ISSN: 2633-4828 Vol. 8 No.1, January, 2026

International Journal of Applied Engineering & Technology

BEARING FAULT CLASSIFICATION USING NETWORK-AWARE FEDERATED LEARNING IN HETEROGENEOUS IIOT NETWORKS

Bhupathirao Lakinana[1*] and Dr C Kalyana Chakravarthy[2]

[1] Assistant Professor, Department of Computer Science & Engineering - Artificial Intelligence, Vignan's Institute of Information Technology (Autonomous), Visakhapatnam, Andhra Pradesh

[2] Professor, Department of Computer Science & Engineering, MVGR college of Engineering (Autonomous), Vizianagaram, Andhra Pradesh

*Correspondence Author: E-mail: bhupathilakinana@gmail.com.

ABSTRACT

Accurately classifying distributed vibration signals of rotating machines under heterogeneous wireless network settings is necessary for diagnosis of bearing faults in Industrial Internet of Things (IIoT) systems. Centralizing raw sensor data results in excessive communication overhead and privacy issues, while traditional FL techniques involve in the participation of every client in each round without considering their network conditions, which results in significant latency, upload failures, and energy usage. In order to classify bearing faults, this paper proposes a Network-Aware Federated Learning framework that incorporates four mechanisms: (i) Using network parameters' including bandwidth, latency, packet loss, and local model accuracy, a Deep Q-Network (DQN) at the server automatically chooses the best clients in each round while protecting data privacy. (ii) FedProx regularization minimizes local model drift between clients to handle strong Non-Independent and Identically Distributed (Non-IID) data distributions. (iii) By balancing conflicting updates from various clients, server-side momentum further stabilizes global model convergence. (iv) Furthermore, Top-K gradient sparsification drastically reduces communication overhead and energy consumption by reducing the model upload size. Using 12 IIoT clients under strong Non-IID conditions and four heterogeneous network tiers ranging from GOOD to SEVERE, the proposed approach is evaluated using the Case Western Reserve University bearing dataset. Proposed FedProx+FedDQN outperform FedAvg and random client selection techniques to attain the maximum global diagnostic accuracy under strong Non-IID conditions. Furthermore, by combining DQN selection with Top-K sparsification, communication is reduced by around 47.49% while accuracy is only slightly compromised. In heterogeneous IIoT federated learning environments, the proposed approach demonstrates how network-aware client selection can enhance both diagnostic accuracy and communication efficiency. The framework automatically chooses reliable high-bandwidth clients for stable and effective fault detection while protecting data privacy by relying only on server-observable network parameters.

Keywords: *Predictive Maintenance, Reinforcement Learning, Deep Q- Networks, Federated Learning, Bearing Fault Diagnosis, Communication Efficiency, Heterogeneous IIoT Networks*

1. INTRODUCTION

About 40–50% of all electric motor failures are caused by bearings, which are the most important and often changed mechanical parts in rotating industrial machinery [1]. Because unexpected machine breakdowns can result in significant financial losses, reliable condition monitoring is crucial for predictive maintenance in modern industries. The most popular method for evaluating bearing health is vibration-based fault identification, which takes advantage of the distinctive frequency signatures created by flaws in the rolling parts, inner race, and outer race [2]. On centralized Case Western Reserve University benchmark datasets, recent deep learning techniques, particularly one-dimensional Convolutional Neural Networks (1D-CNNs) that directly employ raw signal data, have achieved extremely high fault classification accuracy [3]. However, implementing such models in actual Industrial Internet of Things (IIoT) settings presents basic difficulties that centralized methods are unable to handle. Vibration sensors are dispersed throughout several edge devices in IIoT deployments, each of which monitors a particular machine or production line. These sensors continuously produce high-frequency signals (12–48 kHz), which add up to terabytes of data every day [4]. There are three primary challenges when this raw

data is centralized on remote server. First, data privacy because vibration signals might reveal private information about machine functioning, product quality, and manufacturing procedures, industrial organizations need to safeguard their raw production data. Second, communication overhead, as it uses excessive bandwidth to continually broadcast high-frequency vibration data from numerous edge sensors via wireless channels, especially in constrained industrial wireless networks where spectrum must be shared with real-time control traffic. Third, network heterogeneity**,** because in an industrial environment, edge devices function under quite diverse wireless conditions based on their exposure to electromagnetic interference from industrial machinery, proximity to the access point, and physical location. Sensors in optimal locations can take advantage of 80–100 Mbps bandwidth with nearly-zero packet loss, while sensors behind metallic enclosures or away from the router struggle with 1–10 Mbps bandwidth, 400–700 ms latency, and 30–50% packet loss — making dependable centralized data collecting practically impossible.

McMahan et al. [5] presented Federated Learning (FL), which directly solves data privacy and communication overhead issues by facilitating cooperative model training across distributed edge devices without requiring the sharing of raw data. Each device in FL uploads only model parameters, not raw vibration signals, to a central aggregation server after training a local model on its private data. An updated global model is generated by the server using a weighted averaging rule, and it is subsequently sent to every device. These studies demonstrate that federated models are capable of achieving diagnostic accuracy on par with centralized methods.

Despite these developments, there are three major limitations in the current FL frameworks for IIoT defect diagnostics, which are the primary motivation for this work. The first limitation is network-agnostic participation. In every round, typical FL algorithms like FedAvg and FedProx engage all available clients without considering the state of their wireless networks. The model update may not reach the server when a client with a bad network connection participates with 40% packet loss, wasting both the client's energy and the server's communication opportunity. The client's battery still gets drained by the same unsuccessful transmission, even when the server receives a partial gradient set that impacts model quality. This is a critical inefficiency in IIoT contexts with limited resources where edge sensors run on batteries. The second limitation is unintelligent random client selection. While some studies use client sub-sampling to save communication costs, random selection is no more effective than chance. Wang et al. [6] showed that FL convergence can be improved under Non-IID conditions by using reinforcement learning-based client selection; however, their approach was not intended for network condition awareness, but rather for data diversity optimization, which involves choosing clients based on gradient diversity. It is insufficient to choose clients based just on data diversity without considering their existing network resilience in IIoT environments with diverse wireless conditions. The third limitation is Non-IID data degradation. In actual industrial settings, many machines experience different types of faults. When a bearing sensor is placed next to a centrifugal pump, it mostly detects Inner Race and Normal conditions; when it is placed close to a gearbox, it primarily detects Outer Race and Ball defects. This affects global model convergence under FedAvg aggregation by producing significant Non-IID distributions across clients, which lead local models to drift toward their own fault type distribution during training. Li et al. [7] demonstrated that FedProx's proximal regularization term successfully reduces this drift, increasing absolute test accuracy by an average of 22% in extremely heterogeneous situations.

No current FL paradigm for bearing fault diagnosis has concurrently addressed a combination of these three constraints: network heterogeneity, intelligent client selection, and non-IID data distribution. For IIoT bearing failure diagnostics, a single framework that simultaneously solves intelligent network-aware client selection, non-IID drift mitigation, and communication efficiency reduction is still unaddressed.

This paper proposes a **Network-Aware Federated Learning (NA-FedDQN) framework** for bearing fault classification that uses four combined techniques in order to fill this gap. The main contribution is a Deep Q-Network (DQN) client selection agent at the central server that learns to choose the best client subset each round based only on real-time observable network parameters - bandwidth, latency, packet loss, and local model accuracy through reward-driven experience replay. All of this is done without requiring any knowledge of client

data distributions, ensuring data privacy. This intelligent client selection is enhanced by: **FedProx proximal regularization** to limit local drift in strong Non-IID scenarios when only two of the four fault classes are held by each of the twelve clients; **Momentum on the server side** to smooth conflicting Non-IID updates and convergence stabilization; and By multiplicatively compounding the communication savings from DQN client selection, **Top-K gradient sparsification** with error feedback decreases per-client upload size by 60%, resulting in a total communication reduction of about 47.49% compared to the FedAvg baseline.

The rest of the paper is structured as follows: Section 2 summarizes related work in gradient compression, reinforcement learning-based client selection, and FL-based bearing fault diagnosis; Section 3 describes the suggested methodology; Section 4 presents experimental results and comparative analysis against eight FL baselines; Section 5 concludes the paper.

2. RELATED WORK

2.1. Bearing Fault Diagnosis Using Machine Learning Approaches

Conventional machine learning classifiers were used after manually created features were extracted from vibration signals in order to diagnose bearing faults. Yang et al. [8] demonstrated that SVM with appropriately built frequency characteristics performs dependable fault pattern recognition under stable operating conditions by applying SVM with IMF envelope spectrum features acquired through empirical mode decomposition for roller bearing fault diagnosis. Zhang et al. [9] a thorough analysis of machine learning techniques using CWRU data, benchmarking SVM, k-NN, decision trees, and random forests, and came to the conclusion that the shift in feature distribution between training and testing loads severely restricts the generalization of classical machine learning, necessitating the use of domain-adaptive techniques. Smith and Randall [10] established the CWRU benchmark dataset offers four fault health states at three severity levels across four operational loads, continues to be the primary evaluation foundation utilized in this work.

