training. Then the preprocessed data is entered into a 1D CNN model trained using the Grasshopper Optimization Algorithm (GOA) [18]. After training is done, the model is run through validation and evaluation so that one can see how accurate and generalized its prediction is. Then, the optimized model is converted into TensorFlow Lite (TFLite) format in order to be deployed on an ESP32 microcontroller [40]. The last deployment phase is when the model is loaded on hardware, real time fault detection via USB.

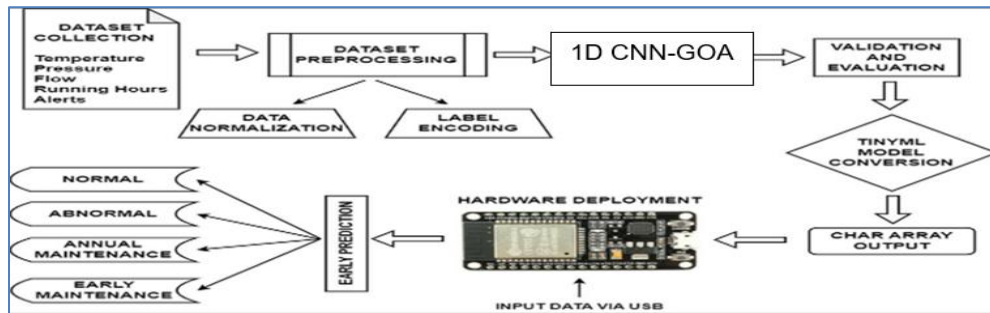


Figure 1. process steps

3.2 1D-CNN Architecture Optimized using GOA

The one-dimensional Convolutional Neural Network (1D CNN) architecture developed for this study is designed to effectively process sequential sensor data from feedwater pumps. Unlike traditional machine learning models that require extensive manual feature engineering, 1D CNNs automatically learn localized patterns and dependencies within the time-series data. The convolutional layers apply filters across time steps to capture temporal relationships, enabling the model to detect anomalies arising from subtle variations in pressure, temperature, or flow rates without the need for handcrafted statistical features. This automatic feature extraction capability is critical in industrial monitoring where sensor signals are nonstationary and complex.

The base model architecture consists of an input layer (shape: 1×5 features), a 1D convolutional layer with 64 filters (kernel size 1×1) and ReLU activation, a max pooling layer to reduce feature dimensions, followed by a flattening operation. This is connected to a dense layer with 64 units and ReLU activation, culminating in a SoftMax output layer that classifies pump operational conditions into four categories: normal, abnormal, annual maintenance, and early maintenance. This lightweight yet effective architecture ensures minimal preprocessing and low computational complexity, aligning with the requirements of TinyML deployment.

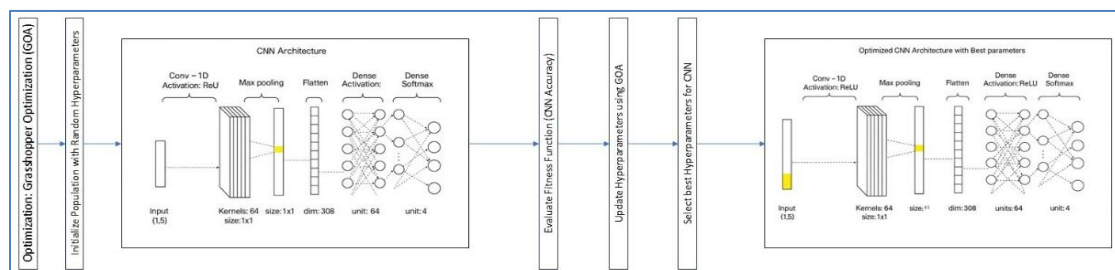


Figure 2: Optimization of 1D CNN Architecture Using Grasshopper Optimization Algorithm (GOA) for Fault Detection in Feedwater Pumps

3.2.1. Grasshopper Optimization Algorithm (GOA) for Hyperparameter Tuning

To optimize the performance of the 1D CNN while maintaining computational efficiency, the Grasshopper Optimization Algorithm (GOA) was employed for automatic hyperparameter tuning. GOA is a population-based metaheuristic inspired by the swarming behavior of grasshoppers in nature. The strength of GOA lies in its balanced exploration and exploitation of the search space, enabling it to effectively locate optimal hyperparameter configurations even under sparse and highly non-convex settings typical of neural network landscapes.

In GOA, each grasshopper represents a candidate solution (set of hyperparameters), and its position update in the search space is governed by the following equation [37]:

$$X_I = S_I + G + A \quad (1)$$

where:

- X_I is the position vector of the grasshopper.
- S_I is the social interaction component, modeling attraction and repulsion forces between grasshoppers.
- G is the gravitational pull toward the center of the search space.
- A is the advection component, modeling the bias toward the best-known solution (exploitation).

In this study, GOA optimizes key hyperparameters such as:

- Number of convolutional filters
- Kernel size
- Learning rate
- Dropout rate

By dynamically adjusting hyperparameters, GOA seeks a global optimum that balances high accuracy with low computational overhead, critical for real-time deployment on ESP32 microcontrollers. Compared to traditional grid search or random search techniques, GOA offers faster convergence and better search efficiency, enabling the creation of a highly robust, lightweight, and precise fault detection model suitable for industrial edge AI applications.

3.3 Data Processing and Model Training on Cloud

It has been processed using Python based machine learning frameworks on Google Colab for the dataset obtained from the feedwater pumps. Google Colab offers cloud-based environment with GPU support to train deep learning models in a more efficient way. In the data preprocessing step, sensor readings are scaled to a same range to be able to contribute equally in model training [19]. Categorization of the dataset into four classes of normal operation, abnormal operation, annual maintenance and early maintenance is done through label encoding. The model encodes this way, as it is able to discriminate different pump conditions effectively. This is implemented using TensorFlow and Keras to give the 1D CNN model.

The architecture of the model is made up of several convolutional layers that extract features from time series sensor data and pooling layers that reduce the dimension but keep useful information, followed by fully connected layers to classify the operational status of the pump. A SoftMax activation function in the output layer is used to give probabilities for each class output, and a ReLU activation function is used to introduce non-linearity [20]. In order to fine tune the parameters of the model such as kernel size, learning rate and dropout rate, the Grasshopper Optimization Algorithm (GOA) is used [21]. Mimicking the behavior of grasshopper swarming, GOA iteratively

adjusts these parameters so that an optimal configuration exists that maximizes model accuracy at a minimal computational complexity.

