electronics

MDPI

Article

FedLSTM: A Federated Learning Framework for Sensor Fault Detection in Wireless Sensor Networks

Rehan Khan [1], Umer Saeed [2] and Insoo Koo [1,*]

[1] Department of Electrical Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, Republic of Korea; rehan77@mail.ulsan.ac.kr
[2] School of Computer Science and Technology, University of Bedfordshire, Luton LU1 3JU, UK; umer.saeed@beds.ac.uk
* Correspondence: iskoo@ulsan.ac.kr

Abstract: The rapid growth of Internet of Things (IoT) devices has significantly increased reliance on sensor-generated data, which are essential to a wide range of systems and services. Wireless sensor networks (WSNs), crucial to this ecosystem, are often deployed in diverse and challenging environments, making them susceptible to faults such as software bugs, communication breakdowns, and hardware malfunctions. These issues can compromise data accuracy, stability, and reliability, ultimately jeopardizing system security. While advanced sensor fault detection methods in WSNs leverage a machine learning approach to achieve high accuracy, they typically rely on centralized learning, and face scalability and privacy challenges, especially when transferring large volumes of data. In our experimental setup, we employ a decentralized approach using federated learning with long short-term memory (FedLSTM) for sensor fault detection in WSNs, thereby preserving client privacy. This study utilizes temperature data enhanced with synthetic sensor data to simulate various common sensor faults: bias, drift, spike, erratic, stuck, and data-loss. We evaluate the performance of FedLSTM against the centralized approach based on accuracy, precision, sensitivity, and F1-score. Additionally, we analyze the impacts of varying the client participation rates and the number of local training epochs. In federated learning environments, comparative analysis with established models like the one-dimensional convolutional neural network and multilayer perceptron demonstrate the promising results of FedLSTM in maintaining client privacy while reducing communication overheads and the server load.

Keywords: federated learning; fault detection; wireless sensor networks; IoT; machine learning; data privacy

check for updates

Citation: Khan, R.; Saeed, U.; Koo, I. FedLSTM: A Federated Learning Framework for Sensor Fault Detection in Wireless Sensor Networks. *Electronics* **2024**, *13*, 4907. https://doi.org/10.3390/electronics13244907

Academic Editors: Paul Mitchell

Received: 18 November 2024
Revised: 9 December 2024
Accepted: 11 December 2024
Published: 12 December 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The Internet of Things (IoT) is a vast network of interconnected smart devices that continuously generate and collect substantial volumes of data [1]. This technology has become deeply integrated in almost every facet of modern life, spanning smart appliances and entertainment systems in our homes to invisible infrastructure in smart cities, including pedestrian and road sensors [2]. The applications extend even further into areas like smart grids, autonomous vehicles, healthcare equipment, industrial control systems, and robotics, demonstrating the expansive impact of the IoT. Figure 1 illustrates the wide-ranging influence of IoT.

The immense data gathered by these devices necessitate robust storage and advanced analytical capabilities. Machine learning (ML) and deep learning (DL) approaches have been employed to derive insights from these data, enabling intelligent applications [3]. Detecting various sensor faults in wireless sensor networks (WSNs) is a critical application within intelligent IoT systems. With the continuous rise in anomalies across IoT infrastructures [4], effectively monitoring this vast data volume for anomaly detection has become

essential. Automated techniques based on ML and DL offer promising solutions to address these challenges [5].

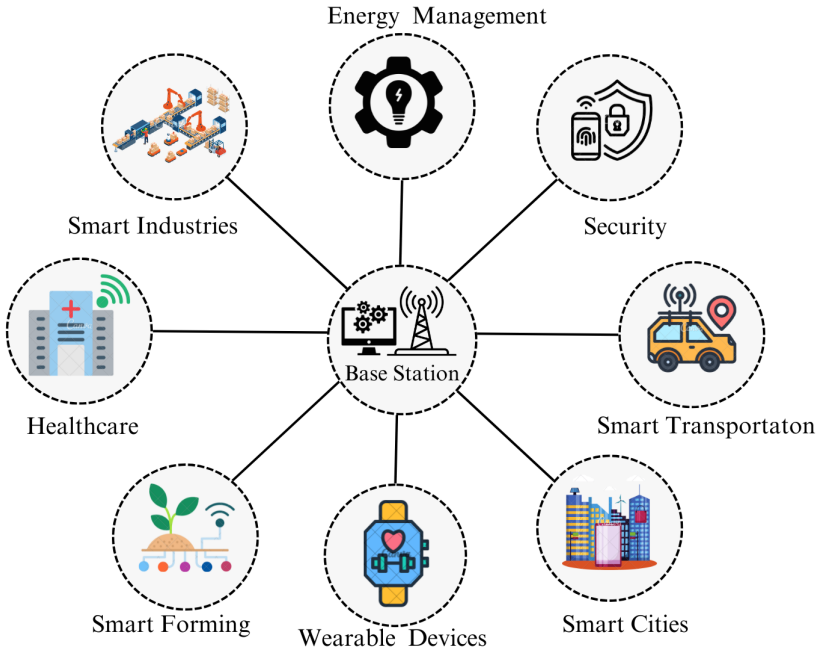


Figure 1. IoT applications connected to a central base station in various smart environments.

DL, a subset of ML, has gained widespread popularity across various fields, including science, engineering, healthcare, and finance [6]. DL has also become increasingly valuable for sensor fault detection because it enables more sophisticated analysis of WSNs, and offers more precise anomaly detection than traditional ML methods. In highly complex environments with massive amounts of data, DL models generally outperform ML models in terms of accuracy and performance, although they require greater computational power [7]. Despite their differences, both ML and DL traditionally follow a centralized approach requiring the data and the models to be stored in a single location [8].

In a centralized setup, data are gathered from various remote sources and transmitted to a central repository where the data are processed and prepared for classification. This proves effective in settings confined to a single organization or where locations are situated within the same geographic area [9]. However, for organizations with multiple sites or collaborative setups, centralized data transfers can introduce privacy concerns. Another challenge is the volume of data collected from WSNs sensors because transferring large datasets to a central base station for classification can strain network resources and create significant bottlenecks [10]. Additionally, centralizing data transfers may impair the functionality of certain IoT applications that are latency sensitive [11].

Federated learning (FL) is the ML concept introduced by Google in 2017 [12] to address challenges such as data privacy and network limitations. FL provides a distributed environment in which individual clients analyze their own data locally, eliminating the need for direct data transfer. Instead, clients collaborate by sharing a common model that they train on their own local data, while the server functions as an aggregator, only making model parameter exchanges. The server coordinates training across clients, aggregating their updates through algorithms such as federated averaging (FedAvg) until the training process is complete [13].

IoT networks are inherently distributed, and FL provides an effective solution to the limitations of centralized approaches for sensor fault detection in WSNs, as it allows monitoring in the immediate vicinity of the sensor. However, FL-based approaches to sensor fault detection remain in the developmental stages, requiring extensive evaluation and refinement to ensure viability for real-world applications [14].

In this study, we evaluate the performance of the proposed Federated Long Short-Term Memory (FedLSTM) model, where eight distributed clients collaboratively classify sensor faults in both single-hop and multi-hop scenarios with WSNs datasets. The aim is to assess FL as a viable alternative to the traditional DL approach, which relies on centralized data storage and analysis at a single site. In this federated system, horizontal data partitioning is applied, allowing each client to work with distinct data samples that share the identical feature space, ensuring that no data sharing occurs between clients. This setup prioritizes user privacy and security, making it suitable for a variety of sensitive environments. We then compare the results of the proposed FedLSTM model with those obtained from a centralized approach. We compare the performance of FedLSTM with other architectures, including multilayer perceptron (MLP) and one-dimensional convolutional neural network (1D-CNN) models.

The main contributions of this study are as follows:

- We present a novel FL framework, FedLSTM, specifically designed for sensor fault detection in WSNs. Unlike centralized learning methods that require transferring both features and labels, FedLSTM facilitates distributed model training by transmitting only model weights to the central server. This approach significantly reduces the data transfer overhead, enhancing system efficiency.
- This study utilizes publicly available datasets containing normal temperature readings. Since no public datasets currently include sensor fault data, we address this limitation by synthetically simulating and including a range of common sensor faults in WSNs, such as bias, drift, spike, erratic, stuck, and data-loss faults, enabling more robust model testing and validation.
- A comparative analysis between FedLSTM and centralized models highlights the security benefits of our approach. In FedLSTM, only model parameters are shared with the server, ensuring that user data, including features and labels, remain confidential. This is particularly valuable for sensitive applications such as healthcare, commercial operations, and military settings, where privacy concerns, especially around location data, are critical.
- We compare FedLSTM to other models, including MLP and the 1D-CNN. Findings demonstrate that FedLSTM performs better in the distributed learning context, surpassing these models in sensor fault detection accuracy.
- Extensive experimentation with WSNs datasets shows that FedLSTM achieves performance levels comparable to centralized models, even under the added constraint of protecting data privacy. This underscores the practicality of our approach for real-world industrial applications where privacy is paramount.

