Journal Européen des Systèmes Automatisés
Vol. 58, No. 6, June, 2025, pp. 1123-1133
Journal homepage: http://iieta.org/journals/jesa

A Centralized Federated Learning Algorithm Based Multi Classification Predictive Maintenance in Industrial Internet of Things System

Ruaa W. Abdalah[1*], Osamah F. Abdulateef[1], Ali H. Hamad[2]

[1] Automated Manufacturing Engineering Department, AL Khwarizmi College of Engineering, University of Baghdad, Baghdad 10071, Iraq
[2] Information and Communication Engineering Department, AL Khwarizmi College of Engineering, University of Baghdad, Baghdad 10071, Iraq

Corresponding Author Email: ruaa.abd2204@kecbu.uobaghdad.edu.iq

https://doi.org/10.18280/jesa.580604

Received: 6 April 2025
Revised: 4 May 2025
Accepted: 15 May 2025
Available online: 30 June 2025

Keywords:
predictive maintenance, Industry 4.0, IoT IIoT, DNN, multiclassification

ABSTRACT

Predictive maintenance (PdM) is essential for maintaining sustained operation for Industry 4.0 systems. Using artificial intelligence is crucial when PdM is required. However, there are several difficulties because of growing requirements for secure learning when uploading and downloading data in cloud servers. This led to the use of a training algorithm that preserves the privacy and security of the dataset. This work proposed a Federated Learning (FL) algorithm in PdM emphasizing its benefits in terms of quicker training time, lower latency, low power consumption, and, mainly, more security and privacy. The proposed system uses FL with deep neural network (DNN) model for both client and global models. Three client systems are represented by three AC motors equipped with different sensors, such as temperature, vibration, and current, which have been interfaced with Raspberry Pi through the I2C communication protocol. The weight values for the models are uploaded and downloaded between the local model and the global model in the cloud server using the MQTT Internet of Things (IoT) protocol. Results show good training performance metrics enhancement for the FL algorithm over the local model training without FL, where the accuracy has been increased from (0.9915) to (0.9983) in FL while the loss is decreased from (0.0232) to (0.0104) in FL.

1. INTRODUCTION

Predictive maintenance (PdM) and the Industrial Internet of Things (IIoT) together mark a significant change in equipment management, especially when deep learning (DL) is used to interpret time series data from IIoT devices [1]. IoT is a scalable network system of distributed devices and connectivity that is having a dramatic impact on a wide range of application areas. Particularly in the context of Industry 4.0, the increasing velocity and volume of data generated by Industrial IoT (IIoT) devices, combined with advanced analytics capabilities, enables intelligent industrial operations, which improve manufacturing processes and operational efficiency [2].

Four main types of maintenance in industry: Predictive Maintenance (PdM) improves equipment longevity and prevents unplanned machine downtime by identifying possible problems before they materialize. While Reactive Maintenance (RM) focuses on fixing equipment as soon as it breaks down, Proactive Maintenance (PRM) aims to prevent failures by addressing their root causes. This suggests that because failure duration is unpredictable, downtime may be significant. Consequently, regardless of the machine's real condition, Preventive Maintenance (PM) is carried out regularly. Unlike reactive, which is less costly but results in higher expenses if equipment breaks down and creates unexpected downtime. Predictive methods enable planning and minimize unexpected downtime, while preventative measures may result in needless maintenance. Because PdM entails condition monitoring, it only minimizes unexpected downtime, while preventative measures may result in needless maintenance. Because Predictive maintenance entails condition monitoring, it is only carried out when required. As a result, there should be less need for replacement components, which would save expenses. Additionally, it helps prevent excessive maintenance, which frequently leads to problems. While predictive maintenance primarily focuses on scheduling maintenance for equipment situation, preventive maintenance may entail reevaluating systems to prevent breakdown. Using contemporary technology like data analysis, machine learning, and Internet of Things sensors yields further benefits. As a result, forecasts became more precise. Equipment life can be increased by addressing issues early. PdM necessitates real-time data analysis and storage while accounting for the many facets and impacts of the signals gathered. Predictive maintenance use various techniques, such as artificial intelligence, time series analysis, data analytics, , and different failure modes ,etc. [3].

In an IIoT setting, federated learning enables several devices (or clients) to work together to train machine learning models without disclosing private information. In PdM, senssory data should de secured when uploaded to a cloud server. Federated learning resolves privacy issues while facilitating efficient model training by maintaining data decentralization [4]. Combining both PdM and FL will enhance prediction accuracy while protecting privacy through local data processing. This tactic promotes inter-organizational collaboration while reducing the cost of data transit. FL is therefore an effective tool in the context of Industry 4.0 [5].

Currently, deep learning is applied across various fields, including PdM. When ample historical data is provided, deep learning models outperform both statistical and conventional machine learning approaches. At the core of deep learning there are artificial neural networks (ANN), which mimic the brain's functioning. Unlike shallow networks with one or two hidden layers, deep learning utilizes more complex architectures [6]. Sensors directly attached to industrial equipment produce data that condition monitoring (CM) systems analyze to detect irregularities and forecast potential failures. This enhances reliability and optimizes maintenance costs. CM relies on diverse sensors, including those measuring temperature and vibration, to track system parameters, model behavior, and detect anomalies signaling wear or damage. Such monitoring is critical, as equipment failures can disrupt entire production processes, resulting in costly repairs and significant financial losses. As simple sensors alone cannot provide enough power, the data is usually transferred to an external server or cloud for analytics jobs such as machine learning models.

Industrial facilities are often located far away from the actual cloud data centers, which can limit network capabilities. For example, they can suffer from high latency and unstable network connections, which can cause network link congestion. Early fault diagnosis leads to faster repairs and reduced follow-up costs [7]. It is the reason why FL as a distributed machine learning paradigm has been more and more popular for the IIoT recently. FL involves training the models locally using real data sources, and then aggregating them into a global model without sharing any training data. This makes it possible to integrate the knowledge of other faculties while keeping training ML models inside the organizational limits of IIoT locations [8].

This work introduces and applies a PdM algorithm that combines FL with a DNN. The propsed system has been tested on a benchmark system of three AC motors equipped with four sensor types. A substantial offline dataset was gathered and trained the DNN model on a public server. During the online process, real-time data is gathered from different sensors and fed into the DNN model to predict potential failures. Additionally, the method integrates condition monitoring technologies with the FL framework, enabling local model training at individual IIoT sites while facilitating cloud-based model aggregation to enhance knowledge sharing across different locations.

The rest of this paper is organized as follows: Section 2 presents the current research in PdM systems. prerequisites necessary to design the proposed system is introduced in section 3. Section 4 shows how the custom dataset being gathered from real system. Section 5 describes the proposed PdM design and section 6 introduced the discussion and results of the proposed system. Finally, Section 7 introduce the conclusion of this work.