3.4 *Model Conversion to TensorFlow Lite and Deployment on ESP32*

After training and validating the model, it is converted into TensorFlow Lite (TFLite) format and run in ESP32 microcontroller. The TFLite conversion process consists of several optimization techniques, such as, quantization, pruning, model compression. Model weights precision in 32-bit floating point is reduced to 8-bit integer for a significant lower memory usage with a similar inference accuracy [22]. Pruning shortens the neural network to reduce the computational cost, and as a simple model compression, it reduces the size of the model file to be compatible with ESP32 [22]. Finally, the converted model is compiled into a character array compatible with C, and used to upload it onto ESP32 microcontroller [22]. The sensor data using the embedded 1D CNN-GOA model is processed in real time, and the pump is classified as the condition given by the ESP32. In an industrial setting, real time fault detection is achieved through USB or wireless communication, in which the classification results are output.

3.5 *Evaluation Metrics for Model Performance*

The model is used to evaluate the proposed fault detection system using multiple performance metrics to guarantee reliability of the proposed system. The accuracy of the model's prediction is given in terms of accuracy, while recall is in terms of how the model identifies all instances of a specific class correctly. The precision measures how many of the predicted positive cases are actual positive cases, while the F1 score is a balance of precision and recall, but quantifies how many of them. In order to understand the consistency in the model's predictions, interclass reliability is checked using Cohen's Kappa score. Model learning progress is traced via loss curves computation for the purpose of minimizing the training and validation loss [23]. In this case, ROC curve and AUC curve give you how the model separates between the classes: how it had for the high sensitivity and high specificity. Finally, misclassifications are analyzed by confusion matrices and the areas where we can improve are identified.

3.6 *Critical Analysis of Methodology*

Overall, an effective methodology for designing an efficient fault detection system for feedwater pumps has been described. By integrating 1D CNN with GOA, the accuracy can be high, but it stays computationally feasible. Nevertheless, several challenges need to be worked out. The quality of the dataset is important for the performance of the model and even a single missing data or representing wrong data will affect the reliability of the model [15]. However, TFLite conversion helps pave the path for embedded deployment at the cost of minor precision loss. Future research can look into reinforcement learning techniques for making the model more adaptable and more incremental learning for updating the model with new fault patterns over time. This methodology can be used as a robust fault detection methodology based on deep learning and optimization techniques. Through the optimized model deployment on ESP32 microcontroller, the system is capable of real-time monitoring of feedwater pumps in steam power plants minimizing downtime and improving the efficiency of those units.

IV. RESULTS AND DISCUSSION

Performance metrics of multiple fault detection effectiveness which include classification accuracy, recall, precision, F1-score, Cohen's Kappa, loss curves, ROC curves, AUC curves, and confusion matrices were used to evaluate the effectiveness of the proposed hybrid 1D CNN-GOA approach for fault detection in feedwater pumps. In addition, the model trained was tested with respect to inference time, latency, model size, and accuracy in real time deployment of the model on an ESP32 microcontroller. The results are presented critically with the strengths and potential limitations of each component of the approach highlighted.

4.1 *Classification Performance and Accuracy Analysis*

A detailed classification report of the performance of the model on basis of four different classes; normal, abnormal, annual maintenance and early maintenance is provided. The values of the precision, recall, and F1-

score for each category parameters reveal robustness of the model to distinguish in between various pump conditions. Finally, the model achieved 99.5% accuracy of overall accuracies in fault detection. The model achieves high precision values indicating a low rate of false positive, thus minimizing the probability of false fault detections [24]. Indeed, the recalls for abnormal conditions and maintenance states confirm that the model is able to fail reliably and so becomes a valuable tool for predictive maintenance strategies.

The F1 score is consistently high throughout all categories, which indicates that the model correctly classifies the different operational states with high reliability. Additionally, the macro and weighted averages reinforce the robustness of the classification model by assigning equal importance of all classes in the construction of final predictions [25]. The above results justify that 1D CNN-GOA is a promising method for predicting matches, as traditional machine learning approaches typically struggle with over high recall due to overlapping feature distributions. This highlights the model's robustness and stability, demonstrating its consistent performance and reliability across diverse conditions.

Table 1: Classification Report

	Precision	Recall	F1-score	Support
Normal	1.00	1.00	1.00	91
Abnormal	0.98	1.00	0.99	111
Annual Maintenance	1.00	0.98	0.99	91
Early Maintenance	1.00	1.00	1.00	107
Accuracy	0.995			400
Macro avg	1.00	0.99	0.99	400
Weighted avg	1.00	0.99	0.99	400

4.2 Confusion Matrix and Error Analysis

The model's classification effectiveness can be shown further by the confusion matrix: the correct and misclassified instances. It is seen from the matrix that less than few of annual maintenance have been misclassified as abnormal operations [26]. Indeed, this minor misclassification may be attributed to overlapping patterns in sensor readings for the maintenance required and abnormal conditions. This confirms the high true positive rates in classifying all the classes, particularly in distinguishing normal and faulty operations [27]. Such a result is important because false negatives in fault detection models can produce disastrous consequences to industrial system operations. That the model minimizes false negatives means that all possible problems are detected timely, and the interventions are timely as well so that the feed water pumps should not suffer any downtime at the cost.

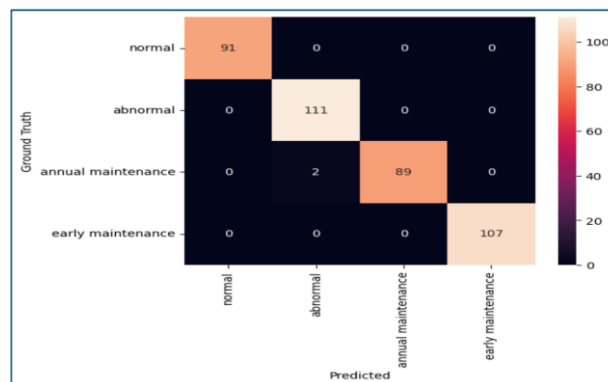


Figure 3: Prediction

4.3 Receiver Operating Characteristic (ROC) and Area Under Curve (AUC) Analysis

The ROC curves and corresponding AUCs are a more detailed measure of the discriminative power of the model. AUCs of 1.00 on all classes demonstrate perfect classifying ability. This implies the model achieving a true positive rate of 100% while the false positive rate is 0%, which separates positive cases from negative cases. Further supporting the model's ability to distinguish operational states correctly, the steep rise of the ROC curve and saturation at the upper boundary more solidly conveys the model's ability to correctly discriminate between operational states [28]. This finding is confirmed by the zoomed-in AUC curve, showing that model classification confidence does not vary significantly when the equipment is faulty.