2.2. Bearing Fault Diagnosis Using Deep Learning Approaches

In the diagnosis of bearing faults, deep learning replaced manual feature engineering. WDCNN (Wide Deep Convolutional Neural Network), a 1D-CNN with wide first-layer kernels applied directly to raw vibration signals, was proposed by Zhang et al. [11]. The wide kernels extract features while suppressing high-frequency noise and AdaBN (Adaptive Batch Normalization) provides domain adaptation. Under normal conditions, WDCNN achieves 100% classification accuracy on CWRU and outperforms frequency-domain DNN models under various working loads and noisy environments. Because low-level features contain local fault characteristics that are not preserved in high-level abstract representations, Xu et al. [12] showed that combining CNN multi-level features with random forest ensemble classifiers improves accuracy and generalization when compared to using only the final CNN layer. In order to achieve enhanced robustness under variable speed settings, Fu et al. [13] suggested a parallel CNN-LSTM architecture in which the CNN branch records frequency-domain fault signals and the LSTM models temporal sequential patterns by jointly leveraging spatial and temporal vibration features. After reviewing more numbers of bearing fault diagnosis papers , Hakim et al. [14] confirmed that multi-scale 1D-CNN architectures represent the state of the art and identified cross-domain generalization across varying loads as the main unresolved issue driving federated learning approaches.

2.3. Federated Learning for Bearing Fault Diagnosis to Maintain Privacy

Despite their effectiveness, centralized approaches may result in increased communication costs and data privacy risks, particularly in data distributed sensor network environments. To address these limitations, recent research has increasingly focused on Federated Learning (FL), which enables multiple distributed clients to collaboratively train a global model without sharing their raw data. Federated Averaging (FedAvg), the fundamental federated learning algorithm, was first presented by McMahan et al. [5]. It demonstrated that without sharing raw data, merging locally trained models by periodic averaging can reach accuracy comparable to centralized training. In industrial contexts, centralizing raw vibration data presents serious privacy issues. Liu et al. [15] proposed a federated transfer learning scheme to alleviate the data island problem in industrial production while protecting

data privacy; experimental validation on two bearing datasets confirmed both diagnostic performance and privacy preservation. Yan et al. [16] similarly emphasized that federated learning effectively protects data privacy in fault diagnosis, proposing a repaired-data cross-device strategy using random forest regression to address data corruption and parameter negative transfer, with experimental verification across three bearing data sets demonstrating both diagnostic effectiveness and privacy protection. Lu et al. [17] integrated biometric authentication into distributed industrial environments ensures a privacy preserving framework where sensitive raw datasets remain localized on edge devices.

2.4. Federated Learning for Bearing Fault Diagnosis to Handle Non-IID

Sensitivity to non-IID data distributions was found to be a key drawback in Federated Learning. FedProx, an extension of FedAvg with a proximal regularization term that limits local model deviation from the global model during training, was proposed by Li et al. [7]. FedProx uses proximal regularization to enhance federated learning under statistical and system heterogeneity. Zhou et al. [18] address non-IID data challenges in federated learning by measuring local data imbalance to optimize model updates before global aggregation. This framework maintains high system accuracy despite significant data heterogeneity across devices. Geng et al. [19] proposed FA-FedAvg, an improved federated learning algorithm for bearing fault diagnosis that gives higher weight to better-performing local models and uses a precision-difference based aggregation strategy to speed up convergence under Non-IID data. Deng et al.[20] proposed the Decoupled Federated Fault Diagnosis (DecFFD) framework to resolve the cross-location Non-IID problem unique to multi-site industrial deployments. This personalized federated learning framework for cross-location fault diagnosis separates global and personalized features during local training. This mechanism enables robust bearing fault diagnosis by preventing localized manufacturing noises from degrading global diagnostic performance.

2.5. Federated Learning to Reduce Communication Cost for Bearing Fault Classification

One of the main FL bottlenecks for IIoT edge devices is communication overhead. For IIoT fault diagnosis, the SQC-FLMGAN framework [21] combined Top-K sparsification and quantization compression in FL. The Top-K algorithm chooses important model parameters for transmission while quantization further reduces their precision. This combined compression is integrated with a multi-agent GAN that produces a variety of training samples to mitigate Non-IID distribution, achieving simultaneous communication reduction and diagnostic accuracy improvement. Du et al. [22] demonstrated that communication costs can be significantly decreased while maintaining diagnostic performance across bearing, gearbox, and bogie datasets by proposing Lightweight FL for mechanical fault diagnosis using non-structural pruning and lightweight training methodologies at the local network level.

2.6. Federated Learning Under Heterogeneous Network Conditions

IIoT factory installations are characterized by heterogeneous wireless network conditions, which include fluctuating bandwidth, latency, and packet loss. Selecting clients without considering their real time network conditions leads to persistent upload failures, straggler – induced round delays and wasted transmission energy, directly degrading both model convergence quality and communication efficiency in resource constrained industrial environments. Chen et al.[23] formulated client selection as a joint optimization of overall training latency and model convergence, deriving optimal client selection probabilities and number of participants by considering only training latency and data heterogeneity among clients. For FL over wireless networks without prior channel state information, Xia et al. [24] proposed update-importance-based and latency-based client scheduling. They showed that scheduling based on channel quality significantly reduces overall training time without degrading accuracy; the latency-based scheme specifically addresses the straggler problem caused by wireless heterogeneity. However, Chen et al. [23] and Xia et al.[24] consider only heterogeneity in latency without accounting for bandwidth and packet loss constraints. Using an experience-driven contextual bandit approach, Wang et al. [25] demonstrated reinforcement learning-based client selection in Non-IID FL, demonstrating that RL-based policies learned from experience outperform random selection and establishing the use of reinforcement learning for intelligent client selection that inspired this work's DQN approach.

To the best of the knowledge, no existing work has applied network-aware client selection-considering bandwidth, latency and packet loss simultaneously-to federated bearing fault diagnosis. Therefore this paper proposes the Network aware FedDQN frame work jointly considers bandwidth, latency, and packet loss for network-aware client selection while simultaneously addressing Non-IID data drift, communication cost, energy consumption-without compromising global bearing fault diagnosis accuracy.

3. METHODOLOGY

3.1. Overview of system Architecture

The proposed architecture uses 12 IIoT edge enabled clients, each of which connected to individual industrial rotating machines, to implement a FL-based bearing fault diagnosis system. The overall system consists of three functional layers as shown in Fig. 1. The layer1 consists of 12 edge enabled IIoT clients with accelerometer sensors to collect vibration data. Each client act as independent data silo and there is no communication between clients. The second layer is heterogeneous wireless communication layer that simulates the realistic industrial wireless network environment, in which each client simulates with dynamically changing bandwidth, transmission delay, and packet loss because of industrial electromagnetic interference, channel fading, and network traffic congestion. The third layer is a Federated server that contains the global model evaluation module, DQN agent for client selection and aggregation process.

3.2. Dataset Description and Non-IID Data Partitioning

3.2.1. *CWRU Bearing Dataset*

The Case Western Reserve University (CWRU) bearing fault dataset is used in this study. At four motor load conditions—0, 1, 2, and 3 horsepower (HP) with motor speeds of 1797, 1772, 1750, and 1730 RPM, respectively—signals are recorded from the drive-end accelerometer. The dataset covers the **Normal** (no defect), **Inner Race Fault (IF)**, **Ball Fault (BF)**, and **Outer Race Fault (OF) states** with fault diameters—0.007", 0.014", and 0.021" inches.