2. RELATED WORK

Much researches have been done on FL for PdM systems using various modeling techniques. Singh et al. [9] introduced FL architecture designed specifically for predictive maintenance is shown in this study, enabling several clients to train machine learning models concurrently while maintaining decentralized data. Numerous machine learning techniques were employed, such as Random Forest, Logistic Regression, with a remarkable accuracy of 93.17% in the federated learning technique, these findings suggest that in PdM applications, FL can successfully strike a compromise between privacy and performance requirements. Zhao et al. [10] provided an autoencoder-based FL approach to address this issue, which uses vibration sensor data from spinning machines to enable distributed training on edge devices that are situated on-site and next to the machines under observation. using two real-world datasets and many testbeds. Liu et al. [11] introduced a decentralized system based on FL to guarantee low-latency and safe industrial data processing. It uses the distributed Alternating Direction Method of Multipliers (ADMM) method, which has the benefits of parallelism and decomposability to improve the suggested data processing model demonstrates that the proposed inexact algorithms can achieve comparable statistical accuracy while reducing response time by up to 17.2% and 58%, respectively when compared to other existing algorithms using industrial datasets from a thermal power plant for steam prediction case. Putra et al. [12] introduced Decentralized FL (DFL) method for joint learning in the identification of bearing faults. The collaborative framework's susceptibility to assaults can be reduced by utilizing the decentralized FL idea. To cut down on communication overhead, suggested DFL incorporates continuous learning strategies. With an accuracy percentage of 96.08% and a learning time reduction of up to 37.52%, the data show that decentralized collaborative learning performs satisfactorily. Praveena et al. [13] introduced a FL enabled advanced PdM framework for the prediction and detection of defects in industrial machinery, based on a hybrid MLP-GRU model. The IMS bearing dataset, C-MAPSS dataset, and pump sensor dataset are three separate datasets that are incorporated into the framework and represent different kinds of equipment and failure scenarios. Through integration and preprocessing, the records are meticulously merged into a training dataset, which facilitates the development of a hybrid MLP-GRU model capable of identifying complex fault patterns and temporal correlations. All things considered; the proposed framework represents a significant advancement in the predictive maintenance of industrial machinery. Its implementation of federated learning techniques and a hybrid MLP-GRU model architecture yields encouraging results of 0.94% accuracy in real-world industrial applications.

Guduri et al. [14] provided a Federated Convolutional Neural Network with Temporal Attention Mechanism (FedCNN-TAM) is presented in this work for PdM tasks in IoT networks. This approach improves prediction accuracy by using temporal aspects in sensor data. FedCNN-TAM exhibits a notable performance improvement over representative models when tested on the CMAPSS dataset. Lu et al. [15] proposed a brand-new clustering-based FL technique in which customers are grouped according to how similar their datasets are. To estimate dataset similarity between clients without exchanging data directly, probabilistic deep learning models are trained, and each client analyzes the prediction uncertainty of the models of other clients on its local dataset. Based on relative forecast accuracy and uncertainty, clients are then grouped for FL. Three bearing fault datasets were used in the experiments; two were publically accessible, and one was specifically gathered for this study. Von Wahl et al. [16] presented an approach to a thorough framework for combining federated learning (FL), foundation models (FMs), and artificial intelligence of things (IoT) technologies. Extensive simulations on a typical fleet of aircraft showed that the combined FM and FL strategy consistently outperformed standalone implementations in several important measures, such as convergence speed, model size efficiency, and forecast accuracy. For fleet-wide optimization, predictive maintenance, and real-time monitoring. Sanchez et al. [17] presented an approach to a hybrid framework that blends models of Multi-Layer Perceptions (MLP) and Gated Recurrent Units (GRU). This system created especially for predicting and detecting defects in industrial machinery, enabling the efficient use of dispersed information while protecting privacy and data security. Abdalah et al. [18] laid the foundation for our work. Building on their system of AC motors, our approach treats each motor as an independent system, with training performed on its respective local dataset only. The proposed system uses the deep learning technique to train the system with an accuracy of 100% and a loss of 0.0014.

3. PRELIMINARIES

To enable preventive maintenance, predictive maintenance leverages IIoT and data analytics to check machines status and predict faults. In this work, a predictive maintenance system has been built based on federated deep learning.

3.1 Working sensors

Data has been gathered from the workbench using a variety of sensor types, including: As a 13-bit, 3-axis shaking sensor, the ADXL345 is an ideal device for monitoring motor vibrations. It has good qualities such as a tiny profile, a compact form factor, and extremely low power consumption. With the ACS712 current sensor module, the motor's current is tracked. It is an entirely integrated current sensor that does not experience magnetic hysteresis. With just one 5V power source, it provides linear current sensing for both AC and DC currents via the Hall Effect. It also uses a low-resistance current conductor and an analog-to-digital converter (ADS1115). Certain essential qualities must be possessed by the temperature sensor that was employed in this study. The MLX90614 contactless temperature sensor was chosen for its wide range, high accuracy and precision. Its affordability and compact size also make it a good fit for the requirements of this study. A SHT21 digital humidity and temperature sensor was also used to measure the outdoor temperature [19]. That operating efficiency is greatly increased by predictive maintenance with vibration sensors. Makes highly accurate predictions about production stoppages by using machine learning techniques. This proactive strategy reduces downtime and boosts overall productivity by enabling prompt maintenance interventions [20]. The specifications of the sensors used in this work are shown in Table 1.

Table 1. Specification of the sensors [20]

Sensor	Definition	Range	Accuracy	Operating Voltage	Current Consumption	Address
ADXL345	3-axis accelerometer	±2g, ±4g, ±8g, ±16g	0.0038g	2V-3.6V	0.11mA	0×53
ACS712	Current	±5A, ±20A, ±30A	65mV /±5A	5V	10mA	None
MLX90614	Contactless	70°-380°	0.02℃	3.6V-5.5V	0.5mA	0×5A, 0×5B
SHT-21	Temperature/ humidity	0-100% 40°-125°	0.3℃ , 2% humidity	3.3V-5.5V	0.1mA	0×40
ADS1115	A/D converter	0-5V	±1% to ±0.1%	0-10V	1mA	0×48
ADXL345	3-axis accelerometer	±2g, ±4g, ±8g, ±16g	0.0038g	2V-3.6V	0.11mA	0×53

3.2 I2C protocol

The Raspberry Pi's mobility, parallel, affordability, and low power consumption make it a very practical, promising, and perfect technology for IoT applications. The Raspberry Pi is an affordable, energy-efficient computer capable of performing many tasks typically handled by a traditional desktop PC, making it ideal for IoT applications. One tool for accelerating processes and calculations is the Raspberry Pi 4, it is a quad core processor [21]. The different types of sensors is connected using the I2C protocol (a two-line interface enables communication between different sensors at the same time). Sensors are connected to the RPi using two wires Serial Clock (SCL) and Serial Data (SDA).

3.3 Deep neural networks

The proposed DNN model is described in Table 2. The model consists of eight layers, with the input layer matching the feature dimensions and the output layer represents the five classes of faults [22]. In this work, ReLU is used as the activation function in hidden layers while Softmax is used in the output layer which enable multi classification required in this work [23]. The proposed DNN model layers used in this work is chosen such that a decision impacted by a number of input features. By striking a balance between model capacity and issue complexity, it enables the network to learn abstract patterns without going into too much detail. This setup maintains computational efficiency while reducing the chance of overfitting, particularly when data is few. It also conforms to the empirical results of experiments and the principles of architectural design. In general, eight layers each of 50 neurons work well for a variety of datasets and classification applications. The loss function used in this work is categorical cross-entropy which is best work with multiclass labels.