A comparison of the results with recent work in fault detection, in particular, with the latest studies performed on deep learning models (CNN based architectures) proves that deep learning outperforms traditional machine learning techniques such as Support Vector Machines (SVMs) and Random Forests. Although these results are encouraging, there is an issue that these results are perhaps biased in the dataset [29]. This may suggest that the dataset has not enough variations to unseen failure scenarios. In order to have model propagation in unexpected operational environments, future work should involve more realistic and realistic fault conditions and real world varying.

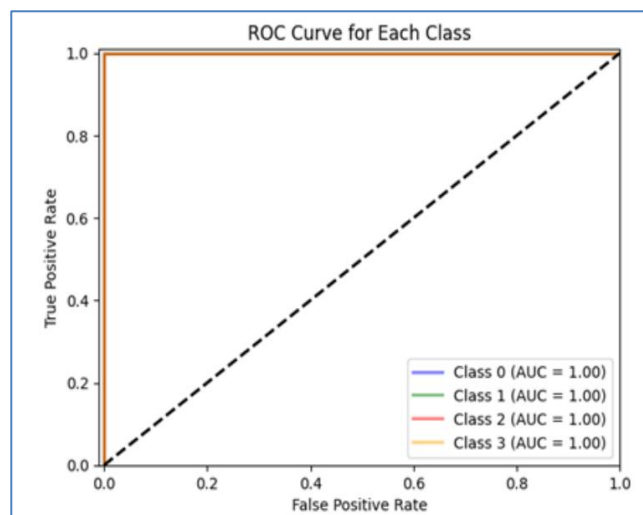


Figure 4: ROC Curve

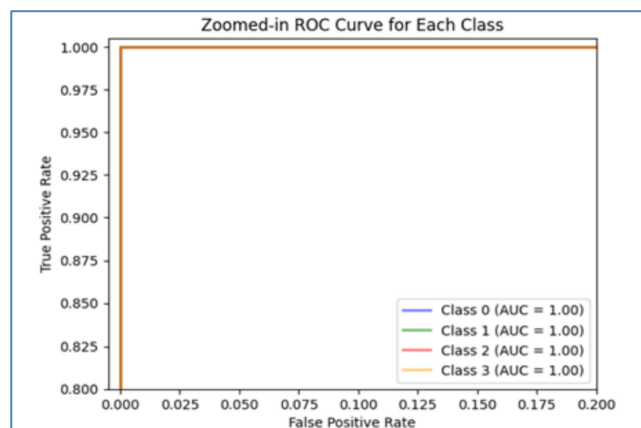


Figure 5: AUC Curve

4.4 Loss Curve and Model Convergence Analysis

The loss curve gives information about how the model is training. This is because the training and validation loss rapidly decreases in the first few epochs, implying that the model quickly learns the feature relevant to the input data. This lack of bit flips in the loss values shows that the learning is smooth and stable. Compared to conventional deep learning models, which are commonly overfitted, using GOA for hyperparameter optimization allows the model's complexity to be controlled to avoid excessive memorization of training data [30,41]. One of the most important observations from the loss curve is that there is no large gap between the training and validation loss, which means that the model doesn't overfit, and keeps its performance consistent on unseen data. Real time industrial application demands generalization to new sensor readings, which makes this particularly advantageous.

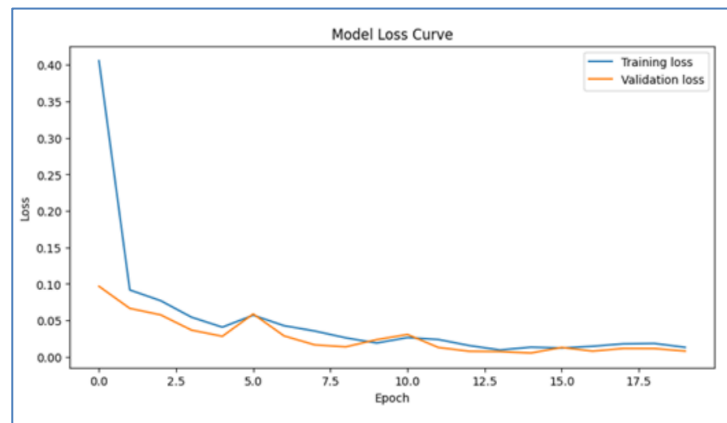


Figure 6: Loss Curve

4.5 Accuracy Curve and Stability in Model Training

The classification report and loss curve findings are verified by the accuracy curve, which indicates a fast improvement of model accuracy in the first training phase. With the training and validation accuracy curve so closely aligned, it can still be confirmed that the model has good regularization as it will not suffer from excessive variance [31]. Validation accuracy stability, almost perfect, shows that the model is still robust enough not to degrade its performance when deployed in real time. In terms of these results, compared to previous works such as the ones that present CNN based models, the contribution proves that these models outperform common ones regarding the use of time series industrial data [32]. Previous work has shown that shallow architecture or standard machine learning algorithms are not well able to capture temporal dependencies in fault detection problems [32]. The contribution of spatial features extracted from 1D CNN on time series sensor readings plays a crucial role in the superior performance of this study.