The remainder of this article is organized as follows: Section 2 reviews the relevant research on fault detection within IoT environments using FL, providing a contextual background. Section 3 presents our proposed FL method for sensor fault detection, covering data acquisition, preparation, and the methodology. Section 4 outlines the performance metrics and discusses the results. Finally, Section 5 concludes the article by summarizing the key findings.

2. Literature Review

Integration of the IoT into industrial systems has enabled greater automation and increased economic benefits, including enhanced productivity and efficiency. Recent advancements in IoT-based services reflect a growing trend of adoption across various industries [15]. A crucial factor in maintaining high-quality service in these modern IoT systems is the reliability of the data transmitted by sensors. In real-world scenarios, sensors are often deployed in harsh environments, such as extreme industrial or natural conditions, to monitor various physical parameters [16]. Low-cost sensors, often deployed in challenging environments, are susceptible to various factors over time, including aging, hardware defects, improper installation, and adverse environmental conditions such as excessive humidity or extreme temperatures. These issues can result in data anomalies, which under-

mine the accuracy and reliability of IoT systems in critical applications [17]. Since many IoT applications rely on sensor data for control and decision-making, undetected defective data can significantly compromise system accuracy and reliability, potentially leading to severe consequences [18]. In earlier decades, fault detection and diagnosis in industries faced challenges due to system complexity, which made parameter determination difficult. However, in recent years, numerous data-driven anomaly detection approaches have been developed and effectively applied across a range of industrial applications [19–21].

DL and ML [22], which enable the creation of multilayer data representations, have been widely employed in fault diagnosis, particularly within sensor networks, where faults can lead to severe consequences. The early detection of sensor faults in WSNs is crucial for ensuring safety, improving data quality, enhancing network security, extending network lifespan, and reducing response times. As WSNs have become more prevalent, research has increasingly focused on fault detection and identification to improve network safety and reliability [23]. For instance, a distributed, online approach utilizing a support vector machine model was proposed in [12] to detect anomalies within WSNs. The authors in [24] utilized Extremely Randomized Trees, an advanced ML technique, that showed promise in diagnosing sensor faults, including hardover, drift, spike, erratic, data-loss, stuck, and random faults, by outperforming traditional methods in accuracy and precision while reducing training time.

Researchers have developed several feature extraction techniques aimed at automatic anomaly detection. By transforming sensor data into time–frequency domain representations, CNNs are employed to classify sensor faults across different applications, including drone signals and aero-engine control systems [25,26]. In recent years, generative adversarial networks (GANs) have emerged as powerful techniques for detecting sensor faults. By generating virtual models of physical sensors, GANs, in combination with Gramian Angular Field image encoding, preserve temporal correlations in sensor data, allowing for more accurate fault detection [27]. Centralized intelligent anomaly detection techniques, which rely on collecting data from mobile devices and transmitting them to a cloud server or data center for processing and training, present several limitations. This centralized approach often struggles to improve the model's generalization and robustness, while simultaneously imposing significant communication burdens and computational overhead on the system [28]. Privacy and security concerns further complicate this setup with certain edge devices unwilling to share private data with a central server for training, fearing data leakage involving sensitive user information.

Recently, federated learning (FL), a distributed learning paradigm, has emerged as a promising solution to address the pressing challenges of privacy protection and data security inherent in centralized systems [29]. FL enables multiple clients to train models locally, sharing only model updates to achieve learning objectives without exchanging raw data, which significantly enhances data privacy. Concurrently, edge computing alleviates the computational burden on cloud centers [30] by performing data processing, providing storage, and installing applications directly at the network edge (i.e., the data source), reducing the need for cloud-based transmission. Recently, FL algorithms, such as federated averaging [31] have efficiently reduced communication costs by aggregating client model parameters at a central server, which then distributes the averaged updates to the clients. This method has been widely used in practice, including in applications such as traffic monitoring [32], healthcare [33], and smart applications [34], demonstrating the feasibility and efficiency of integrating FedAvg with deep neural networks.

Additionally, FL has advanced fault diagnosis across various domains. In [35], the authors introduced the Whitening-Net framework, which emphasizes domain generalization to detect machine faults under diverse conditions based on causal mechanisms. In the manufacturing sector, the authors of [36] developed an FL-based defect detection framework that distributes model training across nodes, thereby reducing communication costs.

In [37], the authors enhanced data privacy by training local models on client devices and securely aggregating them using a federated transfer learning model to further im-

prove learning efficiency. The authors in [38] achieved higher performance from fault datasets, particularly bearing faults, by combining FL with deep transfer networks. In [39], the authors addressed the need for real-time fault diagnosis updates by asynchronously aggregating client parameters through a linear fusion method, effectively mitigating network delays. To address data heterogeneity and to enhance diagnostic accuracy while reducing computational complexity, the authors [40] proposed a clustered FL framework for wind turbine diagnostics. This framework groups clients based on canonical correlation coefficients. Similarly, the authors in [41] employed a federated CNN model to detect faults in HVAC systems, maintaining data privacy while achieving high accuracy across fault levels in air handling units and chillers.

In WSNs where sensor nodes are dispersed across large areas, centralized fault detection faces challenges due to communication latency and power constraints. In [42], the authors proposed an FL-based framework for sensor fault detection in WSNs, enabling each node to independently train a local model while sharing only the model parameters. This approach effectively mitigates the communication overhead and enhances fault detection accuracy in resource-constrained environments. In [43], the authors integrated shallow artificial neural networks with aggregation algorithms such as FedAvg and FedAdam for intrusion detection, demonstrating the ability of FL to balance privacy and efficiency in IoT systems. Similarly, in [44,45], the authors proposed FL strategies that optimize communications protocols and reduce bandwidth in wireless networks, ensuring that energy constraints in sensor networks do not hinder the FL process.

Centralized models, though widely utilized for anomaly detection in sensor networks, face significant limitations that reduce their applicability in real-world IoT environments. These systems require transmitting raw data to a central server for processing, resulting in substantial communication overhead, particularly in resource-constrained scenarios such as WSNs. Furthermore, centralized models pose considerable privacy and security risks, as sensitive data are vulnerable to breaches during transmission. The reliance on a single server also introduces scalability challenges as IoT networks expand, while restricting the model's ability to generalize effectively across diverse sensor environments. In contrast, the proposed FL model decentralizes the training process by enabling edge devices to perform local training and share only model updates. This approach significantly reduces the communication overhead, enhances data privacy by eliminating the need for raw data transmission, and improves model generalization by accommodating the heterogeneity of sensor environments. By employing techniques such as federated averaging, the proposed FL framework ensures scalability, adaptability, and efficient fault detection in real-time scenarios. These advancements demonstrate the superiority of the FL-based approach in addressing critical challenges related to privacy, scalability, and efficiency, establishing it as a robust and effective solution for sensor fault detection in IoT systems.

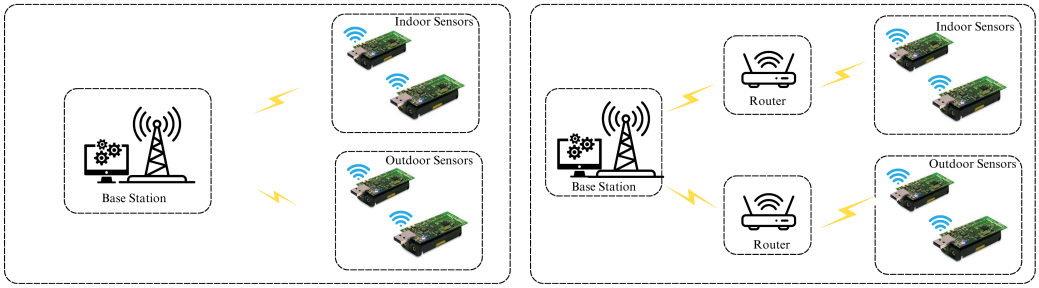
3. Materials and Methods

In this section, we outline an effective method for detecting sensor faults in WSNs. The process begins with data acquisition from open-source datasets, followed by addressing the simulation of synthetic common sensor faults in normal temperature sensor data because there are no such data available that offer faulty sensor readings that reflect real-world conditions. We then introduce the FedLSTM framework, an FL adaptation of LSTM networks that facilitates distributed training across sensor nodes. This approach not only preserves data privacy but enhances the robustness of the model.