3.4 Federated learning

In the conventional federated learning (FL) framework as shown in Figure 1, multiple edge devices function as independent clients connected to a central global server. Typically, a predetermined number of FL clients (D) are randomly chosen from a group of edge devices that have expressed willingness to participate in the training process. During a training iteration at time t, each selected client (d) retrieves the global model parameters (θ_t) from the server and trains the model using its local dataset. If m$_d$ denotes the number of data samples for a client, then $\sum_{d=1}^{D} m_d$ =M, where M is the total size of data samples from D clients. The FL attempts to optimize $f(\theta)$ [24].

Table 2. Hyperparameter of DNN

Hyperparameter	Value
neuron/layer	50
Layers	8
Epochs	25
Activation-function	ReLU, SoftMax
Loss-function	Categorical cross entropy
Optimizer	Adam
Classes	5
Batch	64

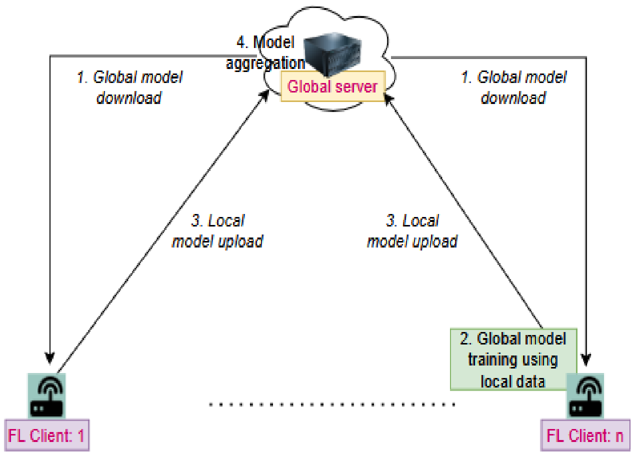


Figure 1. Traditional fedrated learning [24]

$$\min f(\theta)$$
$$f(\theta) = \sum_{d=1}^{D} \frac{m_d}{M} F_{d\,(\theta)} \tag{1}$$
$$F_d(\theta) = \frac{1}{m_d} \sum_{i\in m_d} f^{i}(\theta)$$

In the local training phase, each client d optimizes the global model by minimizing its loss function $f^{i}(\theta)$, which corresponds to the i-th sample in its datasetOptimization techniques like stochastic gradient descent (SGD) or Adam are employed for this purpose. Once local training is complete, the client transmits its updated version of the global model, denoted as θ_{t+1}^{d}, back to the global server.

$$\theta_{i+1}^{d} = \theta_i - \alpha_d \lambda_d \tag{2}$$

Here, α_d represents the learning rate, while λ_d denoting the gradient calculated by client d using its local dataset and the parameters θ_t.

The global server combines the collected local models and calculates the enhanced global model (θ_{t+1}) in the following manner.

$$\theta_{i+1} = \sum_{d=1}^{D} \frac{m_d}{M} \theta_{i+1}^{d} \tag{3}$$

In FL, clients download the improved global model for subsequent training rounds, continuing this iterative process until the model converges. Federated Averaging (FedAvg) is a decentralized learning method where multiple participants collaboratively train a model while keeping their raw data private. Each client performs local training using its dataset and sends only the adjusted model parameters—not the raw data—to a central server. The server aggregates these updates by taking a weighted average of the model weights, with the weights determined by the size of each client's dataset. This approach enhances privacy, reduces bandwidth usage, and scales well across numerous clients. However, it faces challenges like data heterogeneity and communication efficiency. Overall, FedAvg is crucial for effective federated learning. The FedAvg formula can be describe as:

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^{k} \tag{4}$$

where, n_k represents the aggregate number of data samples from all devices; $w_{t+1}^{k}$is the weight vector of the local model modified by device k in round t+1. Given a loss function η and a gradient ∇L(w_t^k;b) computed regarding the model weights w_t^k and input data b, the local and global model updates can be presented as illustrated in Algorithm 1.

Algorithm 1: Federated averaging (FedAvg) algorithm.

```
Input: Initial global model w, number of clients K, number of rounds t
Output: Updated global model $w_t$
for each round t = 1, 2 ..., t do
  for each client k = 1,2, ..., K in parallel do:
    batches = divide $D_k$ client dataset into chunk of size B
      $w_t^k \leftarrow w_t$, initialize local model on each client
      for each batch b ∈ batches do:
        $w_t^k \leftarrow w_t^k - \eta \nabla L(w_t^k;b)$
      end for
  end for
  $w_{t+1} \leftarrow \frac{1}{K} \sum_{k=1}^{K} w_{t+1}^{k}$ ;
end for
```

4. DATASET IMPLEMENTATION

This work utilizes various sensors, including current sensors, contactless temperature sensors, ambient temperature with humidity sensors, and 3-axis accelerometers, to collect sensor data. The vibration sensor produced three distinct features corresponding to the X, Y, and Z axes, creating multiple columns within the same dataset and effectively recording a fault status every one second sample time. Data collection was conducted both online and offline, capturing detailed fault classifications such as normal operation, heavy load, overcurrent, stopping and vibration. These conditions were precisely induced in the benchmark system to generate the necessary dataset for each fault type.

To simulate vibrations across the motor's three axes, an unbalanced mass was employed. Variations in motor current were replicated using capacitors of different sizes, while a mechanical braking system mimicked stop and heavy load scenarios. Table 3 outlines the quantity and distribution of datasets for each fault category.

Table 3. Size and counts of data set for each class

Failure Type	File Size (KB)	Data Records
Normal	843	20159
Vibration	351	6762
Stop rotating	641	16466
Havey load	141	2711
Over current	202	4591

In this work, a deep neural network model was used to classify the four types of faults with the normal operation as the fifth class. Before training the DNN model, the dataset needs to be properly preprocessed such as normalized and labeled as follows:

i) Dataset Normalization: it is a key preprocessing in machine learning training. In this work MinMaxScaler algorithm has been used to normalize the features (train dataset and test dataset). This method scales features to a given range, typically between 0 and 1, to ensure that no single feature has an excessive influence on the model learning process. This improves model performance and speeds up the learning process. It also ensures a fair feature contract by ensuring that features with a wide range do not overwhelm features with a narrow range.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (5)$$

ii) Dataset labeling: Dataset labeling Categorical classes of data is preprocessed by One-hot encoding algorithm which is a technique used to convert string data into a numerical format that machine learning can process. This involves generating numbers for each distinct category.