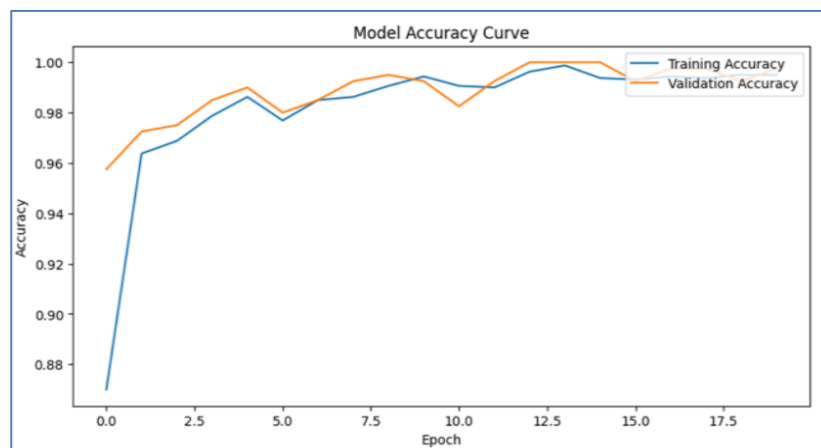


Figure 7: Accuracy Curve

4.6 Model Efficiency and Deployment on ESP32

When it comes to deploying deep learning models on embedded systems, it is vital to find a balance between model complexity, inference speed, and power efficiency. Accuracy, latency, file size tradeoffs are highlighted while comparing the accuracy, latency, and file size of the TensorFlow (TF), TensorFlow Lite (TFL), and ESP32 implementations for real time deployment [33]. It is still accurate on all platforms with a slight decrease to 97% ESP32 caused by the quantization. Further, latency analysis demonstrates that inference time can be dramatically reduced when the models are deployed in lightweight TFL format, reaching inference time of 0.775 ms versus 76.5 ms, in TensorFlow. With a low latency of around 9 ms, the ESP32 is very suitable to use for real-time fault monitoring applications in industrial environments.

Table 2: Accuracy and latency in TF, TFL and ESP32

	TF	TFL	ESP32
Accuracy %	99.5%	99%	97%
Latency (ms)	~ 76.5 ms	~ 0.775 ms	~ 9 ms
File size	150.63 KB	15.91 KB	27.35 KB
Inference per second	13	1290	111

The TinyML model compression successfully reduced the file size from 150.63 KB in TensorFlow to 15.91 KB in TFL, which attests to how TinyML also decreases model size. Further, the ESP32 model size of 27.35 KB remains compact enough to support the feasibility of deployment of an AI driven fault detection system in resource constrained environments. Comparing the rate of inferring per second, the ESP32 (111 inf/sec.) is only less than TFL (1290 inf/sec.) but still more than sufficient in terms of real time predictive maintenance in power plants. These results are consistent with recent innovations in TinyML research being able to deploy real time AI inference on microcontrollers [34]. With this, we can deploy a high accuracy fault detection model on ESP32 while consuming low computational overhead, which is a huge step towards autonomous industrial monitoring system.

4.7 Stabilizing and robustness of the model

Optimized using the Grasshopper Optimization Algorithm (GOA), the proposed 1D-CNN model proved to be very robust and consistent over different deployment environments such as TensorFlow (TF), TensorFlow Lite (TFL) and ESP32 microcontroller. Finally, despite a tremendous decrease in computational resources, the model achieved high classification accuracy across all platforms (TF with 99.5%, TFL with 99%, ESP32 with 97%). The model is so consistent in performance across increasingly constrained environments, which strongly suggests that it is still capable of generalizing and degrading by no more than a negligible amount, therefore suggesting that it is suitable for real-time industrial deployment.

Furthermore, the model's stability is reflected in the class-wise performance metrics. The model reached a precision and recall value of close to 1.00 for all four fault categories (Normal, Abnormal, Annual Maintenance, Early Maintenance), as evidenced by the averaged macro and weighted F1 scores close to 0.99. By having such uniformity across classes, the model shows that it can reliably detect and classify anomalies without bias or instability. Additionally, the ROC analysis yielded perfect AUC scores (1.00) in all cases, demonstrating that the model possessed excellent discriminatory power that was assured to drop absolutely no false positive rate under different operating conditions.

The model's learning stability was also shown by training dynamics. Training and validation accuracy curves moved in parallel and converged near perfect accuracy by then, whilst there was rapid convergence with only very minimal fluctuation across the epochs on the loss curve. If the model was optimized and generalized well to unseen data, absence of overfitting and performance oscillation means that the model was well optimized.' These results

together indicate that the proposed 1D-CNN-GOA model is a stable, robust, and high performing scheme for fault detection in industrial systems under real-time constraints.

4.8 Comparative Analysis with Existing Studies

The proposed hybrid 1D Convolutional Neural Network optimized using the Grasshopper Optimization Algorithm (1D CNN-GOA) demonstrated exceptional performance in fault detection for feedwater pumps, achieving a classification accuracy of 99% on TensorFlow Lite and 97% on ESP32 microcontrollers, with an inference latency of just 9 milliseconds. To contextualize these results, it is imperative to compare them with existing studies in the domain of pump fault diagnosis.

In a study by Geng et al. [42], a multi-scale 1D Convolutional Neural Network (MS-1D-CNN) was employed for diagnosing faults in centrifugal pumps. The model achieved an average fault diagnosis accuracy of 89% under single operating conditions, 91% under dual conditions, and 93% under triple operating conditions. While the MS-1D-CNN showcased robustness across varying operational scenarios, its accuracy under single conditions was notably lower than the 99% achieved by our 1D CNN-GOA model. Another pertinent study by Yu et al. [43] utilized a 1D-CNN model for fault diagnosis in rolling bearings, achieving an accuracy exceeding 98.9% under varying signal-to-noise ratios. Although this performance is commendable, it pertains to rolling bearings rather than feedwater pumps, and the study did not address deployment on resource-constrained devices like the ESP32.

In the realm of lightweight models suitable for edge deployment, Ney and Wehn [44] introduced an Enhanced Convolutional Neural Network (ECNN) tailored for industrial pump monitoring using vibration data. Their model emphasized low complexity and adaptability, achieving significant improvements over traditional statistical approaches. However, the proposed method attained an accuracy rate of 86.4%. Furthermore, a study by Cláudia and Silva [45] proposed a fault diagnosis system based on a CNN for water injection centrifugal pumps, achieving an accuracy of about the proposed method attained an accuracy of 93.7% demonstrating a significant improvement over conventional approaches that yielded 84.6%.