3.1. Data Acquisition

The experiments utilized open-source datasets developed by research teams at the University of North Carolina at Greensboro, where single-hop and multi-hop WSNs scenarios were created by gathering temperature and humidity readings with Telos-B motes [46]. Data collection employed TinyOS-based sensor nodes, taking humidity and temperature measurements in both indoor and outdoor environments. In the single-hop setup, two

indoor and two outdoor sensors simultaneously collect temperature and humidity data, transmitting them directly to a central base station as shown in Figure 2a. By contrast, the multi-hop network covers a broader area, resulting in an extensive WSN architecture, as illustrated in Figure 2b. Given the limited transmission range of each sensor node, intermediary routers were activated to relay data from the nodes to the base station.



(**a**) (**b**)

Figure 2. Communication setups in IoT-based wireless sensor networks: (**a**) single-hop and (**b**) multi-hop scenario.

To manage data flow, two indoor and two outdoor sensors were deployed, generating data packets containing 10 readings each that were subsequently transmitted to the base station. Data were collected over a six-hour period with a sampling frequency of one reading every five seconds. However, there are no publicly available databases that contain defective sensor readings. To address this gap in sensor fault detection research, we introduced various common WSNs sensor faults by synthetically injecting faults into normal temperature sensor readings, a widely adopted approach among researchers [47,48]. Using the Conditional Generative Adversarial Network (GAN) model, synthetic samples were generated, and artificial sensor faults were injected into the normal temperature sensor data. Conditional GANs extend traditional GANs by incorporating additional input, such as class labels, to guide both the generator and discriminator in producing data specific to designated categories. This supervised learning enhancement enables Conditional GANs to generate class-specific synthetic samples that closely align with the desired distributions, making them particularly effective for addressing the scarcity of labeled fault data in sensor fault detection research.

3.2. Synthetic Fault Injection

Sensor faults are considered deviations from the expected or normal operational output of a sensor. Depending on the circumstances, these faults can be intermittent, persistent, or temporary. Common causes include network congestion, complex operating conditions, poor manufacturing, and aging. Faults are typically categorized based on their deviations from normal behavior. Mathematically, a normally functioning sensor can be described as shown in Equation (1):

$$S_n = f(t) + \eta \tag{1}$$

where $f(t)$ represents the sensor output at time t, while η denotes noise. Ideally, the signal S_n should equal $f(t)$; this study examines six common types of sensor faults as described below, which were derived by modifying the baseline equation.

3.2.1. Bias Fault

A bias fault occurs when a constant value is added to the sensor output, which, as a result, consistently deviates from a normal reading. This type of fault often arises from hardware issues, including short circuits or calibration malfunctions, within the sensing unit. A bias fault can be expressed mathematically as follows:

$$S_n = f(t) + \eta + \alpha, \quad \alpha = \text{constant} \tag{2}$$

3.2.2. Drift Fault

A drift fault is a common sensor malfunction, where the sensor output steadily increases at a constant rate over time. Let $i(t)$ represent the incremental bias added to the output signal at time t, modeled as $i(t) = i(t-1) + \alpha$. Causes of drift faults include physical damage from harsh environmental conditions or the corrosion of sensor material over time. The drift fault can be mathematically represented by Equation (3):

$$S_n = f(t) + \eta + i(t), \quad i(t) = i(t-1) + \alpha, \quad \alpha = \text{constant} \tag{3}$$

3.2.3. Spike Fault

Spike faults, as the name suggests, involve high-amplitude deviations that occur intermittently and at regular intervals in the sensor's output signal. These faults are introduced by periodically adding a constant bias, denoted by η, to the nth element of the non-faulty signal $f(t)$, thereby generating a spike fault represented as S_n. In this context, $\alpha(t)$ accounts for additional fluctuations or noise introduced to the faulty signal at time t. Here, $n = (1, 2, 3, \ldots.)$ represents natural numbers, and $\tau \geq 2$ is a positive integer that defines the periodic interval between successive faults. Such faults often result from vibrations or disturbances from other system components affecting the sensor. The mathematical model of a spike fault is Equation (4):

$$S_n = f(t) + \eta + \alpha(t); \quad \forall t = n \times \tau, f(t) + \eta, \quad \text{otherwise}, \quad n = \{1, 2, 3, \ldots\}, \quad \tau \geq 2 \tag{4}$$

3.2.4. Erratic Fault

An erratic fault introduces high-variance, zero-mean noise into the sensor's output signal, and typically arises due to physical damage, high-frequency noise within the system, or loose circuitry connections. Equation (5) is a mathematical expression of the erratic fault:

$$S_n = f(t) + \eta + \alpha\alpha \sim N(0, \delta_\alpha^2)\delta_\alpha^2 \gg \delta_\eta^2 \tag{5}$$

Here, S_n represents the sensor's output signal affected by an erratic fault, with $f(t)$ denoting the underlying non-faulty signal as a function of time t. The parameter η accounts for a constant bias introduced by minor drift in the signal. The erratic noise, represented by α, is modeled as a Gaussian distribution with a mean of zero and variance δ_α^2, expressed as $\alpha \sim N(0, \delta_\alpha^2)$. This formulation captures the high-variance and stochastic nature of erratic faults, where the noise α introduces significant deviations in the sensor's output.

3.2.5. Stuck Fault

In the stuck fault, the sensor output fluctuates by zero or almost zero, and undergoes no variation, effectively remaining constant (often referred to as complete failure). This fault can be either persistent or transient; nevertheless, it is detrimental to the system in both situations, and requires timely detection. Short circuits (disconnects in the circuitry or degradation of sensor materials) are common causes. Mathematically, a stuck fault is represented as

$$S_n = \alpha, \quad \alpha = \text{constant} \tag{6}$$

3.2.6. Data-Loss Fault

A data-loss fault occurs when sensor readings contain missing values within the series from a specific node. Such faults can arise due to hardware malfunction, poor calibration, or battery failure.

Figure 3 shows representative plots of the various faults alongside normal sensor readings.

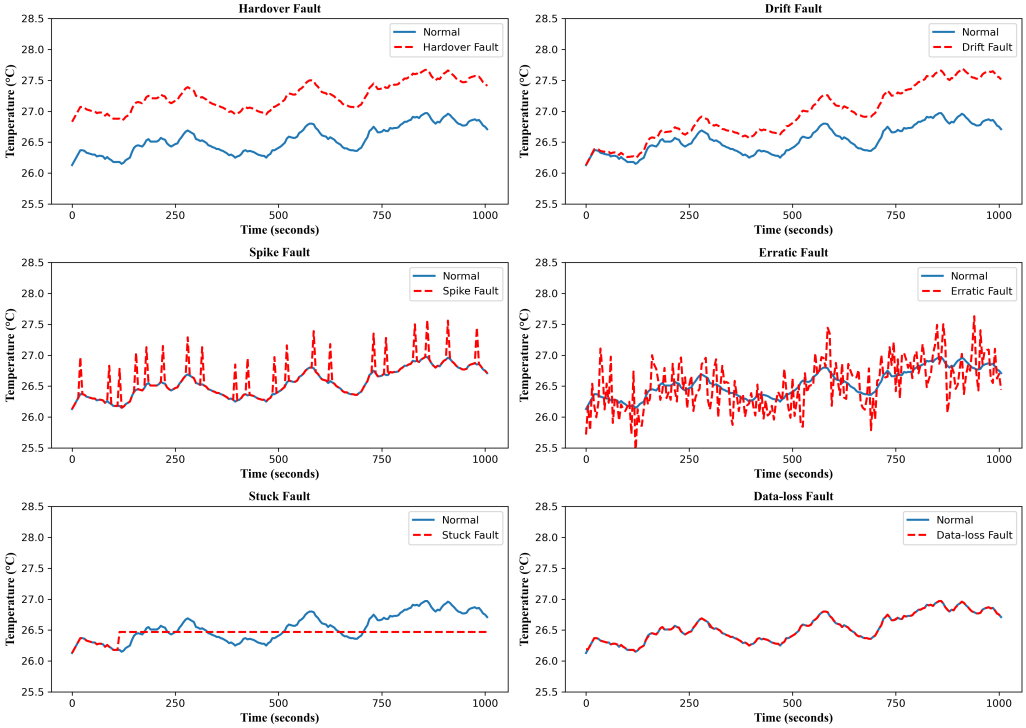


Figure 3. Representative plots of the various faults monitored by the proposed FedLSTM for distributed sensor fault detection and employed across multiple clients (sensors).