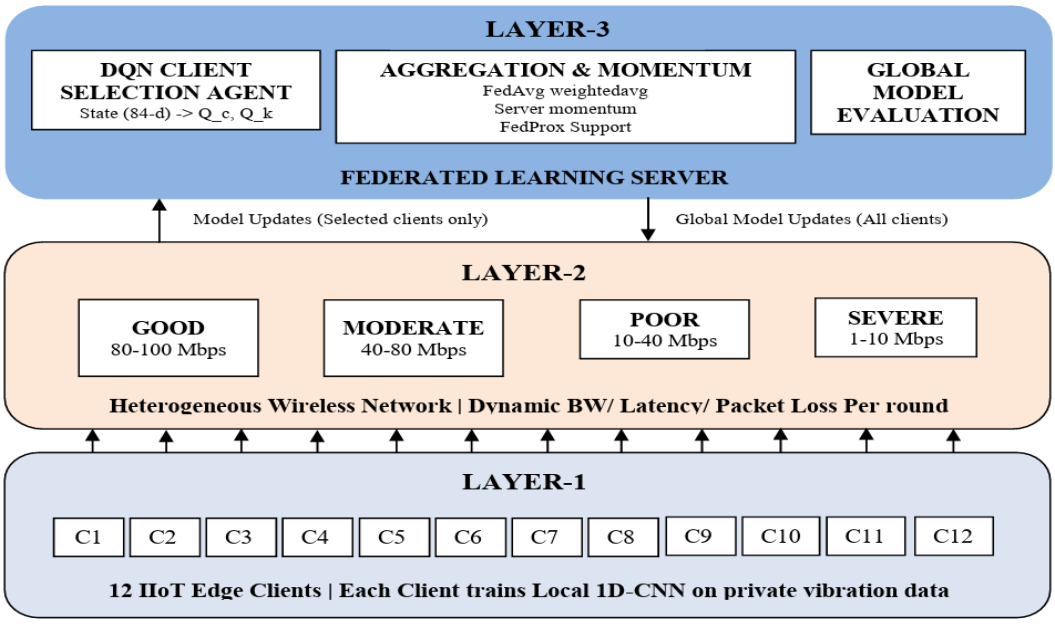


Fig. 1. Proposed NA-FedDQN Three-Layer System Architecture: IIoT Edge Clients, Heterogeneous Wireless Network, and Federated Learning Server.

3.2.2. *Vibration Signal Preprocessing*

All data files contain raw vibration signals. Using a non-overlapping, sliding window approach each vibration signal is divided into fixed-length segments with each segment size of L = 1024 samples.. Every segment is normalized using the per-sample z-score:

$$x_i = \frac{(x_i - \mu_i)}{(\sigma_i + \varepsilon)} \quad (1)$$

where μ_i and σ_i represent the segment mean and standard deviation respectively, and $\varepsilon = 10^{-8}$ is a small constant for numerical stability.

3.2.3. *Strong Non-IID Creation*

A basic difficulty in FL is the Non-IID (Non-Independent and Identically Distributed) nature of client data in actual deployments. In actuality, various industrial machines are susceptible to various fault modes based on their age, maintenance history, and operating role. In order to model the Non-IID partitioning realistically, every client is assigned with exactly two faults out of four faults.

3.2.4. *Train-Test Split and Server Test Set*

To ensure repeatability, an 80:20 stratified train-test split is applied individually at each client. This 80% training data is only used for Local model training. Importantly, the central server gathers the 20% test split from each of the 12 machines to create a global balanced test set. All four fault classes, all three fault severity levels (0.007", 0.014", and 0.021"), and all four operating conditions (0–3 HP) are covered by this server-side global test set. This server-side test set is the only one used to assess global model accuracy at each round, offering a really objective global performance metric that is unaffected by the Non-IID distribution at client nodes.

3.3. Local Model Training using 1D CNN Architecture

Fig 2 illustrates the architecture of one-dimensional Convolutional Neural Network (1D-CNN). It is trained on each of the twelve clients' local vibration segment. The 1D-CNN directly learns both temporal and spatial hierarchical features from raw time-domain signals, in contrast to traditional machine learning methods that require human feature engineering. This makes it especially appropriate for deployment on IIoT edge devices with limited resources. Using 16 filters with a kernel size of 7 and padding of 3, the first convolutional layer finds basic overall characteristics while maintaining the temporal dimension. Non-linearity is introduced by a ReLU activation, and the feature map is reduced from 1024 to 256-time steps by max-pooling with stride 4. In order to capture mid-frequency fault harmonics, the second layer uses a narrower kernel (size 5) and doubles the filter count to 32. The third convolutional layer extracts high-level abstract representations using 64 filters with kernel size 3 after a second max-pooling operation. This is followed by global adaptive average pooling, which completely collapses the temporal dimension to a 64-dimensional feature vector regardless of input length. In order to avoid overfitting on tiny client datasets, the fully connected classifier uses a 64-unit linear layer with ReLU activation, followed by Dropout (p = 0.25). This is crucial because each client has data from just two fault classes. The four output logits that correspond to the four fault classes are mapped to the final linear layer.

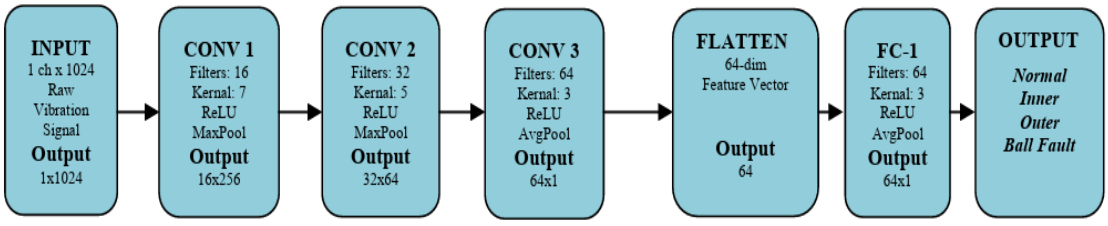


Fig. 2. One Dimensional CNN Architecture for Local Training

Using the Adam optimizer with learning rate lr = 1e-3, batch size B = 64, and E = 15 local epochs each FL round, local training maximizes the cross-entropy loss. In strong Non-IID settings, the learning rate is carefully selected to avoid excessive local drift. For a batch of size B, the cross-entropy loss is:

$$L_{CE} = -\left(\frac{1}{B}\right)\sum_{b=1}^{B}\sum_{c=1}^{4} y_{\{b,c\}} \log p_{\{b,c\}} \quad (2)$$

Where $p_{\{b,c\}}$ is the expected probability for class c of sample b and $y_{\{b,c\}}$ is the one-hot label.

3.4. Heterogeneous IIoT Network Model

Inorder to provide a realistic heterogeneous network model between all edge clients, four network tiers are assigned across 12 clients based on three wireless parameters –bandwidth, latency and packet loss. The client assigned with GOOD tier operates at 80-100 Mbps bandwidth, 5-30ms latency, and 0-5% packet loss representing sensors with direct line-of-sight to the access point. The clients assigned with MODERATE tier operates at 40-80 Mbps bandwidth, 30-150ms latency, and 5-15% packet loss representing sensors at intermediate distance with occasional interference. The POOR tier operates at 10-40 Mbps bandwidth, 150-400ms latency, and 15-30% packet loss, representing sensors partially obstructed by industrial machinery. The SEVERE tier operates at 1-10 Mbps bandwidth, 400-700ms latency, and 30-50% packet loss, representing sensors behind metallic enclosures or in high electromagnetic interference zones. Each client is placed in one of four network quality levels that correspond to its unique communication environment in order to accurately model this. For every client i at round t, three network parameters are modeled: packet loss PL_i (ratio), latency LAT_i (ms), and bandwidth BW_i (Mbps). The transmission cost and dependability of each client's model upload are directly determined by these.

3.4.1. *Transmission Time Model*

The total of the data transfer time and the network propagation delay is used to represent the upload transmission time for client i:

$$T_i^{tx} = \frac{(M_{bytes}*8)}{(BW_i*10^6)} + \frac{LAT_i}{1000} \quad (3)$$

where the model size in bytes is represented by M_{bytes}, and bytes are converted to bits by the ratio of 8.

3.4.2. *Energy Consumption Model*

Client i's energy usage during model upload is modeled as:

$$E_i = P_{tx} * T_i^{tx} \quad (4)$$

where the nominal wireless transmission power is P_{tx}= 0.5 W. This is in line with standard IIoT sensor WiFi transmit power levels.

3.4.3. *Network Quality Score*

For every client, a composite network quality score NQ_i∈ [0, 1] is calculated by averaging and normalizing the three network characteristics:

$$NQ_i = \frac{\left(\frac{BW_i}{100} + \left(\frac{1-LAT_i}{700}\right) + (1-PL_i)\right)}{3} \quad (5)$$

Better network conditions are indicated by higher scores, which combine the effects of latency, bandwidth, and dependability into a single scalar. Each client reports NQ_i as part of its metadata, which serves as the DQN reward function's main input.

3.5. Federated Learning Aggregation

3.5.1. *Pure FedAvg Aggregation*

The weighted federated averaging aggregation formula, which was first put forth by McMahan et al. [2], is used by all six assessed techniques:

$$\theta^{t+1} = \sum_{i \in S_t} \left[\frac{n_i}{\sum_{j \in S_t} n_j} \right] * \theta_i^t \quad (6)$$

where n_i is the number of local training samples at client i,θ_i^t are the local model parameters of client i following E epochs of local training, and S_t is the set of clients chosen at round t. In order to ensure a fair comparison across approaches with different k values, a key design decision in this work is that only the selected clients S_t contribute to aggregation, and communication cost and energy are tracked exclusively for selected clients.