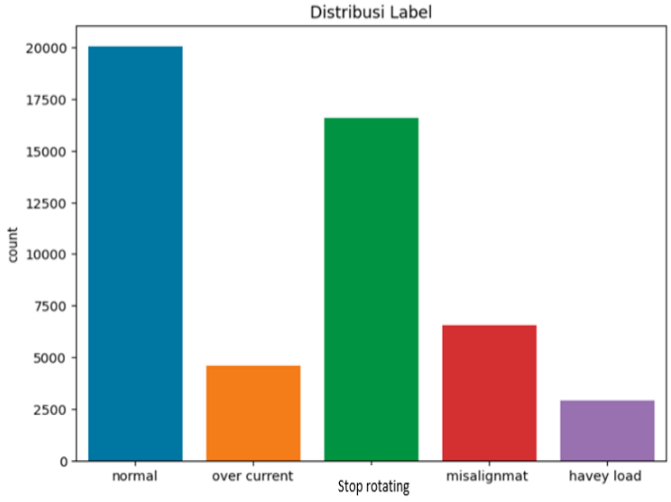


Figure 2. Five classes distribution

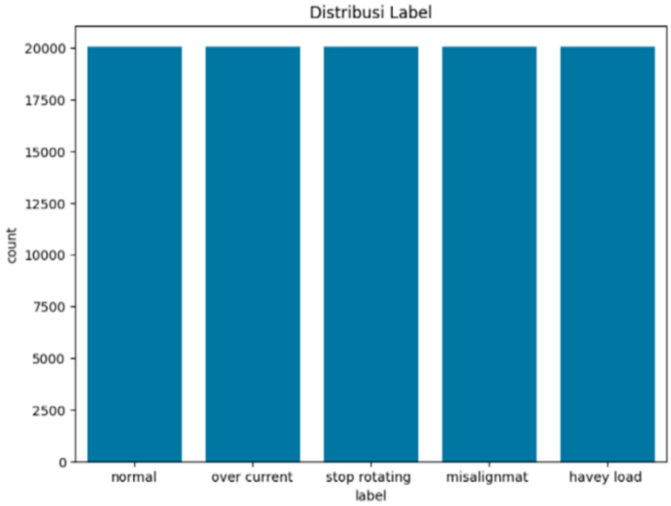


Figure 3. Five classes after balance

Table 4. Sample of data collected

Accel x	Accel y	Accel z	Amb_Temp	Object_Temp	Current	Label
0.437	-0.0937	7.937	17.82	31.05	0.334	Normal
0.812	0.781	7.875	17.82	31.01	0.332	Normal
2.093	-0.281	7.281	17.83	31.03	0.335	Normal
6.187	5.468	15.968	35.175	36.23	0.410	Over current
7.406	8.718	15.978	35.185	36.21	0.446	Over current
1.593	0.40	7.312	33.073	31.97	0.404	Stop rotating
1.781	0.25	7.187	33.062	31.99	0.400	Stop rotating
0.0281	-0.0232	-0.248	0.010	34.51	28.525	Misalignment
0.0367	-0.0294	-0.267	0.009	34.43	28.515	Misalignment
12.656	15.969	2.281	23.731	55.09	0.758	Heavy load
15.968	15.968	-11.625	23.742	55.17	0.692	Heavy load

iii) Dataset splitting: The dataset was divided into 60% for training, 10% for validation, and 30% for testing. The training part, typically the largest portion, is used to train the model. The testing part evaluates the final trained model using testing data where the model hasn't seen this data before. The validation part is used along with the train set to enhance the training performance. Figure 2 shows the five classes among the collected dataset: normal, overcurrent, heavy load, stopping, and vibration. The data is unevenly distributed among these categories.

iv) Class balance: To resolve class imbalance in datasets, in this work the Synthetic Minority Over-sampling Technique (SMOTE) is used. This gives a DNN model more balanced perspective on the issue and increases the likelihood that it will learn to categorize all classes efficiently by boosting the representation of minority classes in training data. This contributes to the confirmation that the SMOTE operation was effective in achieving more equitable class allocations. Figure 3 shows the five classes after balance.

The confusion matrix is a 5×5 array, with 5 denoting the number of classes. A binary classification model employs a 2×2 confusion matrix, whereas a multiclass model uses an N×N matrix, where N denotes the classes. The value $c_{i,}$ denotes the frequency with which a status of class i was categorized as class j. Moreover, a comprehensive array of metrics can be extracted from the complete confusion matrix.

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} TP(C_i)}{\sum_{i=1}^{N}\sum_{j=1}^{N} C_{i,j}} \tag{6}$$

$$recall(C_i) = \frac{\mathrm{TP}(C_i)}{\mathrm{TP}(C_i) + \mathrm{FN}(C_i)} \tag{7}$$

$$precision\ (C_i) = \frac{\mathrm{TP}(C_i)}{\mathrm{TP}(C_i) +\ \mathrm{FP}(C_i)} \tag{8}$$

$$\mathrm{F1}(C_i) = \frac{2 *\ \mathrm{recall(Ci)} * \mathrm{pression(Ci)}}{\mathrm{recall}(C_i) + \mathrm{recall}(C_i)} \tag{9}$$

where, *TP (C_i)* True positive for class$_i$ FN false negative for class$_i$. An example of the gathered data is displayed in Table 4. The DNN model leverages the dataset to build a predictive system that conducts real-time monitoring on the cloud server and forecasts potential failures.

5. PROPOSED SYSTEM DESIGN

The proposed benchmark testbed consists of three motors each equipped with four sensors: 3-axis vibration sensor, two temperature sensors a contactless infrared sensor and ambient sensor, and current sensor. All these sensors have been interfaced with three Raspberry Pi 4 for each motor via I2C communication protocol where each sensor has its unique address to avoid packet collision. Each Raspberry Pi then communicates with the cloud server via the MQTT protocol.

The Raspberry pi has a local DNN model (i.e., three DNN model or clients) while a global model is presented in the cloud server. Figure 4 shows the proposed system architecture.

The proposed system has two mode of operation, local and global models training and predictive mode. Before train the models, a realistic dataset has been collected from the motor in different failed types which are deliberately done.

Within the training mode, a FL approach has been used where in each round the local models are trained on their part of the dataset where each part consists of all features (sensor) but with two failure classes only. After the local model is trained to its desired accuracy and loss values, the final weights of the model are uploaded to the cloud server. In this stage, the three local models upload their weights to the cloud and an aggregation algorithm based on FedAvg is used to compute the new global model weight. Then the global model will train again for the new weights. After that, the global model will download its weight to all local models which will retain their models again. This process is the first round, the proposed system consists of 20 rounds. By doing so, the local model which trains only on its part of the dataset and fewer classes of failure will be trained for all types of classes and this is the main advantage of FL in addition to the security and privacy of the sensory data. Algorithm 1, mentioned earlier, illustrates the machine learning training process. In the second mode of operation, the local model cloud predicts the failure types, and also the global model does. Also, the proposed system included data visualization to show the different types of failures.