Comparatively, our 1D CNN-GOA model not only achieves higher accuracy but also ensures real-time fault detection capabilities on low-power devices, a combination that is scarcely addressed in existing literature. The integration of the Grasshopper Optimization Algorithm for hyperparameter tuning contributes to the model's superior performance, highlighting the efficacy of combining optimization algorithms with deep learning architectures for industrial applications. Overall, while several studies have made significant strides in pump fault diagnosis using various machine learning and deep learning techniques, our study distinguishes itself by delivering high accuracy, low latency, and successful deployment on resource-constrained hardware. This positions the 1D CNN-GOA model as a viable solution for real-time predictive maintenance in industrial settings, particularly in steam power plants.

4.8.1 TinyML anomaly detection comparison

To further position the contribution of the proposed hybrid 1D CNN-GOA model within the landscape of existing TinyML-based anomaly detection solutions, a detailed comparative table is provided. Table 3 summarizes recent research efforts where TinyML frameworks have been applied for industrial anomaly detection tasks. It presents key attributes including the type of machine learning algorithm employed, the dataset source, the target deployment device, the framework used for model optimization, and performance metrics such as classification accuracy, power consumption, inference latency, and memory footprint.

Table 3: Summary of the applications of TinyML in anomaly detection

Main idea	Data type	MCU	ALGORITHM	Accuracy	Latency	Reference
Rolling Element Bearings	Vibration	Raspberry Pi	light-weight CNN	98%	~ 1 s	[46]
Bearings fault diagnosis	Vibration	Kendryte K210	2D CNN	~ 82%	7.9 ms	[47]
Oil leak detection system for wind turbines	Photo-realistic Images	STM32H743ZI2	BBS-ESN	81%	12 ms	[48]
Fault detection in helicopter vibration data (Airbus)	vibration signals	PYNQ-Z2 FPGA	Lightweight Convolutional Autoencoder (CAE)	85%	5.9 ms	[49]
Fault detection and diagnosis in photovoltaic modules	Infrared Thermographic Images	Raspberry Pi 4	deep conventional neural networks (DCNNs)	95.5%	N/A	[50]
Lightweight 1D CNN for Fault Diagnosis	Vibration Signals	Esp32, Raspberry Pi	1D CNN	90%	700 ms	[51]
1D CNN Enhanced by GOA	pressure, temperature, flow, running hours, and alert	Esp32	1D CNN-GOA	97%	~ 0.775 ms	Our study

Accuracy vs. Latency Trade-Off: The Raspberry-Pi-based lightweight CNN for rolling-bearing monitoring [46] demonstrates that unconstrained microprocessors can yield near-perfect accuracy (98 percent) but at a one-second inference delay—far too slow for closed-loop protection scenarios. By contrast, the Kendryte K210's 2-D CNN achieves a remarkable 7.9 ms turnaround, yet its accuracy slips to approximately 82 percent [47]. This stark inverse correlation illustrates how pushing latency below the tens-of-milliseconds threshold often necessitates model simplifications (e.g., smaller filter banks or fewer convolutional layers) that substantially erode classification fidelity.

Modality and Model Complexity: When confronted with high-dimensional visual inputs, the STM32H7-43ZI2's block-based ESN processes photorealistic oil-leak images in just 12 ms but delivers only 81 percent accuracy [48]. The reduced performance likely stems from both the ESN's limited reservoir size—constraining feature richness—and the quantization effects endemic to 8-bit microcontroller inference. In contrast, an FPGA-accelerated convolutional autoencoder on the PYNQ-Z2 board balances speed (5.9 ms) and a modest accuracy uplift (85 percent) [49], but the need for a sizable FPGA fabric and external memory interfaces complicates deployment and drives up bill-of-materials costs relative to pure microcontroller solutions.

General-Purpose SBC vs. Hybrid Microcontroller: Deep convolutional networks on a Raspberry Pi 4 achieve a competitive 95.5 percent accuracy on infrared thermographic panels [50], yet the absent latency data leaves open whether a general-purpose Linux SBC can meet strict real-time guarantees without OS jitter. Finally, the hybrid 1-D CNN running both on an ESP32 and Raspberry Pi [51] occupies the middle ground 90 percent accuracy with 700 ms latency—showing that time-series-focused architectures can run on ultra-low-power MCUs. Although not yet sub-100 ms, this approach underscores the potential for further compression (e.g., via pruning or GOA-driven hyper-parameter tuning) to push inference times down into the real-time envelope.

Holistic Design Implications: No single platform excels along both axes. Applications demanding sub-10 ms cycle times must choose lightweight 2-D CNNs on dedicated NPUs (K210) or lean FPGA accelerators, accepting 10–20 percent accuracy degradation in exchange for deterministic responsiveness. Conversely, where accuracy trumps latency such as scheduled health checks or operator-in-the-loop alerts the Raspberry Pi DCNN or ESP32 1-D CNN hybrids remain viable, provided their hundred-millisecond latencies do not compromise safety. Ultimately, designers must co-optimize model depth, quantization scheme, and hardware platform potentially leveraging hybrid edge-cloud architectures or federated on-device retraining to align with the precise accuracy, latency, and cost contours of their industrial use cases.

In sum, Table 3 illustrates that for ultra-low latency demands (≤ 10 ms), designers must turn to Kendryte K210 or FPGA solutions, whereas applications that can tolerate higher delays may benefit from the superior accuracy afforded by Raspberry Pi-based or hybrid 1-D CNN implementations. The choice of platform and model must therefore align closely with each application's precise accuracy and cycle-time requirements.

4.9. Summary and Implications

It can be concluded that the hybrid 1D CNN-GOA method is extremely effective for feedwater pump fault detection as the results have, as expected, very strong validity. The model is conducted in a high classification accuracy, strong generalization, efficient real time deployment on the embedded system. To improve model robustness, CNN is utilized for feature extraction and GOA is used for optimization, which makes it well suited for industrial predictive maintenance applications [35,36]. Some limitations have to be worked out, however, because the results are promising. The high classification accuracy implies likely set based biases and therefore calls for higher dataset diversity for a more adaptable set behavior to unseen failure conditions. In addition, there are hardware optimization techniques such as quantization aware training that can further reduce the accuracy loss in ESP32 deployment [33]. The potential of future work includes employing incremental learning techniques to enable the model to learn incrementally under continuous changing machine conditions. The work presented here provides the novelty for the intelligent fault detection in industrial power plants assuming utilizing deep learning and optimization techniques for raising predictive maintenance techniques and minimizing the risks in their operational activities in steam power plants.