3.5.2. *Server-Side Momentum (FedAvgM)*

When the chosen client subset shifts, subsequent aggregation rounds may result in contradictory global model updates under strong Non-IID conditions. As a result, instead of smoothly converging, the global model trajectory oscillates. In order to overcome this, we use server-side momentum (FedAvgM), which smooths the global model update by applying a momentum buffer:

$$\theta^{t+1} = \theta^t + \beta * \left(\theta_{agg}^{t+1} - \theta^t \right) \quad (7)$$

where β = 0.9 is the server momentum coefficient and θ_{agg}^{t+1} is the raw aggregated model from equation (6).

3.5.3. *FedProx Local Training*

Each client can freely optimize its local objective for E epochs with standard FedAvg training. This cause's considerable client drift under strong Non-IID conditions, where each client only has two of the four fault classes: local models significantly deviate from the global model in the direction of their own local data distribution. These divergent updates largely cancel each other out when combined, which slows convergence and lowers final accuracy. In order to prevent client drift, FedProx [3] adds a proximal term that penalizes deviations from the current global model to each client's local training objective:

$$h_i(\theta) = F_i(\theta) + \left(\frac{\mu}{2} \right) * \left\| \theta - \theta_{global} \right\|^2 \quad (8)$$

where μ = 0.5 is the proximal regularization coefficient, θ_{global} is the global model obtained from the server at the beginning of round t, and $F_i(\theta)$ is the standard cross-entropy loss on local data. During each gradient update step, the proximal term (μ/2) $\left\| \theta - \theta_{global} \right\|^2$ acts as an elastic restriction that pulls the local model back toward the global model, avoiding excessive drift while yet permitting significant local adaptation.

3.6. Network-Aware DQN Client Selection

This paper's main contribution is a Deep Q-Network (DQN) agent at the central server that, without any prior knowledge of client data distributions or network topology, learns an ideal client selection policy from experience. Client selection is formulated by the DQN as a Markov result Process (MDP) in which the action chooses the best client subset, the state records all clients' current network conditions, and the reward generates a scalar feedback signal that quantifies the quality of the selection result.

3.6.1. *State Representation*

The server creates an 84-dimensional state vector at each round t by concatenating seven observable features from each of the 12 clients after they have finished local training and submitted their model parameters:

$$S(t) = [S_0, S_1, \ldots\ldots, S_{11}] \in R^{84} \quad (9)$$

where the feature vector for client *i* is:

$$S_i = \left[acc_i, \frac{n_i}{1000}, \frac{BW_i}{100}, \frac{1 - LAT_i}{700}, 1 - PL_i, NQ_i, 1.0 \right] \quad (10)$$

The characteristics include: (i) the client's local model quality is reflected in the local validation accuracy after training (acc_i); (ii) the size of the normalized dataset (n_i/1000); (iii) the normalized bandwidth (BW_i/100) ∈ [0,1];

(iv) the inverted normalized latency ($1-LAT_i/700$) ∈ [0,1], where higher values indicate lower latency; (v) the packet delivery ratio ($1-PL_i$) ∈ [0,1]; (vi) the composite network quality score NQ_i from equation (5); and (vii) a constant connectivity indicator (1.0). To guarantee stable DQN training, all features are normalized to [0,1].

3.6.2. Action Space

Together, the DQN decides which particular clients to include in the aggregation round and how many clients to choose (k). The definition of the action space is:

$$k \in K = \{6, 7, 8, 9, 10, 11, 12\} \quad (11)$$

In order to maintain global fault class coverage during aggregation, the minimum k = 6 guarantees that at least one client from each of the six Non-IID class-pair groups is likely to be chosen. When all clients have favorable network conditions, the maximum k = 12 permits complete participation. The DQN uses two specialized output heads: a k-selection head ($Q_k \in \Psi^{7)}$ that scores every potential subset size and a client scoring head ($Q_c \in \Psi^{12}$) that assigns a Q-value to each unique client. The chosen set S_t during exploitation is made up of the top-k clients by Q_c score and $k = \text{argmax}(Q_k)$.

3.6.3. *DQN Architecture*

After processing the 84-dimensional state vector using a shared feature extraction module, the policy network Q_policy divides into two output heads: Client scoring head (Q_c) and k-selection head (Q_k). Fig 3 shows the DQN multi head DQN architecture to select k and k-clients.

3.6.4. Reward Function:

Using only quantities observable at the server without accessing client data, the DQN reward function is intended to concurrently reward high diagnostic accuracy, effective network use, consistent client selection, and low upload failure rates:

$$R(t) = \alpha_1 \Delta Acc(t) + \alpha_2 NQ(t) + \alpha_3 Eff(t) - \alpha_4 Osc(t) - \alpha_5 Fail(t) \quad (12)$$

Using a dynamic baseline instead of a fixed threshold, the **accuracy improvement term** $\Delta Acc(t) = \max(0, acc(t) - acc(t-1))$ rewards the DQN for round-over-round improvements in global model accuracy. This guarantees that even when accuracy hits a high plateau, the DQN will still get a meaningful gradient signal. The main objective of network-aware selection is reflected in the **network quality term** NQ(t), which has the highest weight $\alpha_2 = 0.50$. Higher NQ(t) results in better bandwidth, lower latency, and lower packet loss for the chosen clients, which immediately lowers upload failures and communication costs. The **efficiency term** Eff(t) encourages choosing fewer clients, hence increasing communication efficiency.

$$Eff(t) = \left(1 - \frac{(k - MIN_k}{(Max_K - MIN_K)}\right) \quad (13)$$

The **oscillation penalty** Osc(t) promotes selection stability and quicker DQN convergence by discouraging random selection changes between rounds. The primary mechanism by which the DQN outperforms random selection on network metrics is the **failure penalty** Fail(t), which explicitly penalizes the DQN for choosing clients whose uploads fail, encouraging avoidance of high-PL clients.

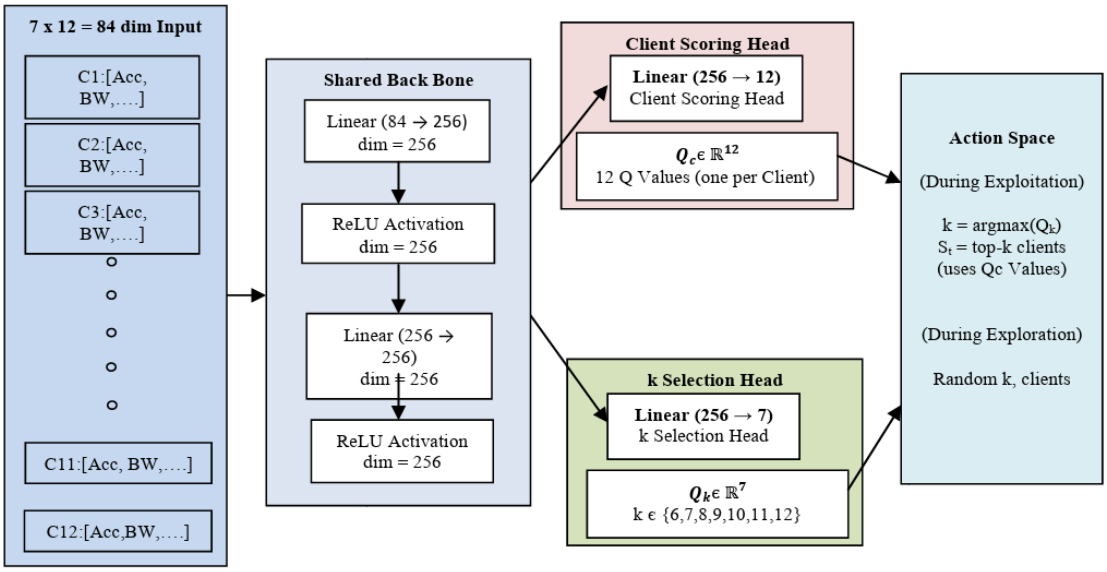


Fig.3. DQN Client Selection Architecture: Shared backbone with dual output heads for simultaneous client scoring ($Q_c \in \mathbb{R}^7$) and subset-size selection ($Q_k \in \mathbb{R}^{12}$).