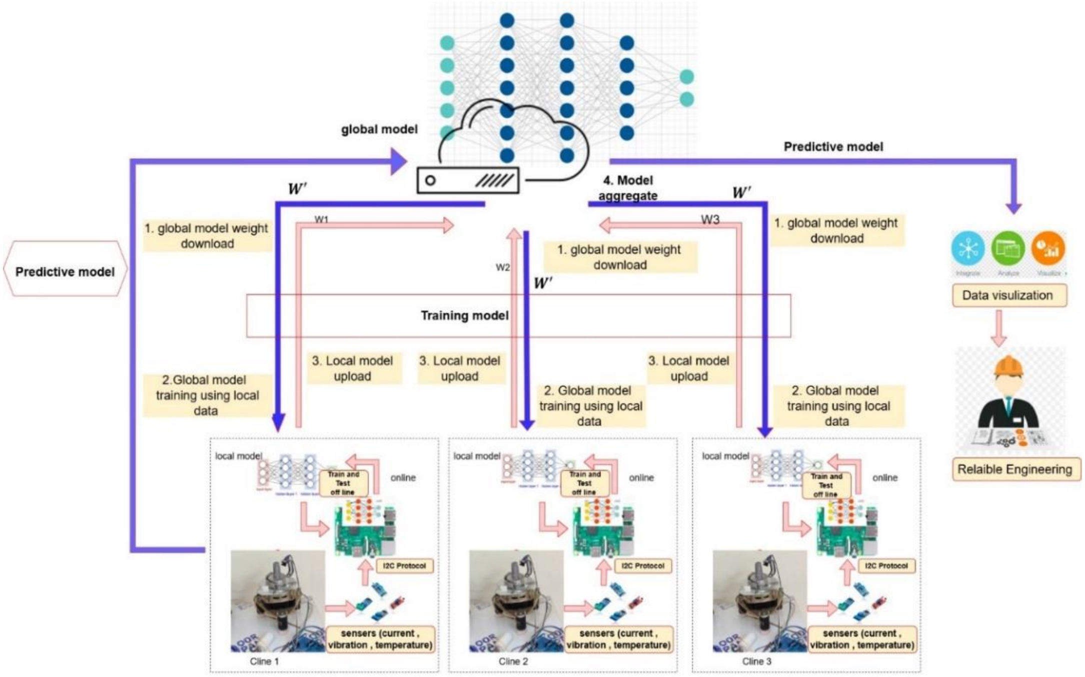


Figure 4. Proposed system architecture

6. RESULTS AND DISCUSSION

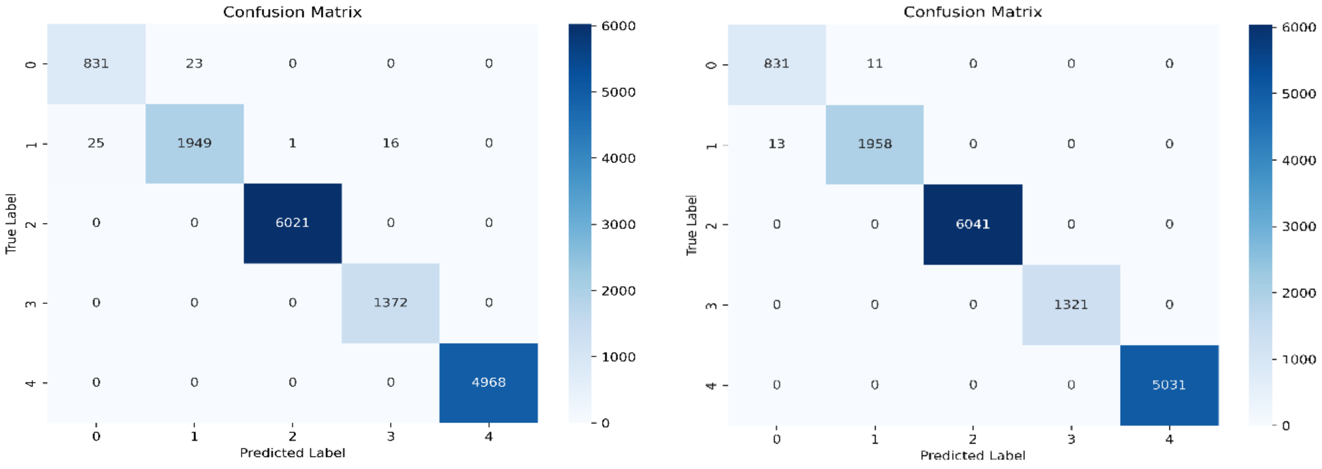
The confusion matrix gives a thorough assessment of the model's performance by comparing the predicted labels to the actual labels for each class. At work, the confusion matrix was presented as a heatmap, with each cell representing the number of incidences for a certain true label (y-axis) and predicted label (x-axis). The diagonal elements denote successfully identified occurrences (true positives and true negatives), whereas the off-diagonal components reflect misclassifications. A confusion matrix misclassifies data points when a machine learning model misclassifies. Here, the model "gets it wrong." Consider the confusion matrix's off-diagonal members to understand misclassification. Figure 5 shows the confusion metrics with and without FL. These cells' predicted class differs from their actual class. Row-wise: Numbers outside the diagonal column (indicating the real class) represent misclassifications where the class was projected as something else. As a comparision between the two confusion matrecies, it could be noted that within the FL the misclassification is less than that without FL. The classification report of different metrics with and without FL is shown in Table 5. In the following a discription for the most important metrics:

Accuracy: The high diagonal values in the confusion matrix indicated a high proportion of properly identified examples, which was consistent with the model's total accuracy of 0.9983. This highlights the efficiency of the federated learning technique and the DNN architecture in capturing the underlying patterns inside the dataset. Figure 6 shows the local model training and validation performance of accuracy and loss. These results are based on the overall dataset for each client. The local model, trained on the whole dataset without using federated learning, earned a respectable accuracy of test (0.9965), and train (0.9915) indicating its capacity to learn from the data and generate correct predictions. However, the centralized method has drawbacks in terms of privacy, generality, and scalability. Centralizing the data for training presented possible privacy problems, and the model's exposure to a single dataset may have limited its capacity to generalize to new or varied data. Employing federated learning to train a local model on distributed data, as shown in Figure 7, resulted in a large accuracy boost to (0.9983) demonstrating the capacity to exploit heterogeneous data while maintaining privacy. This method improved the model's performance by increasing accuracy, recall, and F1-score while also addressing critical issues like overfitting and generalization. Federated learning reduced the danger of overfitting and improved generalization to previously unknown data by training on localized datasets and pooling updates. Furthermore, validation approaches were critical in this process, with local validation tracking individual model growth and global validation offering an unbiased assessment of the final model's performance. Our analysis is shown in Table 6 accuracy and other metrics. The best accuracy (0.9983) was obtained by the federated learning strategy, indicating its exceptional capacity to generalize from a variety of data sources.

Loss Minimization: The federated method demonstrated the least amount of loss (0.0104), indicating that it was more resilient and could efficiently learn from the dispersed data. The federated model shows a deeper comprehension of the underlying patterns and relationships in the data while reducing loss.