3.3. Federated Learning

FL is an innovative decentralized ML approach that enables multiple clients or servers to participate in model training while maintaining local data. Rather than transferring raw data to a central server, FL distributes a shared model to edge devices, such as smartphones, tablets, and IoT devices, enabling each client to perform training on its own dataset. Following local-data training, only the updated model parameters are transmitted to the central server, where they are aggregated to update the global model. This process is repeated iteratively as illustrated in Figure 4, and addresses concerns, like privacy, security, and data ownership, because the sensitive data remain localized at the individual clients. This is especially advantageous in applications like sensor fault detection, where raw data may contain proprietary or sensitive information that cannot be shared. FL ensures that model learning occurs in a distributed manner, mitigating the risk of data leakage while still obtaining a high-performance global model.

In sensor fault detection using FL, the central server initially sets up the global model, denoted w_0. All participating clients receive the current global model parameters w_t from the central server for the ith iteration of the FL process. The model is then trained on the local dataset of each client k, resulting in an updated set of model parameters, w_t^k. Each client sends its updated parameters to the central server after completing local training, and the FedAvg algorithm aggregates the updates as shown in Algorithm 1. This algorithm averages the parameters from all clients, forming an updated global model. The aggregated model for the ith iteration is calculated as follows:

$$\hat{w} = \frac{1}{n} \sum_{k=1}^{n} w_k \tag{7}$$

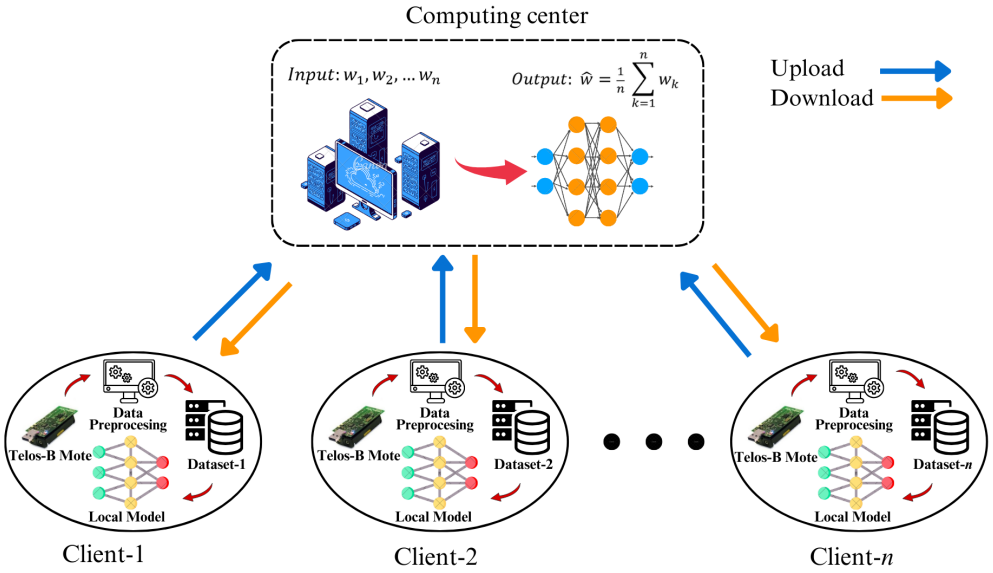


Figure 4. Framework of the proposed FedLSTM for distributed sensor fault detection. Each client trains its local model and collaborates with a central server to build the global model.

Algorithm 1 Federated learning process utilizing FedAvg technique.

Initialize: Set initial global model weights w_0 on the central server.

for each communication round $t = 1, 2, \ldots, T$ **do**
  Randomly select clients C
  for each client $k \in C$ **do**
    Send current global model weights w_{t-1} to each client in C
    Update local weights w_t^k based on received model w_{t-1}
    for each local epoch $e = 1, \ldots, E$ **do**
      for each data batch b **do**
        Update $w_t^k = w_t^k - \eta \nabla L(w_t^k; b)$
      end for
    end for
    Send updated w_t^k back to the server
  end for
  Apply FedAvg $\rightarrow$ Aggregate updates from clients:

$$\hat{w} = \frac{1}{n} \sum_{k=1}^{n} w_k$$

  Set $w_t = \hat{w}$ as the new global model weights
end for

Output: Final model weights w_T.

In order to align the model updates from multiple data sources and preserve both model integrity and relevance, consensus in FL is essential for ensuring that the global model appropriately reflects the aggregate insights obtained from all the distributed clients [49]. In this study, we define ϵ as the number of local epochs, after which clients synchronize their updates to reach consensus. Each client k sends updates to the central server after training its local model for ϵ epochs. Once updates from all clients are received, the server aggregates them and distributes the updated model to a new group of clients for additional training. Until the global model converges, this iterative process is repeated. The advantages of FL are numerous, particularly in a context like a WSNs, where data privacy, communication efficiency, and distributed data sources are critical concerns. By enabling learning across multiple devices without centralizing the data, FL not only preserves

privacy but minimizes the communication load. Additionally, it leverages distributed data sources to enhance the model's generalizations across varied environments, making it an ideal solution for sensor fault detection in federated settings.

3.4. The FedLSTM Architecture

The LSTM network is a specialized type of recurrent neural network (RNN) created especially to address the vanishing gradient problem that conventional RNNs face, making it well suited for learning long-term dependencies. LSTM significantly enhances efficiency and DL models handling sequential data and is commonly used in time-series forecasting and other sequence-based applications. In this study, we employ FedLSTM to detect sensor faults within WSNs. Our approach utilizes a federated environment where a central server updates the global model, and each client (with a distinct sensor dataset) trains the LSTM model locally.

FedLSTM reflects the federated nature of its training process. Initially, the global model is configured with random weights, which each client subsequently updates locally. To collect spatial input and extract pertinent sequential patterns, the FedLSTM architecture begins with an LSTM layer with 128 units. A dropout layer is then added to enhance generalization and reduce overfitting. A second LSTM layer, comprising 64 units, further refines temporal patterns and is accompanied by an additional dropout layer for regularization. The output from these LSTM layers passes through dense layers, with the final layer making predictions across seven distinct sensor fault classes. The layer-by-layer architecture is detailed in Table 1.

Table 1. The layered architecture of the LSTM model for sensor fault detection.

Layer (Type)	Output Shapes	Parameters	Descriptions
$\text{lstm}_1(LSTM)$	(None, 1, 128)	117,248	LSTM layer
$\text{dropout}_1(Dropout)$	(None, 1, 128)	0	Regularization layer
$\text{lstm}_2(LSTM)$	(None, 64)	49,408	LSTM layer
$\text{dropout}_2(Dropout)$	(None, 64)	0	Regularization layer
$\text{dense}_1(Dense)$	(None, 32)	2080	Fully connected layer
$\text{dense}_2(Dense)$	(None, 7)	231	Fully connected layer
Total parameters: 168,967 Trainable parameters: 168,967 Non-trainable parameters: 0			

The federated structure allows the model to learn from a range of sensor environments without requiring centralized data storage, making it highly suitable for real-world applications involving heterogeneous WSNs. This approach enhances the model's robustness, enabling it to manage diverse conditions across different WSNs configurations, thereby substantially improving fault detection accuracy across all sensors.

4. Results and Discussion

4.1. Experimental Setup

Our training processes were conducted using an NVIDIA GeForce RTX 3060 GPU for its efficiency and capability in handling complex DL tasks. This GPU, built on the advanced Ampere architecture, features 3584 CUDA cores, enabling high computational throughput. Equipped with 32 GB of GDDR6 memory, the GPU enables rapid data processing during intensive computational tasks. Our development environment was configured in Python 3.9.18, utilizing TensorFlow 2.15.0 as the primary framework.

In this study, normal temperature sensor data served as the baseline dataset. To simulate real-world scenarios, the dataset was synthetically augmented by increasing its size using a conditional GAN model and introducing faulty samples, encompassing six common sensor faults: bias, drift, spike, erratic, stuck, and data-loss. For training, synthetically generated samples were used, while the original samples were reserved exclusively for testing,

ensuring realistic evaluation. The dataset was partitioned into training (80%) and testing (20%) subsets, ensuring a balanced distribution of fault types across both sets. Training and evaluation were conducted using the proposed FedLSTM model, which employs FL to train local models across distributed clients while collaboratively constructing a global model. Figure 5 illustrates the complete methodology.