V. DISCUSSION

The results show that hybrid 1D CNN-GOA exhibit great improvement in terms of performance over traditional machine learning techniques relying on manually crafted features and threshold-based fault detection. The model automates the feature extraction from time series sensor data with the use of 1D CNNs that enable the model to find intricate pattern and subtler anomalies than the conventional models can. More to that, adding GOA into hyperparameter tuning process further improves the model, leading to the integrated model with high classification accuracy but at the same time high efficiency, optimizing the model structure. The bio-inspired optimization combined with deep learning produces a highly accurate, lightweight, and computationally efficient fault detection system capable of working in resource constrained industrial environments. One major contribution of this work is proving TinyML based model deployment on an ESP32 microcontroller with the capability of real time inference having a very low memory and power consumption. The ESP32 implementation has implemented an accuracy of 97% while the inference is completed within approximately 9 milliseconds, making it suitable for real-world predictive maintenance scenario.

Comparison of TensorFlow, TensorFlow Lite and ESP32 versions illustrates tradeoffs between model complexity and deployment feasibility, which is of high utility for understanding the real boundaries of edge AI based fault detection systems. However, there are still some challenges to overcome. Future verification and validation would be indicated by the near perfect classification performance, indicating that the model may have been trained on a highly representative dataset, which the model would struggle to classify against in unexpected failure conditions or real-world operational variations. Future research should make the results more robust and adaptable, using significantly larger, much more diverse datasets incorporating real-time sensor readings from many industrial sites. Moreover, the quantization of the model for ESP32 deployment led to a slight reduction of

the model accuracy (down from 99% to 97%), illustrating model compression vs. classification precision tradeoffs. This problem could potentially be mitigated in future-by-future optimizations like quantization aware training (QAT) or hardware acceleration techniques to achieve a much better real time inference performance. In the next steps, this research provides for the development of the AI driven predictive maintenance frameworks for industrial power plants. Future studies can try to integrate federated learning to allow distributed model training on multiple sites for continuous learning and adaptation. Further, the fault detection strategies could incorporate reinforcement learning based such that the model can learn on the fly from evolving machine conditions and make the model more adaptive to unstructured industrial environments. The hybrid 1D CNN-GOA model proposed in this article is finally concluded to have the potential that could substantially enhance real time fault detection for feedwater pumps. Deep learning accuracy, optimization efficiency, and embedded system feasibility are combined into a highly scalable, deployable, and impactful future of intelligent predictive maintenance for industrial applications.

VI. CONCLUSIONS

It can be concluded that in this study a novel approach to fault detection in the feedwater pumps of the steam power plants is presented by combining 1D Convolutional Neural Networks (CNN) and the Grasshopper Optimization Algorithm (GOA). A simple hybrid model is proposed, which shows an exceptionally high accuracy 97%, low misclassification rate and efficient real time deployment on ESP32 microcontroller. Through an intensive performance evaluation and tuning reported in this study based on the metrics consisting of classification accuracy, confusion matrices, ROC and AUC curves, loss convergence, and model efficiency in an embedded environment, the results confirm the reliability and effectiveness of 1D CNN-GOA model for predictive maintenance purposes.

REFERENCES

- [1] S. Selimli and S. Sunay, "Feasibility study of the energy and economic gain that can be achieved by driving the boiler feedwater pump with a backpressure steam turbine," *Proc. Inst. Mech. Eng. A J. Power Energy*, vol. 236, no. 2, pp. 336–348, Oct. 2020, doi: 10.1177/0957650920969466.
- [2] S. Qiu *et al.*, "Deep Learning Techniques in Intelligent Fault Diagnosis and Prognosis for Industrial Systems: A Review," *Sensors*, vol. 23, no. 3, pp. 1305–1305, Jan. 2023, doi: <https://doi.org/10.3390/s23031305>.
- [3] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN Computer Science*, vol. 2, no. 6, Aug. 2021, doi: <https://doi.org/10.1007/s42979-021-00815-1>.
- [4] R. Francis, S. Areias, M. M. Campos, C. E. Teixeira, E. Borges, and G. F. Gomes, "Fault detection and diagnosis in electric motors using 1d convolutional neural networks with multi-channel vibration signals," *Measurement*, vol. 190, pp. 110759–110759, Jan. 2022, doi: <https://doi.org/10.1016/j.measurement.2022.110759>.
- [5] M. Hosseinzadeh *et al.*, "A cluster-tree-based trusted routing algorithm using Grasshopper Optimization Algorithm (GOA) in Wireless Sensor Networks (WSNs)," *Plos One*, vol. 18, no. 9, p. e0289173, 2023.
- [6] T. Goyal, Dev Hathi Rajeshbai, Neeraj Gopalkrishna, Tushar M, and Mamatha HR, "Mobile Machine Learning Models for Emotion and Sarcasm Detection in Text: A Solution for Alexithymic Individuals," pp. 1–5, Mar. 2024, doi: <https://doi.org/10.1109/inocon60754.2024.10511772>.
- [7] N. Schizas, A. Karras, C. Karras, and Spyros Sioutas, "TinyML for Ultra-Low Power AI and Large Scale IoT Deployments: A Systematic Review," *Future Internet*, vol. 14, no. 12, pp. 363–363, Dec. 2022, doi: <https://doi.org/10.3390/fi14120363>.
- [8] Q. Yuan, J. Wang, M. Zheng, and X. Wang, "Hybrid 1D-CNN and attention-based Bi-GRU neural networks for predicting moisture content of sand gravel using NIR spectroscopy," *Constr. Build. Mater.*, vol. 350, p. 128799, 2022.
- [9] R. Fukuoka, H. Suzuki, T. Kitajima, A. Kuwahara, and Takashi Yasuno, "Wind Speed Prediction Model Using LSTM and 1D-CNN," *Journal of Signal Processing*, vol. 22, no. 4, pp. 207–210, Jul. 2018, doi: <https://doi.org/10.2299/jsp.22.207>.