Table 5. Classification report a-without FL, b- with FL

(a)					(b)				
Classification	**Precision**	**Recall**	**F1-Score**	**Support**	**Classification**	**Precision**	**Recall**	**F1-Score**	**Support**
Havey load (0)	0.97	0.99	0.98	856	Havey load (0)	0.98	0.99	0.99	842
Misalignment (1)	0.99	0.99	0.99	1957	Misalignment (1)	0.99	0.99	0.99	1971
Normal (2)	1.00	1.00	1.00	6029	Normal (2)	1.00	1.00	1.00	6041
Over current (3)	1.00	1.00	1.00	1362	Over current (3)	1.00	1.00	1.00	1362
Stop rotating (4)	1.00	1.00	1.00	5002	Stop rotating (4)	1.00	1.00	1.00	5031
Accuracy			1.00	15206	Accuracy			1.00	15206
Macro avg	0.99	0.99	0.99	15206	Macro avg	1.00	1.00	1.00	15206
Weighted avg	1.00	1.00	1.00	15206	Weighted avg	1.00	1.00	1.00	15206



(a) (b)

Figure 5. Confusion metrics a-without FL, b- with FL

Table 6. Compare results between the global model and the local model (without/with federated learning)

Model	**Accuracy**		**Precision**		**Recall**		**Loss**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
Local model (without FL)	0.9915	0.9965	0.9916	0.9965	0.9915	0.9965	0.0232	0.0128
Glient model	0.9917	0.9975	0.9922	0.9975	0.9931	0.9975	0.0187	0.012
Global model	0.9956	0.9983	0.9966	0.9983	0.9966	0.9983	0.0124	0.0104

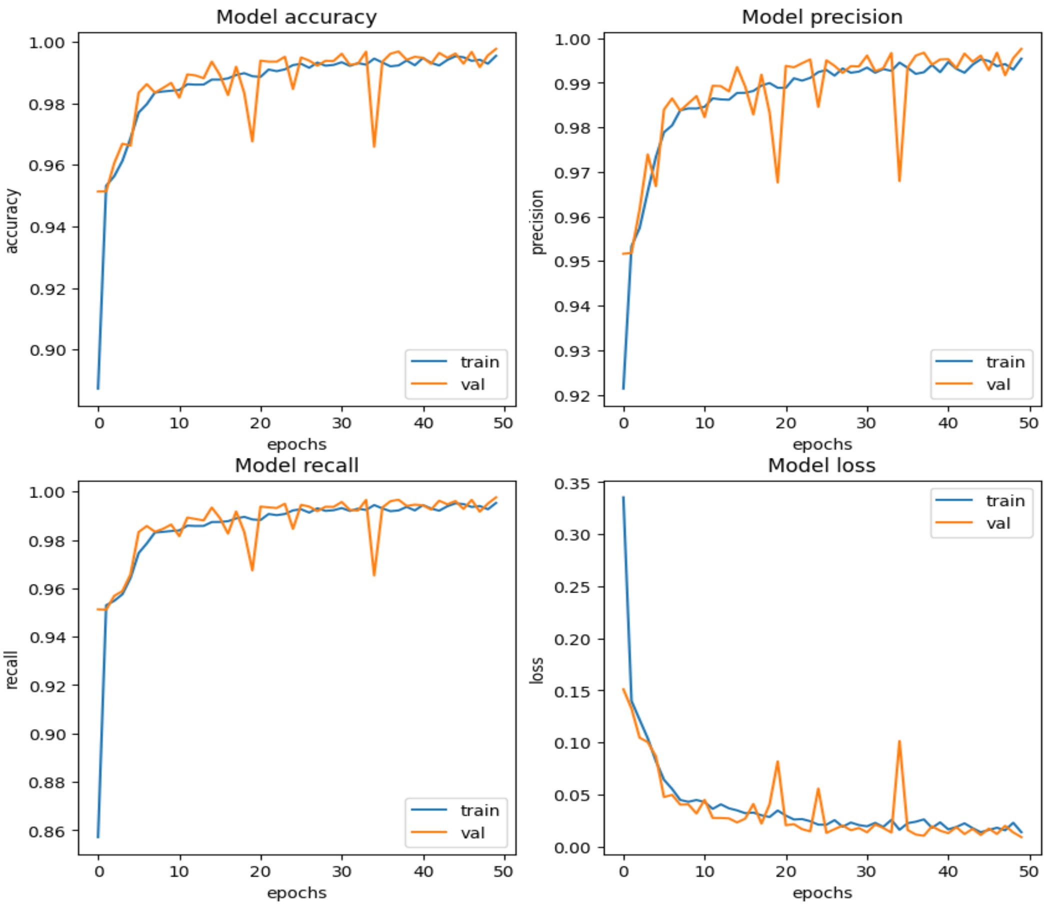
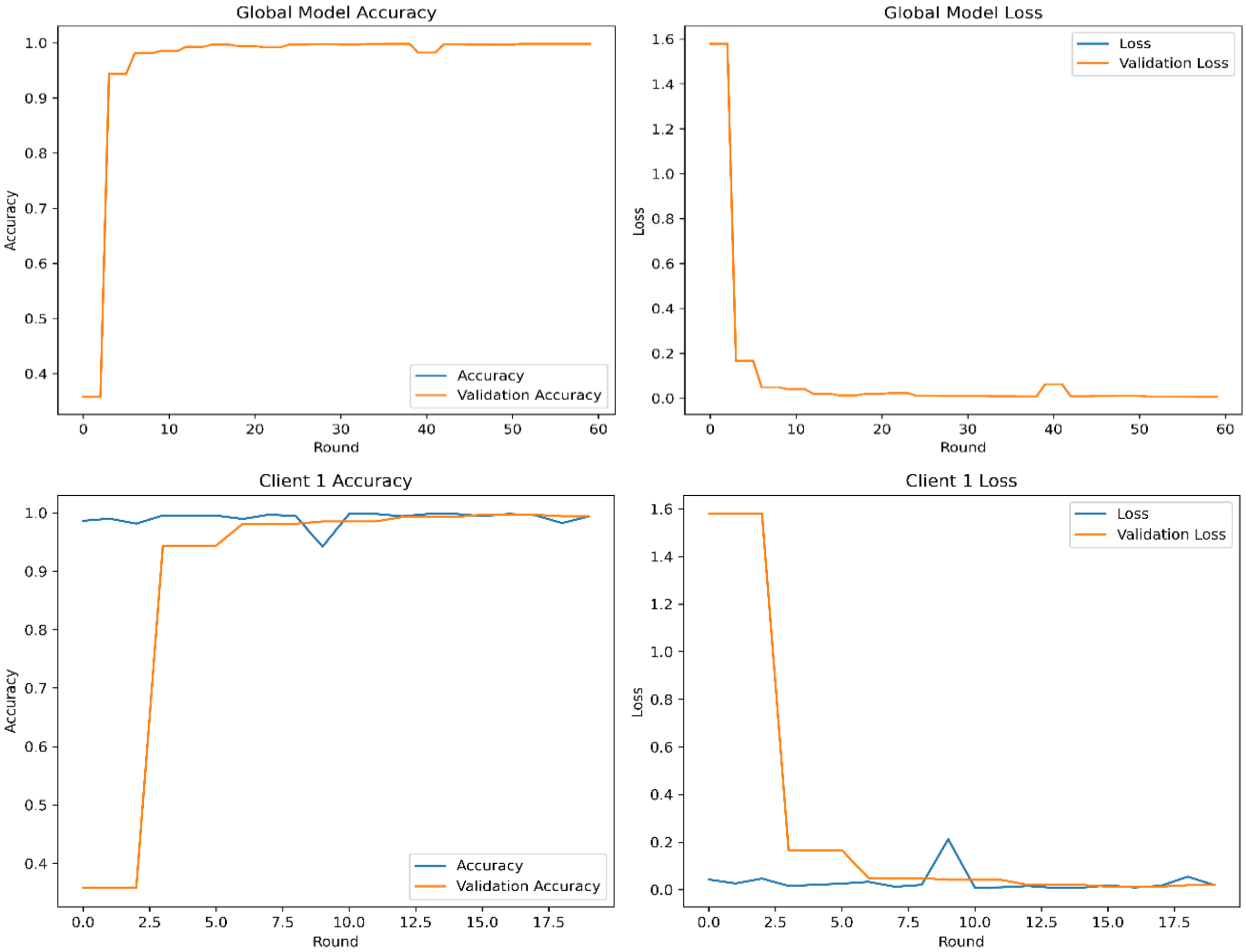