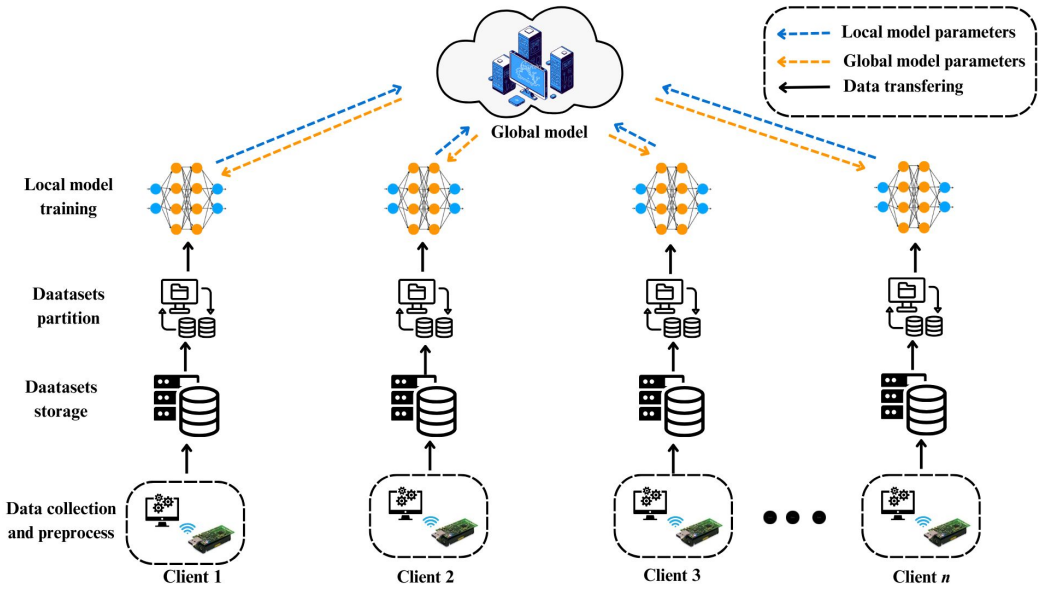


Figure 5. The proposed workflow for sensor fault diagnosis in WSNs depicts the following stages: data acquisition from multiple sensors, data preprocessing, including generating synthetic data for various common sensor faults, data partitioning for distributed storage, local model training, where clients train models locally using their respective datasets, and global model aggregation using FL for fault detection and classification.

4.2. Experimental Results

In this section, we assess the performance of FedLSTM in a range of scenarios. Leveraging the Flower FL environment, our experimental setups were rigorously designed to thoroughly analyze the behavior of the FedLSTM framework under varying conditions. Specifically, we tested it on WSN datasets containing six faulty sensor reading classes and one normal class (seven distinct classes). The experiments involved eight datasets (clients) encompassing both single-hop and multi-hop scenarios with data from indoor and outdoor sensors as listed in Table 2. The primary objective of this classification task was to accurately and efficiently identify sensor faults.

Table 2. Overview of WSNs datasets used in the study, showing single-hop and multi-hop scenarios across indoor and outdoor environments.

Dataset	Scenario	Location	Sensors	Classes	Numbers of Samples
WSNs	Single-hop	Indoor	T1, T2	7	3311, 3395
		Outdoor	T3, T4	7	3878, 3850
	Multi-hop	Indoor	T3, T4	7	3528, 3605
		Outdoor	T1, T2	7	3563, 3605

We focused on evaluating the model's performance when varying the number of clients and the local epoch configurations to understand how well it adapts to devices with diverse computational capacities. Additionally, we compared the performance of FedLSTM with two baseline neural network architectures to assess its relative efficiency.

The federated and centralized models were evaluated using accuracy, precision, sensitivity, and F1-score. The global model was created on the central server and then distributed to participating clients. Each client trained the model on its local data before sending updates to the server, which aggregated them and redistributed the updated model to a randomly selected client for further training. Data processing for the entire experiment was carried out in batch sizes of 32.

As seen in Figure 6, the centralized model slightly outperformed the federated model across all metrics. The centralized model achieved an accuracy of 96.67%, compared to the federated model's 94.45%, and reached a higher precision of 96.86%, sensitivity of 96.59%, and an F1-score of 96.63%, while the federated model recorded 94.75%, 94.62%, and 94.41%, respectively. However, while the centralized model demonstrated marginally superior performance, it also raised significant concerns including heightened data privacy concerns and increased computational resource requirements. In contrast, FL allows clients to retain their raw data locally, making it ideal for scenarios where data sensitivity or privacy concerns are paramount. Furthermore, FL significantly reduces bandwidth requirements by eliminating the need to transmit large datasets to a central server. Despite the slight performance difference, the federated approach offers a practical and scalable solution for distributed learning, particularly in real-world applications where privacy and data protection are crucial.

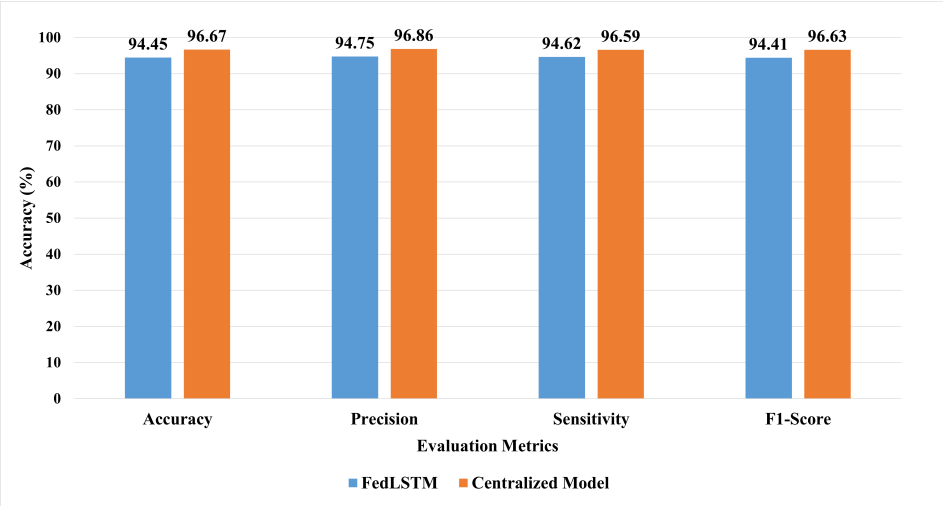
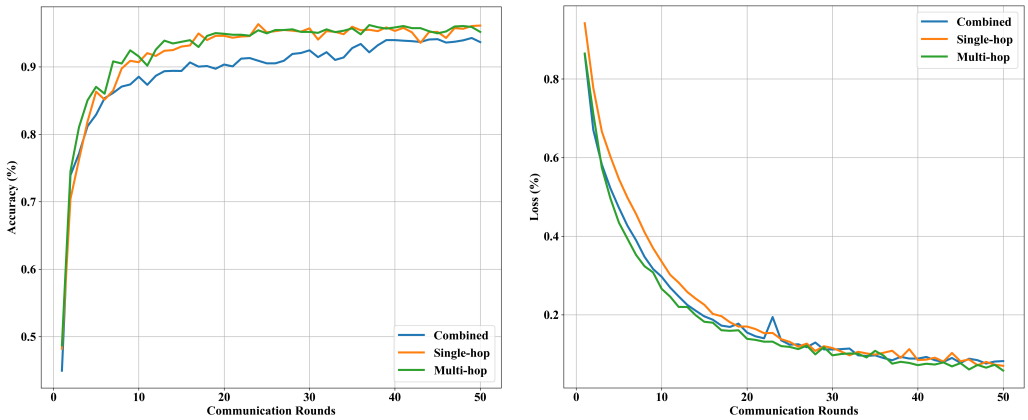


Figure 6. Comparison of FedLSTM and the centralized model for sensor fault detection in WSNs.

To further evaluate the effectiveness of the FedLSTM model in detecting sensor faults, we conducted experiments on a classification task using multiclass WSN datasets encompassing single-hop, multi-hop, and combined (single-hop and multi-hop) scenarios. Figure 7 illustrates the model's accuracy and loss over communication rounds across the three distinct scenarios. All datasets achieved high accuracy (around 90%) after approximately 15 to 20 rounds, with minor fluctuations thereafter.

The single-hop scenarios converged more quickly and sustained higher accuracy throughout communication rounds, compared to the multi-hop scenarios, which exhibited slight performance dips, likely due to the complexity of multi-hop communications affecting data consistency. The combined dataset stabilized at an accuracy level between those of the single-hop and multi-hop datasets. The loss graph in Figure 7b reveals a noticeable decline, showing a significant drop in loss during the early rounds in all three scenarios: combined, single-hop, and multi-hop. Initially, there is a steep drop in loss within the first 10 communication rounds for all datasets. The combined dataset and the multi-hop dataset show slightly faster reductions in loss compared to the single-hop dataset, which exhibits a marginally higher loss initially. However, after approximately 15 rounds, all datasets

demonstrate stable convergence, with only minor fluctuations observed towards the later rounds, ultimately achieving similarly low loss values by round 50.