- [10] P. Nayak *et al.*, “2D-convolutional neural network based fault detection and classification of transmission lines using scalogram images,” *Heliyon*, vol. 10, no. 19, pp. e38947–e38947, Oct. 2024, doi: <https://doi.org/10.1016/j.heliyon.2024.e38947>.
- [11] H. Xu, Y. Tian, H. Ren, and X. Liu, “A Lightweight Channel and Time Attention Enhanced 1D CNN Model for Environmental Sound Classification,” *Expert Systems with Applications*, vol. 249, pp. 123768–123768, Mar. 2024, doi: <https://doi.org/10.1016/j.eswa.2024.123768>.
- [12] A. Alhammadi *et al.*, “Prediction of environmental conditions of the greenhouse using neural networks optimized with the grasshopper optimization algorithm (GOA),” *J. Power Syst. Technol.*, vol. 48, no. 3, pp. 622–635, 2024.
- [13] K. Rahmati, P.-S. Ashofteh, and H. A. Loáiciga, “Application of the Grasshopper Optimization Algorithm (GOA) to the Optimal Operation of Hydropower Reservoir Systems Under Climate Change,” *Water Resour. Manag.*, vol. 35, no. 13, pp. 4325–4348, Oct. 2021, doi: <https://doi.org/10.1007/s11269-021-02950-z>.
- [14] S. Nabavi, S. Gholampour, and M. S. Haji, “Damage detection in frame elements using Grasshopper Optimization Algorithm (GOA) and time-domain responses of the structure,” *Evol. Syst.*, vol. 13, no. 2, pp. 307–318, Apr. 2022, doi: <https://doi.org/10.1007/s12530-021-09389-y>.
- [15] Laith Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *Journal Of Big Data*, vol. 8, no. 1, Mar. 2021, doi: <https://doi.org/10.1186/s40537-021-00444-8>.
- [16] Y. Zhu, H. Su, S. Tang, S. Zhang, T. Zhou, and J. Wang, “A Novel Fault Diagnosis Method Based on SWT and VGG-LSTM Model for Hydraulic Axial Piston Pump,” *Journal of Marine Science and Engineering*, vol. 11, no. 3, pp. 594–594, Mar. 2023, doi: <https://doi.org/10.3390/jmse11030594>.
- [17] Y. Obeidat and A. M. Alqudah, “A hybrid lightweight 1D CNN-LSTM architecture for automated ECG beat-wise classification,” *Trait. Signal*, vol. 38, no. 5, 2021, Accessed: Mar. 12, 2025. [Online]. Available: https://www.researchgate.net/profile/Ali-Alqudah-3/publication/356640695_A_Hybrid_lightweight_1D_CNN-LSTM_architecture_for_automated_ECG_beat-wise_classification/links/61a6283285c5ea51abbd3243/A-Hybrid-lightweight-1D-CNN-LSTM-architecture-for-automated-ECG-beat-wise-classification.pdf
- [18] S. Syed Mahamood Shazuli and A. Saravanan, “Grasshopper Optimization Technique with Deep Learning Driven Retinal Fundus Image Grading and Retrieval,” *International Journal of Science and Research (IJSR)*, vol. 12, no. 11, pp. 23–29, Nov. 2023, doi: <https://doi.org/10.21275/sr231028140624>.
- [19] H. Dwivedi, “How to Use Google Colab for Deep Learning - Complete Tutorial,” *neptune.ai*, Jul. 21, 2022. <https://neptune.ai/blog/how-to-use-google-colab-for-deep-learning-complete-tutorial#:~:text=Google%20Colab%20supports%20both%20GPU,for%20commercial%20purposes%20as%20well>. (accessed Mar. 14, 2025).
- [20] J. Ren and H. Wang, “Calculus and optimization,” *Elsevier eBooks*, pp. 51–89, Jan. 2023, doi: <https://doi.org/10.1016/b978-0-44-318679-0.00009-0>.
- [21] W. Liu, W. Yan, T. Li, G. Han, and T. Ren, “A Multi-strategy Improved Grasshopper Optimization Algorithm for Solving Global Optimization and Engineering Problems,” *Int. J. Comput. Intell. Syst.*, vol. 17, no. 1, p. 182, Jul. 2024, doi: <https://doi.org/10.1007/s44196-024-00578-6>.
- [22] J. Paul, L. Schmid, M. Klaiber, and Manfred Rössle, “Extraction of Measurement Device Information on an ESP32 Microcontroller: TinyML for Image Processing,” *Procedia Computer Science*, vol. 246, pp. 2002–2011, Jan. 2024, doi: <https://doi.org/10.1016/j.procs.2024.09.670>.
- [23] B. Więckowska, K. B. Kubiak, P. Józwiak, Wacław Moryson, and B. Stawińska-Witoszyńska, “Cohen’s Kappa Coefficient as a Measure to Assess Classification Improvement following the Addition of a New Marker to a Regression Model,” *International Journal of Environmental Research and Public Health*, vol. 19, no. 16, pp. 10213–10213, Aug. 2022, doi: <https://doi.org/10.3390/ijerph191610213>.