Figure 6. DNN model performance for stader system (with out FL). a) Accuracy b) Precision c) Recall d) Loss



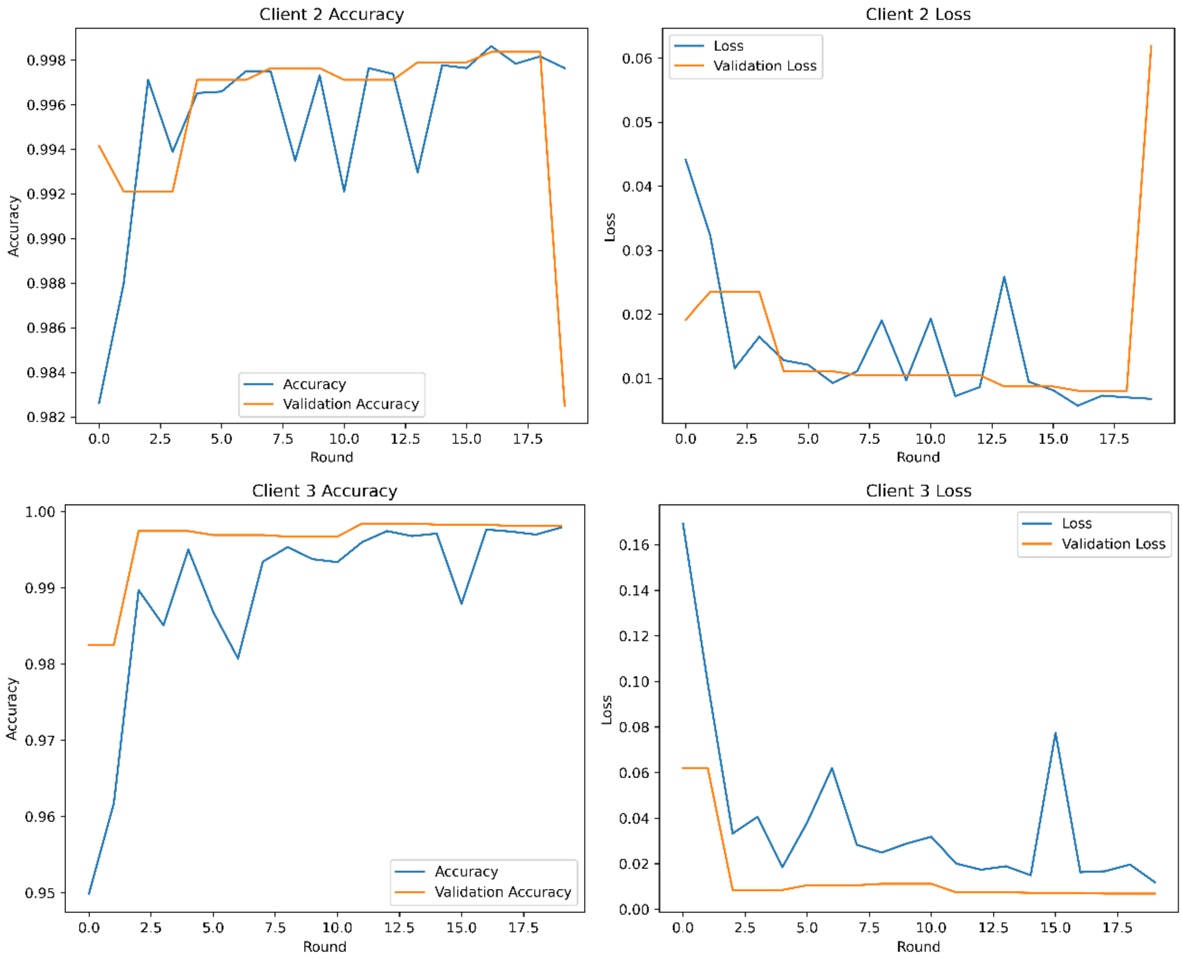


Figure 7. DNN model performance for stader system (with FL). a) Accuracy b) Precision c) Recall d) Loss

7. CONCLUSION

This study demonstrates the effectiveness of federated learning (FL) in enhancing predictive maintenance (PdM) systems within industrial settings. By employing FL, we successfully trained a multi-class classification model on distributed data without compromising data privacy. The implementation of four well-known FL aggregation schemes on various datasets, including a novel real-world dataset, revealed that federated learning significantly improves model accuracy, achieving an increase from 99.15% to 99.83% while reducing loss from 0.0232 to 0.0104. Additionally, federated learning (FL) faces a number of difficulties, such as:

Communication costs: During training in federated learning, clients and a central server (or clients directly in decentralized models) communicate model updates on a regular basis. These updates may be rather big, particularly for large models. In terms of bandwidth utilization, this results in significant communication expenses. In order to combat this, methods like quantization—which lowers the accuracy of weights—and spacing—which sends just a portion of updated weights—are employed to shrink the amount of model updates prior to transmission.

Latency: The training process can be considerably slowed down by high latency between clients and the server, or between clients. Communication times between clients might differ greatly in geographically dispersed locations or those with different network equipment, which can impact the aggregation process's synchronization. Therefore, asynchronous deep learning is employed, where the server may aggregate changes as soon as it receives them from any client, rather than waiting for all clients to give their updates in each round. Although it necessitates more intricate aggregation processes, this lessens the impact of sluggish clients.

The results indicate that FL provides a robust alternative to traditional local and global training methods, particularly for applications involving sensitive or dispersed data sources. This approach not only maintains data privacy but also enables high-performance model training. Our findings support the further exploration of federated learning across diverse domains, encouraging collaborative learning while safeguarding confidential information. Overall, federated learning presents a promising pathway for future advancements in predictive maintenance and industrial IoT applications.