(**a**) (**b**)

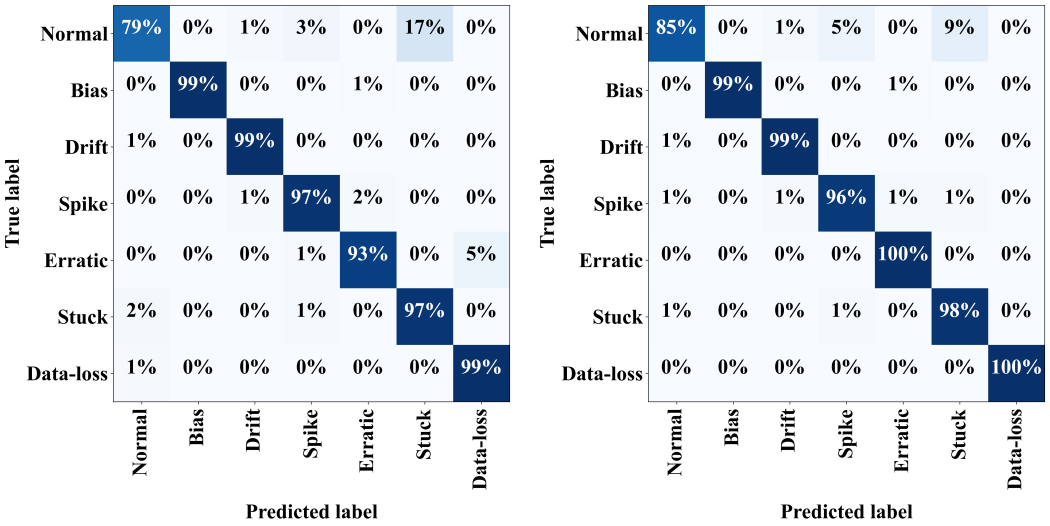
Figure 7. Performance of the FedLSTM model in terms of (**a**) accuracy and (**b**) loss over 50 communication rounds, in single-hop, multi-hop, and combined (single-hop and multi-hop) scenarios.

FedLSTM performed well across single-hop and multi-hop datasets, with the combined dataset being closer to single-hop performance but with a slightly higher loss. While expanding the number of clients in FL usually improves generalization, the distinct characteristics of single-hop and multi-hop sensors introduced minor accuracy variability due to data heterogeneity. Still, convergence was achieved across all setups. Table 3 details the classification results, showing individual client and aggregated FedLSTM and centralized model performance, while confusion matrices in Figure 8 highlight the adaptation challenges in handling diverse sensor data.

Table 3. Performance comparison for sensor fault detection by FedLSTM, a centralized model, and multiple individual clients when trained on WSNs datasets.

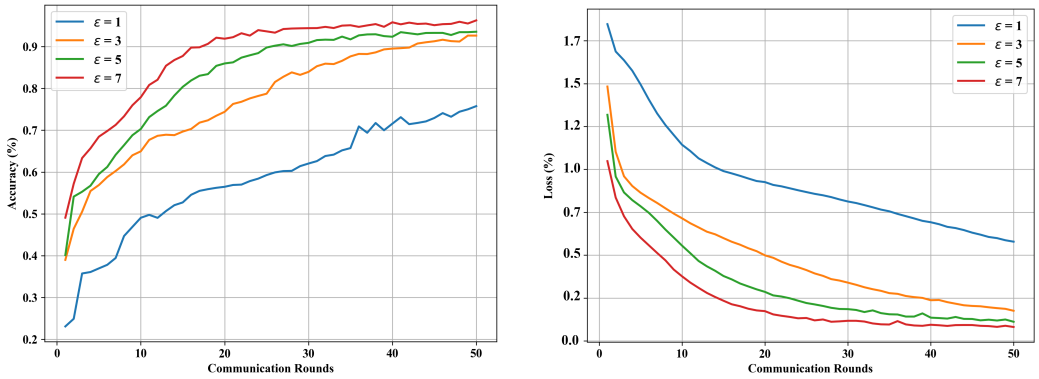
Model	Accuracy (%)	Precision (%)	Sensitivity (%)	F1-Score (%)
Client 1	96.91	97.24	96.91	96.91
Client 2	97.92	97.99	97.91	97.92
Client 3	94.61	95.35	94.61	94.61
Client 4	98.33	98.38	98.33	98.33
Client 5	97.73	98.03	97.73	97.73
Client 6	92.04	95.01	92.04	91.79
Client 7	97.68	97.77	97.68	97.67
Client 8	94.54	94.76	94.54	94.41
FedLSTM	94.45	94.75	94.60	94.41
Centralized Model	96.67	96.86	96.59	96.63

In ML, an epoch signifies a complete pass through the training dataset, during which the model conducts forward propagation (estimating output) and backward propagation (adjusting weights based on error). In FL, where data are distributed across multiple devices or nodes, the number of local epochs (indicating how many times each local dataset cycles through them) significantly impacts the quality and rate of the global model's convergence. Increasing the number of local epochs can improve model accuracy across communication rounds as shown in Figure 9a.



(**a**) (**b**)

Figure 8. Confusion matrices for (**a**) FedLSTM and (**b**) the centralized model from multiclass sensor-fault detection.



(**a**) (**b**)

Figure 9. Impact from varying the number of local epochs ϵ on (**a**) accuracy and (**b**) loss convergence of the FedLSTM model across 50 communication rounds.

When the number of local epochs was set to $\epsilon = 7$, the model initially converged rapidly, plateauing at around 95% accuracy after 40 rounds. In contrast, $\epsilon = 1$ consistently yielded lower accuracy, remaining below 80% throughout the rounds, while $\epsilon = 3$ and $\epsilon = 5$ showed intermediate performance. These trends clearly demonstrate that a higher number of epochs accelerates convergence and enhances overall accuracy, with $\epsilon = 7$ outperforming other configurations.

In Figure 9b, higher ϵ values led to a faster and more efficient reduction in loss. The model when $\epsilon = 7$ exhibited the steepest decline in loss, whereas $\epsilon = 1$ showed the slowest reduction. Both $\epsilon = 3$ and $\epsilon = 5$ consistently reduced loss, with $\epsilon = 5$ achieving the lower final loss of the two. These findings demonstrate that increasing the number of epochs improves accuracy and reduces loss while accelerating convergence.

To further validate the effectiveness of the FedLSTM model for sensor fault detection, we conducted a comparative analysis with two other neural network architectures: MLP and a 1D-CNN. The architectures and hyperparameters of the MLP and 1D-CNN models utilized for comparison are detailed in Table 4. The MLP model comprises two fully connected layers with 128 and 64 neurons, respectively, both employing the ReLU activation function. To mitigate overfitting, dropout layers with a rate of 0.2 were applied after each hidden layer. Similarly, the 1D-CNN model features a convolutional layer with 64 filters, a kernel size of 3, and ReLU activation, followed by a dropout layer and a fully connected

layer with 32 neurons. The output layer for both models consists of 7 neurons with a softmax activation function, aligning with the multiclass classification task. All models were optimized using the Adam optimizer with a learning rate of 0.002 and trained using the sparse categorical cross-entropy loss function. These configurations were carefully selected to ensure a robust and rigorous evaluation of the FedLSTM model's performance in sensor fault detection.

Table 4. The detailed configurations of architectures and hyperparameters for 1D-CNN and MLP models.