4. RESULTS AND DISCUSSION

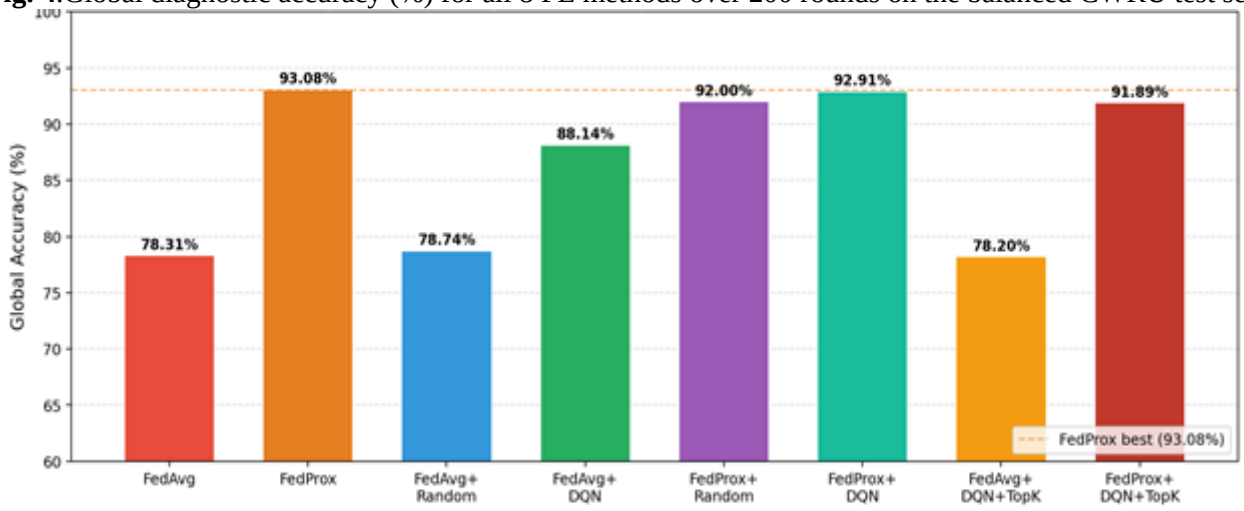
4.1. E*xperimental Setup*

All experiments are carried out using the Flower (flwr) federated learning framework. Over T = 200 communication rounds, the system simulates 12 IIoT edge clients functioning in heterogeneous wireless network settings. After each round, the server uses the balanced test set to assess global model accuracy.

4.2. Global Accuracy Comparison

The overall diagnostic accuracy attained by all eight approaches is shown in Fig 4. The base line approach FedAvg (with all

Fig. 4. Global diagnostic accuracy (%) for all 8 FL methods over 200 rounds on the balanced CWRU test set.



clients) achieve accuracy 78.31% under strong Non-IID setup, while FedProx (with all clients) improves accuracy to 93.08% by constraining local model drift through proximal regularization. FedAvg+FedDQN achieve 88.14% and FedProx+DQN achieve 92.91%, where 1% reduction compared to base line approach FedProx (93.08%). The

results confirm that intelligent client selection maintains competitive diagnostic accuracy with negligible trade off, demonstrating that fewer but reliable clients are sufficient for effective global model training.

4.2.1. DQN vs FedRandom

In order to validate the network-awareness of the proposed DQN, it is crucial to compare techniques that have the same k ranges but different selection methodologies. FedAvg+FedDQN (88.14%) performs 9.4 percentage points better than FedAvg+FedRandom (78.74%). Likewise, FedProx+FedDQN (92.91%) performs 0.91 percentage points better than FedProx+FedRandom (92%).This performance difference can only be attributed to the network-aware selection intelligence of the DQN agent because both pairings employ similar aggregation formulae, identical CNN architectures, and identical k ranges [6,12].

4.2.2. Effect of FedProx

Fig 5 clearly demonstrates, in the strong Non-IID environment, FedProx-based techniques consistently perform better than their FedAvg-based counterparts. The proximal regularization term ($\mu = 0.5$) is responsible for a gain of 4.76 percentage points when FedProx+FedDQN (92.91%) is compared to FedAvg+FedDQN (88.14%). This demonstrates that the proximal constraint successfully reduces local model drift when clients only have two of the four fault classes.

4.2.3. *Effect of Top-K Compression on proposed FedProx+FedDQN:*

The compression technique FedProx+FedDQN+TopK (91.89%) achieve accuracy within 1.02 percentage points of their non-compressed equivalent, demonstrating that the accuracy reduction caused by Top-K sparsification with error feedback is minor. In order to maintain convergence quality, the error feedback system makes sure that discarded gradient components are retrieved in two to three subsequent rounds.

4

.3. Convergence Analysis

4.3.1. Accuracy Progression

The global accuracy convergence curves for each of the six approaches over 200 rounds are displayed in Fig. 5.

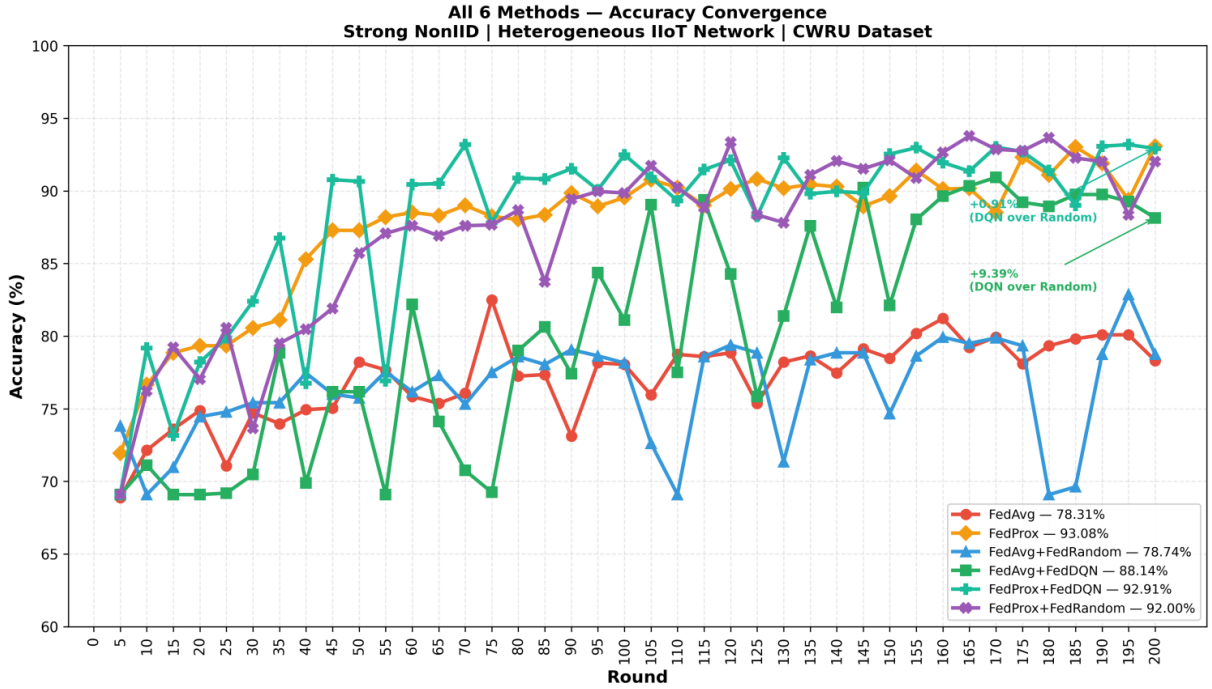


Fig. 5. Round By Round Accuracy of All six methods

The ε-greedy exploration schedule is consistent with the typical two-phase convergence behavior of all DQN-based methods: an exploration phase (rounds 1–170) where accuracy oscillates as the DQN explores the client selection space, and an exploitation phase (rounds 170–200) where the DQN converges to a stable selection

policy and accuracy stabilizes. During the exploration phase, DQN methods exhibit accuracy oscillations comparable to FedRandom, as random selection dominates. After round 170, DQN methods consistently select GOOD and MODERATE tier clients, resulting in fewer upload failures and more reliable gradient aggregation — producing the stable accuracy trend with minimal oscillation as visible in Fig. 5.

4.3.2. DQN Convergence

After gaining enough replay buffer experiences, the DQN policy network starts training at round 50 (MIN_REPLAY = 50). The shift from oscillating to monotonically increasing accuracy shows that the Q-value estimations stabilize by around round 100. The epsilon decay curve and the k-selection trajectory are displayed in Fig. 6, which confirms that after round 130, k converges to 7, the ideal subset size found by the DQN via reward-driven learning.