REFERENCES

[1] Obaid, M.H., Hamad, A.H. (2024). Internet of Things based oil pipeline spill detection system using deep learning and LAB color algorithm. Iraqi Journal for Electrical and Electronic Engineering, 20(1): 137-148. https://doi.org/10.37917/ijeee.20.1.14

[2] Mohammed, N.A., Abdulateef, O.F., Hamad, A.H. (2023). An IoT and machine learning-based predictive maintenance system for electrical motors. Journal

Européen des Systèmes Automatisés, 56(4): 651-661. https://doi.org/10.18280/jesa.560414

[3] Tsai, Y.H., Chang, D.M., Hsu, T.C. (2022). Edge computing based on federated learning for machine monitoring. Applied Sciences, 12(10): 5178. https://doi.org/10.3390/app12105178

[4] Pruckovskaja, V., Weissenfeld, A., Heistracher, C., Graser, A., Kafka, J., Leputsch, P., Kemnitz, J. (2023). Federated learning for predictive maintenance and quality inspection in industrial applications. In 2023 Prognostics and Health Management Conference (PHM), Paris, France, pp. 312-317. https://doi.org/10.1109/PHM58589.2023.00064

[5] Llasag Rosero, R., Silva, C., Ribeiro, B., Santos, B.F. (2024). Label synchronization for Hybrid Federated Learning in manufacturing and predictive maintenance. Journal of Intelligent Manufacturing, 35(8): 4015-4034. https://doi.org/10.1007/s10845-023-02298-8

[6] Becker, S., Styp-Rekowski, K., Stoll, O.V.L., Kao, O. (2022). Federated learning for autoencoder-based condition monitoring in the industrial internet of things. In 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, pp. 5424-5433. https://doi.org/10.1109/BigData55660.2022.10020836

[7] Ganji, A., Usha, D., Rajakumar, P.S. (2025). Hybrid machine learning framework with data analytics model for privacy-preserved intelligent predictive maintenance in healthcare IoT. Journal of Computer Science, 21(1): 1-12. https://doi.org/10.3844/jcssp.2025.1.12

[8] Farahani, B., Monsefi, A.K. (2023). Smart and collaborative industrial IoT: A federated learning and data space approach. Digital Communications and Networks, 9(2): 436-447. https://doi.org/10.1016/j.dcan.2023.01.022

[9] Singh, A., Sampath, R., Raj, S.A.D., Mary, G.I., Aarthi, G., Kumar, R. (2024). Federated learning for predictive maintenance: Model comparisons and privacy advantages. In 2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS), Bengaluru, India, pp. 1408-1415. https://doi.org/10.1109/ICICNIS64247.2024.10823350

[10] Zhao, H., Sui, M., Liu, M., Zhu, C., Xun, W., Xu, B., Zhu, H. (2024). Are you diligent, inefficient, or malicious? a self-safeguarding incentive mechanism for large scale-federated industrial maintenance based on double layer reinforcement learning. IEEE Internet of Things Journal, 11(11): 19988–20001. https://doi.org/10.1109/jiot.2024.3367875

[11] Liu, H., Li, S., Li, W., Sun, W. (2024). Efficient decentralized optimization for edge-enabled smart manufacturing: A federated learning-based framework. Future Generation Computer Systems, 157: 422-435. https://doi.org/10.1016/j.future.2024.03.043

[12] Putra, M.A.P., Zainudin, A., Sampedro, G.A., Utami, N.W., Kim, D.S., Lee, J.M. (2024). Collaborative decentralized learning for detecting bearing faults in industrial internet of things. In 2024 IEEE 29th Asia Pacific Conference on Communications (APCC), BALI, Indonesia, pp. 385-389. https://doi.org/10.1109/apcc62576.2024.10768063

[13] Praveena, K., Misba, M., Kaur, C., Al Ansari, M.S., Vuyyuru, V.A., Muthuperumal, S. (2024). Hybrid MLP-GRU federated learning framework for industrial predictive maintenance. In 2024 Third International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), Trichirappalli, India, pp. 1-8. https://doi.org/10.1109/iceeict61591.2024.10718600

[14] Guduri, N.V.R., Charaan, R.D., Pandi, S., Reddy, D.L. (2024). Predictive maintenance in IoT networks using a temporal attention mechanism based federated convolutional neural network. In 2024 4th Asian Conference on Innovation in Technology (ASIANCON), Pimari Chinchwad, India, pp. 1-7. https://doi.org/10.1109/asiancon62057.2024.10838193

[15] Lu, H., Thelen, A., Fink, O., Hu, C., Laflamme, S. (2024). Federated learning with uncertainty-based client clustering for fleet-wide fault diagnosis. Mechanical Systems and Signal Processing, 210: 111068. https://doi.org/10.1016/j.ymssp.2023.111068

[16] von Wahl, L., Heidenreich, N., Mitra, P., Nolting, M., Tempelmeier, N. (2024). Data disparity and temporal unavailability aware asynchronous federated learning for predictive maintenance on transportation fleets. In Proceedings of the AAAI Conference on Artificial Intelligence, 38(14): 15420-15428. https://doi.org/10.1609/aaai.v38i14.29467

[17] Sanchez, O.T., Borges, G., Raposo, D., Rodrigues, A., Boavida, F., Silva, J.S. (2025). Federated learning framework for LoRaWAN-enabled IIoT communication: A case study. IEEE Internet of Things Journal (Early Access). https://doi.org/10.1109/JIOT.2025.3558560

[18] Abdalah, R.W., Abdulateef, O.F., Hamad, A.H. (2025). A predictive maintenance system based on industrial internet of things for multimachine multiclass using deep neural network. Journal Européen des Systèmes Automatisés, 58(2): 373-381. https://doi.org/10.18280/jesa.580218

[19] Ali, M.I., Lai, N.S., Abdulla, R. (2024). Predictive maintenance of rotational machinery using deep learning. International Journal of Electrical and Computer Engineering (IJECE), 14(1): 1112-1121. http://doi.org/10.11591/ijece.v14i1.pp1112-1121

[20] Dalgkitsis, A., Koufakis, A., Stutterheim, J., Mifsud, A., Atwani, P., Gommans, L., Oprescu, A. (2024). Secure collaborative model training with dynamic federated learning in multi-domain environments. In SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis, Atlanta, GA, USA, pp. 755-759. https://doi.org/10.1109/scw63240.2024.00107

[21] Hosny, K.M., Magdi, A., Salah, A., El-Komy, O., Lashin, N.A. (2023). Internet of things applications using Raspberry-Pi: A survey. International Journal of Electrical & Computer Engineering, 13(1): 902-910. https://doi.org/10.11591/ijece.v13i1.pp902-910

[22] Hussien, H.A., Al-Messri, Z.A. (2023). Synthesis, characterization and evaluation of some Meldrum's acid derivatives as lubricant additives. Iraqi Journal of Science, 64(3): 1041-1048. https://doi.org/10.24996/ijs.2023.64.3.1

[23] Abood, R.H., Hamad, A.H. (2024). Multi-label diabetic retinopathy detection using transfer learning based convolutional neural network. Fusion: Practice and Applications, 17(2): 279293. https://doi.org/10.54216/fpa.170221

[24] Bharti, S., Mcgibney, A. (2021). Privacy-aware resource sharing in cross-device federated model training for collaborative predictive maintenance. IEEE Access, 9: 120367-120379. https://doi.org/10.1109/ACCESS.2021.3108839