Model	Architecture	Hyperparameters
1D-CNN	Conv1D layer: 64 filters, kernel size 3 (ReLU)	Optimizer: Adam Learning rate: 0.002 Loss: Sparse categorical crossentropy Batch size: 32
	Dropout layer: 0.2	
	Dense layer: 32 neurons (ReLU)	
	Output layer: 7 neurons (softmax)	
MLP	Dense layer: 128 neurons (ReLU)	Optimizer: Adam Learning rate: 0.002 Loss: Sparse categorical crossentropy Batch size: 32
	Dropout layer: 0.2	
	Dense layer: 64 neurons (ReLU)	
	Dropout layer: 0.2	
	Output layer: 7 neurons (softmax)	

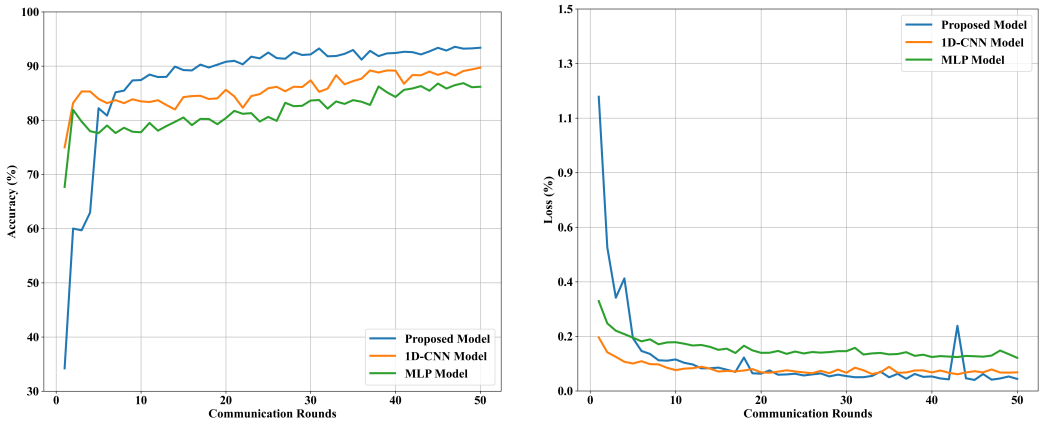
All models were evaluated across multiple FL communication rounds. The results, summarized in Table 5, highlight the superior performance of FedLSTM in terms of accuracy, precision, sensitivity, and F1-score. Specifically, FedLSTM achieved an impressive accuracy of 94.45%, outperforming both the 1D-CNN (nearly 90%) and MLP (86.02%). Additionally, FedLSTM consistently surpassed the other models with precision and sensitivity rates at 94.75% and 94.60%, respectively, and an F1-score of 94.41%. In contrast, the 1D-CNN achieved an F1-score of 89.93%, while MLP lagged behind with an F1-score of 85.84%. These findings underscore the robustness and efficiency of the proposed FedLSTM in sensor fault detection when compared to traditional neural architectures like MLP and the 1D-CNN.

Table 5. Performance of FedLSTM, the 1D-CNN, and MLP in identifying sensor faults.

Model	Accuracy (%)	Precision (%)	Sensitivity (%)	F1-Score (%)
FedLSTM	94.45	94.75	94.60	94.41
1D-CNN	90.00	91.24	90.35	89.93
MLP	86.02	88.05	86.29	85.84

Figure 10 illustrates the accuracy and loss across 50 communication rounds for each model. By the fifth round, FedLSTM had consistently outperformed both the 1D-CNN and MLP across all metrics. As shown in Figure 10a, FedLSTM exhibited a sharp increase in accuracy, surpassing 90% by the 20th round, being significantly faster than the other models. By the 40th round, MLP lagged at around 86%, 1D-CNN plateaued at nearly 90%, and FedLSTM stabilized at approximately 94.45%. These results indicate that FedLSTM not only achieved faster convergence but also sustained higher accuracy throughout the training process.

The loss graph further underscores the ability of FedLSTM to reduce loss more rapidly than the competing models. FedLSTM showed a steep decline in loss within the first 10 communication rounds, while 1D-CNN and MLP converged more gradually as illustrated in Figure 10b. By the 40th round, FedLSTM maintained the lowest loss, underscoring its superior learning efficiency in distributed environments. However, a minor spike in loss was observed around the 40th round, particularly by the combination of the single-hop and multi-hop datasets.



(**a**) (**b**)

Figure 10. (**a**) Accuracy and (**b**) loss from FedLSTM, the 1D-CNN, and MLP.

Around the 40th communication round, the observed spike in loss can be attributed to the integration of single-hop and multi-hop datasets, which are characterized by distinct data distributions. Single-hop datasets generally feature lower latency and more consistent readings, while multi-hop datasets are influenced by greater variability due to factors such as communication delays and complex network topologies. This heterogeneity likely introduced temporary inconsistencies during the global model aggregation process, resulting in the spike. Nonetheless, the loss quickly stabilized, underscoring the capacity of FedLSTM to effectively manage diverse and distributed data sources.

This spike can also be attributed to variability in the data distribution across clients. It is possible that one or more clients sent updates influenced by outlier samples or transient inconsistencies in data patterns, resulting in a slight fluctuation during model aggregation. Such occurrences are common in FL setups where non-IID (non-identically and independently distributed) data are prevalent. Notably, the model quickly stabilized post-spike, highlighting the robustness of FedLSTM in addressing such anomalies. Although 1D-CNN performed better (90%) than MLP (86%), neither model matched the robustness and rapid convergence of FedLSTM. These findings emphasize the ability of FedLSTM to recover from transient anomalies, maintaining superior learning efficiency and performance across diverse client distributions, making it a strong candidate for sensor fault detection tasks in FL setups.

4.3. Discussion

The FL approach in this work enables local model updates on individual devices without sharing the raw data, effectively addressing privacy concerns. Unlike traditional ML models, which rely on centralized data for training, FL aggregates locally trained models from multiple devices, allowing the global model to benefit from local insights without direct access to the raw data. By keeping sensitive data on client devices, our decentralized framework reduces the risk of privacy breaches and unauthorized data access while complying with legal privacy requirements. This configuration is particularly advantageous for real-world applications involving sensitive information like that collected in WSNs. Through extensive experiments using WSN datasets, the proposed approach demonstrated robust performance across varied data distribution scenarios, proving that FL maintains high accuracy while safeguarding privacy.

Despite its decentralized nature, our FedLSTM performed competitively with centralized models. By facilitating collaborative learning across multiple client devices, the global model effectively generalizes across diverse data sources without requiring data sharing, thereby preserving both privacy and high accuracy. Our results showed that FedLSTM achieved performance closely aligning with centralized models that leverage all available data. These findings underscore the effectiveness of FL in distributed environments, particularly where data privacy is paramount. Although centralized models slightly out-

performed in accuracy, precision, sensitivity, and F1-score, achieving 96.67% accuracy compared to that of FedLSTM at 94.45%, FedLSTM offers significant advantages in terms of privacy, scalability, and reduced computational and bandwidth requirements, all critical for real-world distributed systems like WSNs. While centralized models might yield slight improvements in accuracy, these improvements often depend on direct access to raw data, a limitation that is impractical in privacy-sensitive applications.

Our experiments with single-hop and multi-hop sensor data provided additional insights. Single-hop datasets generally achieved faster convergence and higher accuracy, likely due to their lower complexity compared to multi-hop communications. The added complexity in multi-hop datasets introduces variability, leading to slight reductions in accuracy and consistency. Nevertheless, overall performance across both dataset types showed steady convergence, with accuracy stabilizing after approximately 20 communication rounds. These results were further validated by the observed loss. Notably, with the single-hop dataset, FedLSTM demonstrated a rapid initial reduction in loss, reaching optimal levels within about 10 rounds. Although multi-hop settings displayed more variability, they ultimately converged to low loss values, highlighting the resilience of FedLSTM in both contexts despite the greater variability in multi-hop settings.

The influence of local epochs on model accuracy and convergence speed emerged as another significant finding. Increasing the number of epochs accelerated convergence, improving accuracy across communication rounds. However, beyond a certain threshold, the benefits diminished as performance gains plateaued, even with additional epochs. This suggests that fine-tuning the number of local epochs is essential to balancing convergence speed with performance. When comparing FedLSTM to other neural network models (the 1D-CNN and MLP), FedLSTM consistently demonstrated superior efficiency and robustness in FL settings. Across the key metrics of accuracy, precision, sensitivity, and F1-score, FedLSTM outperformed the baseline models, exhibiting faster convergence and competitive performance relative to centralized models, thereby validating its effectiveness for distributed sensor fault detection.

5. Conclusions

In this study, we introduced FedLSTM, a federated learning sensor fault detection model designed for WSNs. In IoT environments, transferring data to a central base station for analysis raises significant concerns, as data privacy is becoming increasingly critical for organizations. Our FedLSTM framework addresses these concerns by enabling decentralized fault detection, allowing clients to contribute to the training process without sharing data with a central server. Through a series of experiments, we evaluated FL as a viable alternative to centralized models for detecting sensor faults in WSNs while maintaining data privacy. The results show that a collaborative federated system employing horizontal data partitioning achieved performance levels comparable to those of a centralized model. To support this approach, we used WSNs datasets and included synthetic faulty sensor readings by simulating common faults. These synthetic samples effectively replicated real-world bias, drift, spike, erratic, stuck, and data-loss faults.