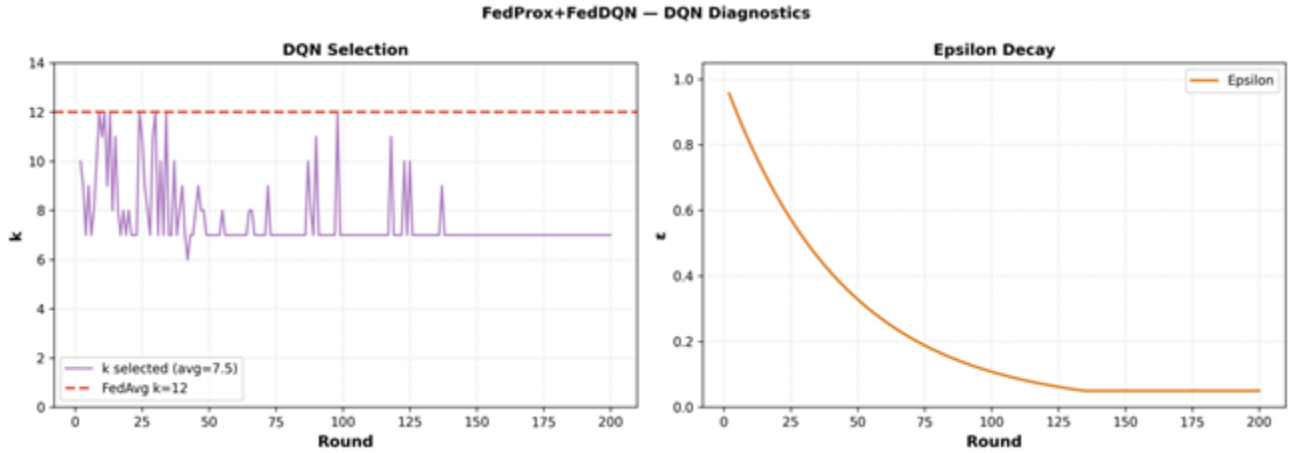


Fig. 6. DQN Diagnostics for FedProx+FedDQN: (left) k selection trajectory Vs FedAvg baseline k=12; (right) epsilon decay schedule over 200 rounds.

4.4. *Per-Class Fault Diagnosis Accuracy*

Per-class accuracy for each of the four fault categories is shown in Table I. Since each client stores data from only two of the four fault classes, achieving balanced performance across all four classes is very difficult under strong Non-IID conditions. To create a comprehensive diagnostic capacity, the global model must compile complementary local knowledge.

TABLE I: PER-CLASS ACCURACY (%)

Method	Normal	Inner	Outer	Ball
FedAvg	100	78.47	1.74	79.51
FedProx	**100**	**76.74**	**85.07**	**93.4**
FedAvg+FedRandom	100	63.89	0	98.61
FedAvg+FedDQN	100	91.32	92.36	39.58
FedProx+FedRandom	100	93.75	89.24	65.28
FedProx+FedDQN	**100**	**88.19**	**100**	**65.97**
FedAvg+DQN+TopK	100	66.66	92.36	0
FedProx+DQN+TopK	**100**	**84.37**	**96.52**	**66.66**

FedAvg has a severe class imbalance; compared to the FedProx. This illustrates the client drift issue, which occurs when clients with complimentary class data tug the global model in opposing directions during aggregation. The greatest balanced per-class performance is achieved by FedProx+FedDQN: Normal=100%, Inner=88.19%, Outer=100%, Ball=65.97%. The DQN implicitly promotes balanced class representation in each aggregation round by choosing clients with complementary fault class coverage, even though this is accomplished without knowing the client data distributions.

Because Inner Race Fault signatures show the most distinctive frequency patterns in the CWRU dataset at the drive-end accelerometer site, the Inner Race Fault class consistently obtains the highest individual accuracy across all approaches. The FedProx proximal restriction, which keeps local models from over-specializing on their designated 2-class subsets, is especially beneficial for Ball Fault and Outer Race Fault, which generate more delicate vibration patterns.

4.5. Network Parameter Analysis

The network efficiency metrics for each approach are shown in Table II. Comparisons of techniques with different k values are directly related to the selection strategy because these metrics are only calculated over the chosen client subset S_t at each round.

TABLE II: NETWORK EFFICIENCY METRICS

Method	Lat (ms)	BW (Mbps)	Waste (MB)	Upload Failuers	k avg
FedAvg	234.48	45.28	23.66	443	12
FedProx	234.48	45.28	24.08	451	12
FedAvg+FedRandom	230.27	45.62	17.94	336	8.8
FedAvg+FedDQN	209.24	51.24	13.94	261	7.5
FedProx+FedRandom	234.25	45.6	18.69	350	9
FedProx+FedDQN	208.25	51.36	14.15	265	7.5
FedAvg+DQN+TopK	208.73	51.37	10.8	223	7.5
FedProx+DQN+TopK	182.74	53.31	9.74	228	7.5

4.4.1. *Latency*

FedAvg and FedProx engage all 12 clients, including three SEVERE tier clients with latencies of 400–700 ms, resulting in an average round latency of 234.48 ms. By learning to avoid SEVERE tier clients, FedProx+FedDQN reduces this to 208.25 ms, a 11.2% reduction. Because FedRandom occasionally chooses GOOD tier clients by chance, it achieves intermediate latency (234.25 ms), but it is unable to continuously avoid SEVERE and POOR tier clients like the DQN does after round 150.

4.4.2. Bandwidth Utilization

Compared to FedAvg (45.28 Mbps) and FedRandom (45.6 Mbps), FedProx+FedDQN obtains an average bandwidth of 51.36 Mbps across a subset of clients. The tier selection frequency analysis (Fig. 8) shows that the DQN has a learnt preference for GOOD tier clients (80–100 Mbps) and MODERATE tier clients (40–80 Mbps). FedProx+FedDQN selects fewer SEVERE and POOR tier clients during the exploitation phase (rounds 170–200) when compare to Random selection.

4.4.3. Upload Failures

The number of gradient updates available for aggregation each round is directly decreased by upload failures, which lowers the quality of the model. FedAvg (2.21 failures/round) and FedRandom (1.75 failures/round) have high failure rates than FedProx+FedDQN (1.32 failures/round). In comparison to FedRandom, the DQN's failure

penalty term directly encourages avoiding high-PL clients, which reduces upload failures by 24.3%. This is the clearest quantitative proof that the DQN has acquired network-aware behavior.

4.4.4. Wasted Communication

As illustrated in Fig 7, FedProx+FedRandom (18.7 MB total) has significantly more wasted communication (bytes from unsuccessful uploads) than FedProx+FedDQN (14.2 MB total), a decrease of 24.1%. Unsuccessful uploads use bandwidth and transmission energy without adding to the aggregation of models. This waste is immediately reduced by the DQN's learnt avoidance of high-PL clients, which makes the framework energy and communication efficient.

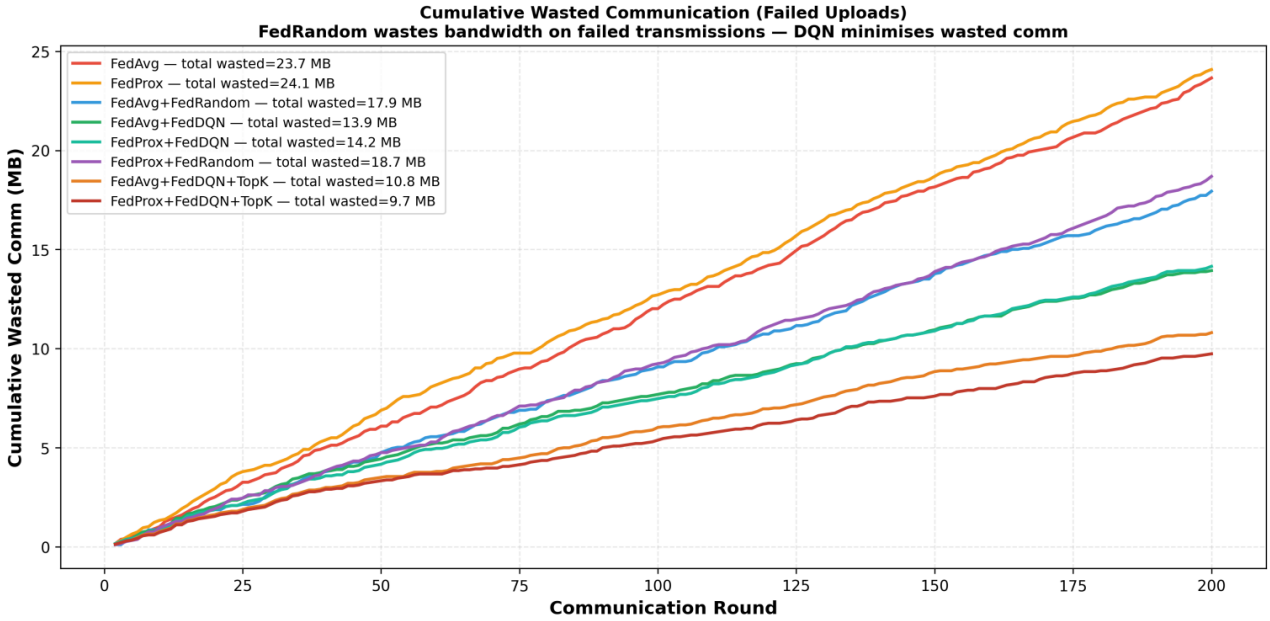


Fig. 7. Cummulative Wasted communication. DQN-based methods reduce both total and wasted bytes by avoiding high-PL POOR and SEVER tier clients.