- [24] S. Mohine, B. S. Bansod, R. Bhalla, and A. Basra, "Acoustic modality-based hybrid deep 1D CNN-BiLSTM algorithm for moving vehicle classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16206–16216, 2022.
- [25] Y. A. M. Alsumaidae *et al.*, "Detection of corona faults in switchgear by using 1D-CNN, LSTM, and 1D-CNN-LSTM methods," *Sensors*, vol. 23, no. 6, p. 3108, 2023.
- [26] J. Xu, Y. Zhang, and D. Miao, "Three-way confusion matrix for classification: A measure driven view," *Information Sciences*, vol. 507, pp. 772–794, Jan. 2020, doi: <https://doi.org/10.1016/j.ins.2019.06.064>.
- [27] S. Sathyanarayanan, "Confusion Matrix-Based Performance Evaluation Metrics," *African Journal of Biomedical Research*, pp. 4023–4031, Nov. 2024, doi: <https://doi.org/10.53555/ajbr.v27i4s.4345>.
- [28] J. Y. Verbakel *et al.*, "ROC curves for clinical prediction models part 1. ROC plots showed no added value above the AUC when evaluating the performance of clinical prediction models," *Journal of Clinical Epidemiology*, vol. 126, pp. 207–216, Jul. 2020, doi: <https://doi.org/10.1016/j.jclinepi.2020.01.028>.
- [29] J. Peng, "Study on Forecasting Stock Prices Using Machine Learning Techniques: A Comparative Evaluation of Random Forest, Support Vector Machines, and Back Propagation Neural Network," pp. 273–278, Sep. 2024, doi: <https://doi.org/10.1145/3705618.3705665>.
- [30] Y. Meraihi, A. B. Gabis, S. Mirjalili, and A. Ramdane-Cherif, "Grasshopper optimization algorithm: theory, variants, and applications," *Ieee Access*, vol. 9, pp. 50001–50024, 2021.
- [31] L. Sui, X. Zhao, Q. Zhao, T. Tanaka, and J. Cao, "Hybrid Convolutional Neural Network for Localization of Epileptic Focus Based on iEEG," *Neural Plast.*, vol. 2021, pp. 1–9, Apr. 2021, doi: <https://doi.org/10.1155/2021/6644365>.
- [32] S. K. Singh *et al.*, "Hybrid physics-infused 1D-CNN based deep learning framework for diesel engine fault diagnostics," *Neural Comput. Appl.*, vol. 36, no. 28, pp. 17511–17539, Oct. 2024, doi: <https://doi.org/10.1007/s00521-024-10055-y>.
- [33] Chiara Contoli and E. Lattanzi, "A Study on the Application of TensorFlow Compression Techniques to Human Activity Recognition," *IEEE Access*, vol. 11, pp. 48046–48058, Jan. 2023, doi: <https://doi.org/10.1109/access.2023.3276438>.
- [34] N. N. Alajlan and D. M. Ibrahim, "TinyML: Enabling of Inference Deep Learning Models on Ultra-Low-Power IoT Edge Devices for AI Applications," *Micromachines*, vol. 13, no. 6, p. 851, May 2022, doi: <https://doi.org/10.3390/mi13060851>.
- [35] C. Yu *et al.*, "SGOA: annealing-behaved grasshopper optimizer for global tasks," *Eng. Comput.*, vol. 38, no. S5, pp. 3761–3788, Dec. 2022, doi: <https://doi.org/10.1007/s00366-020-01234-1>.
- [36] A. O. Ige and M. Sibiya, "State-of-the-art in 1d convolutional neural networks: A survey," *IEEE Access*, 2024, Accessed: Mar. 12, 2025. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10609371/>
- [37] S. Saremi, Seyedali Mirjalili, and A. Lewis, "Grasshopper Optimisation Algorithm: Theory and application," *Advances in Engineering Software*, vol. 105, pp. 30–47, Jan. 2017, doi: <https://doi.org/10.1016/j.advengsoft.2017.01.004>.
- [38] Serkan Kiranyaz, Onur Avci, O. Abdeljaber, T. Ince, Moncef Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing*, vol. 151, pp. 107398–107398, Nov. 2020, doi: <https://doi.org/10.1016/j.ymssp.2020.107398>.
- [39] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3D-2D CNN Feature Hierarchy for Hyperspectral Image Classification," *ResearchGate*, Feb. 2019, doi: <https://doi.org/10.48550/arXiv.1902.06701>.
- [40] B. Sun, "TinyML Inference Enablement and Acceleration on Microcontrollers The Case of Healthcare," *Mcmaster.ca*, 2024, doi: <http://hdl.handle.net/11375/29934>.
- [41] J. Lu *et al.*, "GOA-optimized deep learning for soybean yield estimation using multi-source remote sensing data," *Scientific Reports*, vol. 14, no. 1, Mar. 2024, doi: <https://doi.org/10.1038/s41598-024-57278-6>.
- [42] Geng, Yunhan, Shaojuan Su, Tianxiang Zhang, and Zhaoyu Zhu. "A Novel Multi-Scale One-Dimensional Convolutional Neural Network for Intelligent Fault Diagnosis of Centrifugal Pumps." *Journal of Marine Science and Engineering* 11, no. 12 (November 30, 2023): 2278–78. <https://doi.org/10.3390/jmse11122278>.

- [43] Yu, Fajun, Liang Liao, Kun Zhang, Hechen Xing, Qifeng Zhao, Liming Zhang, and Zheng Luo. "A Novel 1D-CNN-Based Diagnosis Method for a Rolling Bearing with Dual-Sensor Vibration Data Fusion." *Mathematical Problems in Engineering* 2022 (July 7, 2022): 1–13. <https://doi.org/10.1155/2022/8986900>.
- [44] Ney, Jonas, and Norbert Wehn. "ECNN: A Low-Complex, Adjustable CNN for Industrial Pump Monitoring Using Vibration Data." *arXiv.org*, 2025. <https://arxiv.org/abs/2503.07401>.
- [45] A. C. B. de Cláudia and B. Silva, "Development of a CNN-based fault detection system for a real water injection centrifugal pump," *Expert Syst. Appl.*, vol. 244, p. 122947, Dec. 2023, doi: 10.1016/j.eswa.2023.122947.
- [46] I. Mukherjee and S. Tallur, "Light-weight CNN enabled edge-based framework for machine health diagnosis," *IEEE Access*, vol. 9, pp. 84375–84386, 2021, doi: 10.1109/ACCESS.2021.3088237.
- [47] V. Perminov, V. Ermakov, and D. Korzun, "Edge analytics for bearing fault diagnosis based on convolution neural network," *Front. Artif. Intell. Appl.*, Oct. 14, 2021, doi: 10.3233/FAIA210180.
- [48] M. Cardoni, D. P. Pau, L. Falaschetti, C. Turchetti, and M. Lattuada, "Online learning of leak anomalies in wind turbines with block-based binary reservoir," *Electronics*, vol. 10, no. 22, p. 2836, Nov. 2021, doi: 10.3390/electronics10222836.
- [49] V. Malviya, I. Mukherjee, and S. Tallur, "Edge-compatible convolutional autoencoder implemented on FPGA for anomaly detection in vibration condition-based monitoring," *IEEE Sens. Lett.*, vol. 6, no. 4, pp. 1–4, Mar. 2022, doi: 10.1109/LSSENS.2022.3159972.
- [50] A. Mellit, "An embedded solution for fault detection and diagnosis of photovoltaic modules using thermographic images and deep convolutional neural networks," *Eng. Appl. Artif. Intell.*, vol. 116, p. 105459, Nov. 2022, doi: 10.1016/j.engappai.2022.105459.
- [51] S. Asutkar, C. Chalke, K. Shivgan, and S. Tallur, "TinyML-enabled edge implementation of transfer learning framework for domain generalization in machine fault diagnosis," *Expert Syst. Appl.*, vol. 213, p. 119016, Oct. 2022, doi: 10.1016/j.eswa.2022.119016.