4.4.5. Tier Selection Evolution:

The FedProx+FedDQN tier selection frequency heatmap and early vs. late bar chart are shown in Fig. 8.

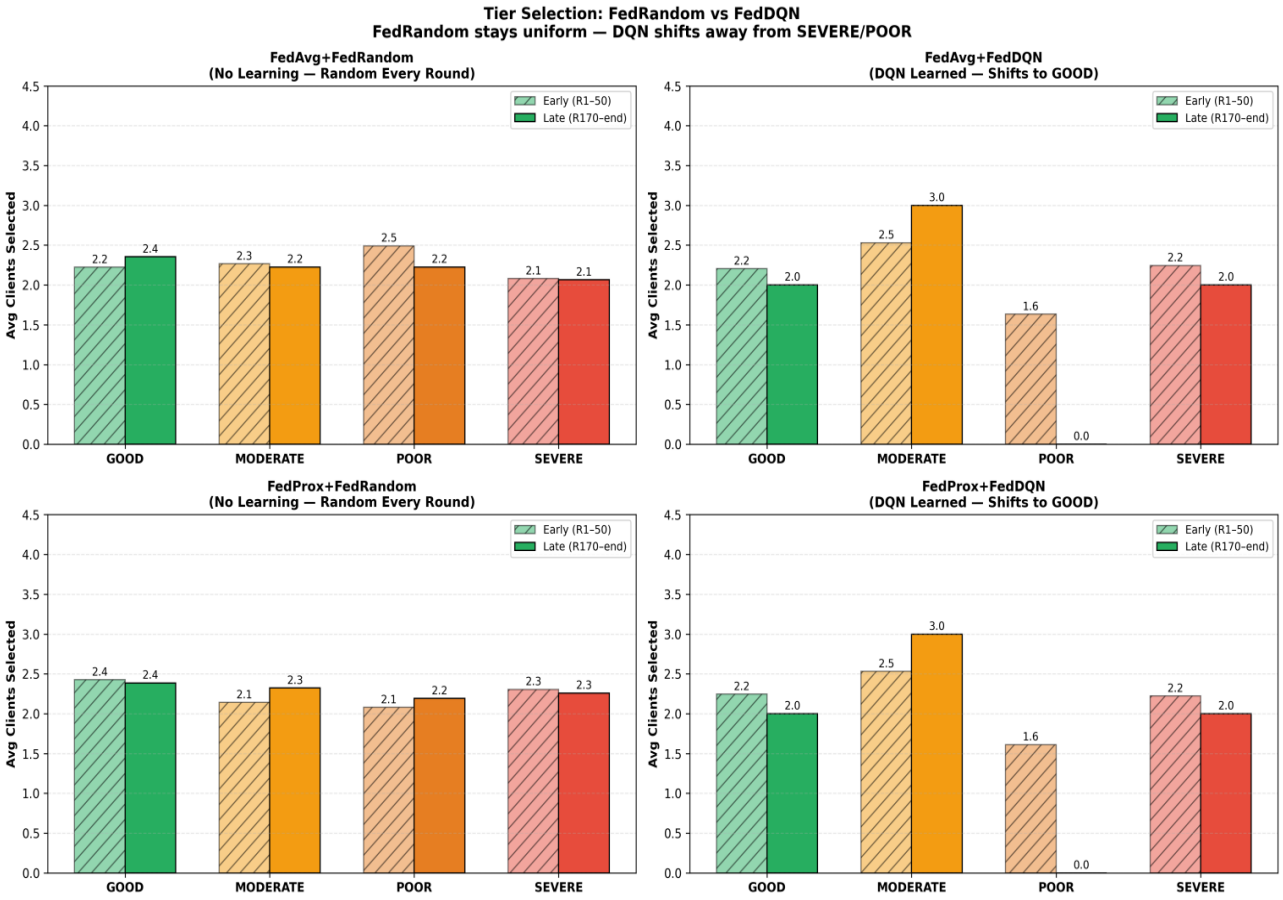


Fig. 8. Tier selection evolution for FedProx+FedDQN: early exploration (rounds 1-50) near-uniform; late exploitation (rounds 170-200) strongly favors GOOD and MODERATE tiers.

The selection distribution across tiers is almost uniform in the early exploration phase (rounds 1–50) (GOOD: ~2.2, MODERATE: ~2., POOR: ~1.6, SEVERE: ~2.2 average clients/round), which is consistent with random ε-greedy exploration. The distribution significantly swings toward high-quality tiers (GOOD: ~2.0, MODERATE: ~3.0, POOR: ~0.0, SEVERE: ~2.0 average clients/round) throughout the exploitation phase (rounds 170–200). The fundamental evidence of the DQN's network-aware intelligence is the fact that FedRandom, on the other hand, keeps a flat distribution for all 200 rounds, demonstrating the lack of any learning.

4.5. Communication Cost and Compression Analysis

The communication costs of various approaches are compared in Table III. FedAvg serves as the baseline, sending the complete models of all 12 clients each round for a total of 103.8 MB across 200 rounds.

Table III: Communication Cost Comparison

Method	Comm (MB)	vs FedAvg (%)	vs DQN (%)	Acc (%)
FedAvg	*103.8*	—	—	*78.31*
FedAvg+FedDQN	*66.3*	*−36.1%*	—	*88.14*
FedAvg+DQN+TopK	*53.3*	*−48.7%*	*−19.6%*	*78.20*
FedProx+FedDQN	*66.0*	*−36.4%*	—	92.91
FedProx+DQN+TopK	*54.3*	*−47.7%*	*−17.7%*	*91.89*

4.5.1. DQN Selection Savings

By choosing only k=7.5 clients each round on average rather than all 12, FedProx+FedDQN lowers overall communication costs by 36.4% in comparison to FedAvg. Gradient compression is not used to accomplish these savings; instead, intelligent client selection is used.

4.5.2. Combined DQN + Compression Savings

Communication costs are further reduced by FedProx+FedDQN+TopK by 17.7% compared to FedProx+FedDQN and by 47.7% compared to the FedAvg baseline. This is the result of the combined multiplicative effects of (i) DQN client selection, which reduces the number of participating clients from 12 to k=7.5; and (ii) Top-K compression, which reduces the upload size of each selected client by 60%. This compression's accuracy cost of just 1.02 percentage points demonstrates how well the error feedback technique preserves gradient information.

4.6. Energy Efficiency

Equation 4 states that energy usage is exactly proportional to a transmission time. FedAvg engages SEVERE and POOR tier

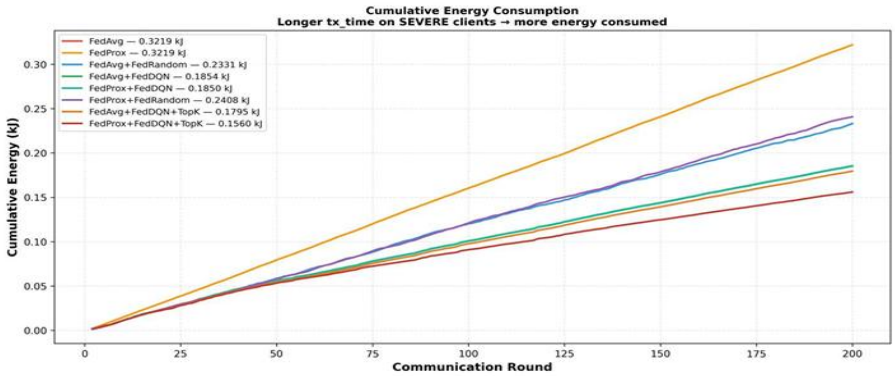


Fig. 9. Cummulative Energy Consumption comparision of All 6 methods

clients with long transmission times at each round, consuming 0.3219 kJ over 200 rounds. By choosing fewer clients and favoring high-bandwidth clients with faster transmission times, FedProx+FedDQN lower energy usage by 42.5% from 0.3219 kJ to 0.1850 kJ. Because Top-K compression lowers upload bytes, which in turn lowers transmission time and energy per chosen client, FedProx+FedDQN+TopK achieves the lowest energy usage of 0.1560 kJ—a 51.5% reduction relative to FedAvg. Fig 9 shows the cumulative energy consumption for all 8 methods.

5. CONCLUSION

This paper proposed NA-FedDQN, a network-aware federated learning system for classifying bearing faults in diverse IIoT settings. To jointly address network heterogeneity, Non-IID data drift, and communication overhead, the framework incorporates a Deep Q-Network (DQN) client selection agent, FedProx proximal regularization ($\mu = 0.5$), server-side momentum ($\beta = 0.9$), and Top-K gradient sparsification (ratio = 0.40) with error feedback. In order to provide true privacy preservation, the DQN chooses clients based only on visible network parameters, such as bandwidth, latency, packet loss, and local accuracy, even without knowledge of client data distributions. Evaluated over 200 FL rounds on 12 heterogeneous IIoT clients using the CWRU bearing fault dataset, FedProx+FedDQN attains 92.91% global diagnostic accuracy, exceeding FedAvg by 14.6 percentage points and FedProx+FedRandom by 0.91%. In contrast to random selection, the DQN minimizes upload failures by 24.28%, latency by 11.11%, and energy usage by 23.17%, with lower accuracy oscillations in the exploitation phase (rounds 170–200). FedProx+FedDQN+TopK attains 47.49% overall reduction of communication in comparison to FedAvg (103.4 MB → 54.3 MB) with negligible accuracy penalty. These findings results demonstrates when real-time network awareness is used to influence client selection, communication efficiency and diagnostic accuracy are complementing goals. To further the framework's suitability in Industry 4.0 settings, future research will examine adaptive compression ratios, actual hardware implementation on IIoT edge devices, and hierarchical FL topologies.

REFERENCES

[1] Nandi, Subhasis., Toliyat, Hamid A. & Li, Xiaodong. (2005). "*Condition monitoring and fault diagnosis of electrical motors—A review*," **IEEE Transactions on Energy Conversion**, 20(4), 719–729. DOI: 10.1109/TEC.2005.847955.

[2] Hakim, Mohammed., Omran, Ahmed A. B., Ahmed, Ahmed N., Al-Waily, Mohammed. & Abdellatif, Ahmed. (2023). "*A systematic review of rolling bearing fault diagnoses based on deep learning and transfer learning: Taxonomy, overview, application, open challenges, weaknesses and recommendations,"* **Ain Shams Engineering Journal**, 14(4), 101945. DOI: 10.1016/j.asej.2022.101945.

[3] Zhao, Zhibin., Li, Tianfu., Wu, Jun., Sun, Cheng., Wang, Shibin., Yan, Ruqiang. & Chen, Xuefeng. (2020). "*Deep learning algorithms for rotating machinery intelligent diagnosis: An open source benchmark study*", **ISA Transactions**, 107, 224–255. DOI:10.1016/j.isatra.2020.08.010.

[4] Xu, Li Da., He, Wu. & Li, Shancang. (2014). "*Internet of Things in industries: A survey*," **IEEE Transactions on Industrial Informatics**, 10(4), 2233–2243. DOI: 10.1109/TII.2014.2300753.

[5] McMahan, Brendan., Moore, Eider., Ramage, Daniel., Hampson, Seth. & Arcas, Blaise Agüera y. (2017). "*Communi-cation efficient learning of deep networks from decentralized data*", **Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)**, 1273–1282. http://proceedings.mlr.press/v54/mcmah an17a.html.

[6] Wang, Hongyi., Kaplan, Zachary., Niu, Dong. & Li, Baochun. (2020). "*Optimizing federated learning on Non-IID data with reinforcement learning*," **Proceedings of IEEE INFOCOM 2020**, 1698–1707. DOI:10.1109/INFOCOM41043.20 20.9155494.

[7] Li, Tian., Sahu, Anit Kumar., Zaheer, Manzil., Sanjabi, Maziar., Smola, Alexander. & Smith, Virginia. (2020). "*Federated optimization in heterogeneous networks,"* **Proceedings of the 3rd Conference on Machine Learning and Systems (MLSys)**. https://proceedings.mlsys.org/paper/2020/hash/1f5fe83998a09396ebe6477d9475ba0c-Abstract.html

[8] Y. Yang, D. Yu, and J. Cheng, "A fault diagnosis approach for roller bearing based on IMF envelope spectrum and SVM," Measurement, vol. 40, no. 9, pp. 943–950, 2007, DOI:10.1016/j.measurement.2006.10.010.

[9] X. Zhang, B. Zhao, and Y. Lin, "Machine learning based bearing fault diagnosis using the Case Western Reserve University data: A review," IEEE Access, vol. 9, pp. 155598–155608, 2021, DOI: 10.1109/ACCESS.2021.3128669.

[10] W. A. Smith and R. B. Randall, "Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study," Mech. Syst. Signal Process., vol. 64, pp. 100–131, Dec. 2015, DOI:10.1016/j.ymssp.2015.04. 021.

[11] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," Sensors, vol. 17, no. 2, p. 425, Feb. 2017, DOI:10.3390 /s17020425.

[12] G. Xu, M. Liu, Z. Jiang, D. Söffker, and W. Shen, "Bearing fault diagnosis method based on deep convolutional neural network and random forest ensemble learning," Sensors, vol. 19, no. 5, p. 1088, Mar. 2019, doi: 10.3390/s19051088.

[13] G. Fu, Q. Wei, and Y. Yang, "Bearing fault diagnosis with parallel CNN and LSTM," Math. Biosci. Eng., vol. 21, no. 2, pp. 2385–2406, Jan. 2024, DOI: 10.3934/mbe.2024105.

[14] M. Hakim, A. A. B. Omran, A. N. Ahmed, M. Al-Waily, and A. Abdellatif, "A systematic review of rolling bearing fault diagnoses based on deep learning and transfer learning," Ain Shams Eng. J., vol. 14, no. 4, p. 101945, Apr. 2023, doi: 10.1016/j.asej.2022.101945.

[15] L.Liu, Z.Yan, T. Zhang, Z. Gao,H. Cai, and J.Wang, “Data Privacy protection: A novel federated transfer learning scheme for bearing fault diagnosis,” Knowledge-Based Systems, vol.291, p.111587, May 2024, DOI:10.1013/j. knosys.2024.111587.

[16] Z. Yan, J. Sun, Y.Zhang,L. Liu,Z.Gao, and Y. Chang, “Federated transfer learning strategy: A novel cross-device fault diagnosis method based on repaired data,” Sensors, vol.23,no. 16,p.7302, Aug. 2023, DOI:10.3390/s23167302.

[17] S. Lu, Z. Gao, Q. Xu, C. Jiang, A. Zhang, and X. Wang, "Class-imbalance privacy preserving federated learning for decentralized fault diagnosis with biometric authentication," IEEE Trans. Ind. Inform., vol. 18, no. 12, pp. 9101–9111, Dec. 2022, DOI:10.1109/TII.2022.3190034.

[18] F. Zhou, Y. Yang, C. Wang, and X. Hu, "Federated learning based fault diagnosis driven by intra-client imbalance degree," Entropy, vol. 25, no. 4, p. 606, Apr. 2023, DOI: 10.3390/e25040606.

[19] D. Geng, H. He, X. Lan, and C. Liu, "Bearing fault diagnosis based on improved federated learning algorithm," Computing, vol. 104, pp. 1–19, Jan. 2022, DOI: 10.1007/s00607-021-01019-4.

[20] D.Deng, W. Zhao, X.Wu, T. Zhang,J.Zheng, J. kang, and D. Niyato, “DecFFD: A personalized Federated learning Framework for Cross-Location Fault Daignosis,” IEEE Transactions on Industrial Informatic, Vol. 20, no. 5, pp. 7802-7091, May 2024, DOI:10.1109/TII.2024.3353920.

[21] X. Sun, Z. Yuan, X. Kong, L. Xue,H. Cheng and, Z. Chen, "An efficient federated learning framework for enhancing data diversity and communication in IIoT fault diagnosis," IEEE Internet Things Journal, vol 12, no. 17, pp. 36562-36576, September 2025, DOI: 10.1109/JIOT.2025.3583081.

[22] J. Du, N. Qin, D. Huang, X. Jia, and Y. Zhang, "Lightweight FL: A low-cost federated learning framework for mechanical fault diagnosis with training optimisation and model pruning," IEEE Transactions on Instruments and Measurement. Pp(99):1-1, January 2023, doi:10.1109/TIM.2023.3328073.

[23] X. Chen, X. Zhou, H. Zhang, M. Sun, and H. V. Poor, "Client selection for wireless federated learning with data and latency heterogeneity," IEEE Internet Things J., vol. 11, no. 19, 2024, doi:10.1109/JIOT.2024.3425757.

[24] W. Xia, W. Wen, K. Wong, T. Quek, J Zhang and H. Zhu, "Federated learning-based client scheduling for low-latency wireless communications," IEEE Wireless Communications. vol. 28, no. 2, pp. 32–38, Apr. 2021, DOI:10.1109/ MWC.001.2000252.

[25] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on Non-IID data with reinforcement learning," in Proc. IEEE INFOCOM, Jul. 2020, pp. 1698–1707, https://doi.org/10.1109/INFOCOM41043.2020.9155494.