The FL architecture comprised eight clients and a central server, with the server managing the overall model update process using the FedAvg algorithm for parameter aggregation. Our results show that the FL system is an effective alternative to centralized models, achieving comparable accuracy, precision, sensitivity, and F1-scores in multiclass fault classification. Although the centralized model showed slightly higher performance, its use raises concerns regarding data privacy and the greater computational resources required. Modifying the number of local training epochs demonstrated that even devices with limited computational power effectively contribute to training the federated model because fewer epochs still yielded competitive results. Moreover, FedLSTM maintained client privacy, minimized communications overhead, and reduced the load on the central server. In comparison with well-known neural networks in FL settings, such as the 1D-CNN and MLP, the proposed FedLSTM maintained high performance while providing

the dual advantages of security and communication efficiency. This study highlights the potential of FL to protect data privacy while ensuring robust fault detection capabilities in IoT and WSN environments.

Future work will focus on addressing data splits and client heterogeneity by employing advanced aggregation techniques, such as adaptive federated averaging, to enhance model performance and convergence. Expanding fault simulations to encompass more diverse and complex scenarios, such as intermittent or cascading faults, could further improve the robustness of the framework. Furthermore, extending the application of FedLSTM to additional sensor types, including vibration and acoustic sensors, would demonstrate its versatility across a broader range of IoT contexts. Hybrid approaches that integrate FL with transfer learning could effectively handle imbalanced datasets and accelerate training processes. Moreover, enhancing communication efficiency and implementing robust security mechanisms to mitigate adversarial threats would bolster system reliability. Lastly, real-world deployments in large-scale, dynamic IoT environments would provide valuable insights into the scalability and practicality of the proposed framework.

Author Contributions: Conceptualization: R.K. and U.S.; methodology: R.K.; software: R.K.; validation: R.K., U.S. and I.K.; formal analysis: R.K. and U.S.; investigation: R.K.; resources: R.K.; data curation: R.K.; writing—original draft preparation: R.K.; writing—review and editing: U.S. and I.K.; visualization: R.K.; supervision: U.S. and I.K.; project administration: I.K.; funding acquisition: I.K. All authors have read and approved the published version of this manuscript.

Funding: This research was supported by the Regional Innovation Strategy (RIS) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2021RIS-003).

Data Availability Statement: The datasets supporting the findings of this study are publicly accessible from Suthaharan, S.; Alzahrani, M.; Rajasegarar, S.; Leckie, C.; Palaniswami, M. "Labelled Data Collection for Anomaly Detection in Wireless Sensor Networks". [Online]. Available: https://home.uncg.edu/cmp/downloads/lwsndr.html, (accessed on 15 July 2024).

Conflicts of Interest: The authors declare they have no conflicts of interest related to the publication of this paper.

References

1. Hussain, F.; Hussain, F. Internet of everything. In *Internet of Things: Building Blocks and Business Models*; Springer: Cham, Switzerland, 2017; pp. 1–11.
2. Ratti, C.; Claudel, M. *The City of Tomorrow: Sensors, Networks, Hackers and the Future of Urban Life*; Yale University Press: New Haven, CT, USA, 2016.
3. Taye, M.M. Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. *Computers* **2023**, *12*, 91. [CrossRef]
4. Rivas, A.E.L.; Abrao, T. Faults in smart grid systems: Monitoring, detection and classification. *Electr. Power Syst. Res.* **2020**, *189*, 106602. [CrossRef]
5. Asharf, J.; Moustafa, N.; Khurshid, H.; Debie, E.; Haider, W.; Wahab, A. A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. *Electronics* **2020**, *9*, 1177. [CrossRef]
6. Khalil, R.A.; Saeed, N.; Masood, M.; Fard, Y.M.; Alouini, M.S.; Al-Naffouri, T.Y. Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications. *IEEE Internet Things J.* **2021**, *8*, 11016–11040. [CrossRef]
7. Shrestha, A.; Mahmood, A. Review of deep learning algorithms and architectures. *IEEE Access* **2019**, *7*, 53040–53065. [CrossRef]
8. AbdulRahman, S.; Tout, H.; Ould-Slimane, H.; Mourad, A.; Talhi, C.; Guizani, M. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet Things J.* **2020**, *8*, 5476–5497. [CrossRef]
9. Andersson, K.P.; Ostrom, E. Analyzing decentralized resource regimes from a polycentric perspective. *Policy Sci.* **2008**, *41*, 71–93. [CrossRef]
10. Noel, A.B.; Abdaoui, A.; Elfouly, T.; Ahmed, M.H.; Badawy, A.; Shehata, M.S. Structural health monitoring using wireless sensor networks: A comprehensive survey. *IEEE Commun. Surv. Tutorials* **2017**, *19*, 1403–1423. [CrossRef]
11. Mahmud, R.; Ramamohanarao, K.; Buyya, R. Latency-aware application module management for fog computing environments. *ACM Trans. Internet Technol. (TOIT)* **2018**, *19*, 1–21. [CrossRef]
12. Miao, X.; Liu, Y.; Zhao, H.; Li, C. Distributed online one-class support vector machine for anomaly detection over networks. *IEEE Trans. Cybern.* **2018**, *49*, 1475–1488. [CrossRef] [PubMed]
13. Sun, T.; Li, D.; Wang, B. Decentralized federated averaging. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 4289–4301. [CrossRef] [PubMed]

14. Abdelmoula, I.A.; Oufettoul, H.; Lamrini, N.; Motahhir, S.; Mehdary, A.; El Aroussi, M. Federated learning for solar energy applications: A case study on real-time fault detection. *Sol. Energy* **2024**, *282*, 112942. [CrossRef]
15. Zikria, Y.B.; Ali, R.; Afzal, M.K.; Kim, S.W. Next-generation internet of things (iot): Opportunities, challenges, and solutions. *Sensors* **2021**, *21*, 1174. [CrossRef]
16. Ali, S.; Qaisar, S.B.; Saeed, H.; Farhan Khan, M.; Naeem, M.; Anpalagan, A. Network challenges for cyber physical systems with tiny wireless devices: A case study on reliable pipeline condition monitoring. *Sensors* **2015**, *15*, 7172–7205. [CrossRef]
17. Taheri, S. A review on five key sensors for monitoring of concrete structures. *Constr. Build. Mater.* **2019**, *204*, 492–509. [CrossRef]
18. Li, D.; Wang, Y.; Wang, J.; Wang, C.; Duan, Y. Recent advances in sensor fault diagnosis: A review. *Sens. Actuators A Phys.* **2020**, *309*, 111990. [CrossRef]
19. Jieyang, P.; Kimmig, A.; Dongkun, W.; Niu, Z.; Zhi, F.; Jiahai, W.; Liu, X.; Ovtcharova, J. A systematic review of data-driven approaches to fault diagnosis and early warning. *J. Intell. Manuf.* **2023**, *34*, 3277–3304. [CrossRef]
20. Khan, R.; Khan, S.U.; Saeed, U.; Koo, I.S. Auscultation-Based Pulmonary Disease Detection through Parallel Transformation and Deep Learning. *Bioengineering* **2024**, *11*, 586. [CrossRef]
21. Saeed, U.; Shah, S.Y.; Shah, S.A.; Ahmad, J.; Alotaibi, A.A.; Althobaiti, T.; Ramzan, N.; Alomainy, A.; Abbasi, Q.H. Discrete human activity recognition and fall detection by combining FMCW RADAR data of heterogeneous environments for independent assistive living. *Electronics* **2021**, *10*, 2237. [CrossRef]
22. Jiang, G.; He, H.; Xie, P.; Tang, Y. Stacked multilevel-denoising autoencoders: A new representation learning approach for wind turbine gearbox fault diagnosis. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 2391–2402. [CrossRef]
23. Zhang, Z.; Mehmood, A.; Shu, L.; Huo, Z.; Zhang, Y.; Mukherjee, M. A survey on fault diagnosis in wireless sensor networks. *IEEE Access* **2018**, *6*, 11349–11364. [CrossRef]
24. Saeed, U.; Jan, S.U.; Lee, Y.D.; Koo, I. Fault diagnosis based on extremely randomized trees in wireless sensor networks. *Reliab. Eng. Syst. Saf.* **2021**, *205*, 107284. [CrossRef]
25. Huang, J.; Li, M.; Zhang, Y.; Mu, L.; Ao, Z.; Gong, H. Fault detection and classification for sensor faults of UAV by deep learning and time-frequency analysis. In Proceedings of the 2021 40th Chinese Control Conference (CCC), Shanghai, China, 26–28 July 2021; pp. 4420–4424.
26. Gou, L.; Li, H.; Zheng, H.; Li, H.; Pei, X. Aeroengine control system sensor fault diagnosis based on CWT and CNN. *Math. Probl. Eng.* **2020**, *2020*, 5357146. [CrossRef]
27. Hasan, M.N.; Jan, S.U.; Koo, I. Wasserstein GAN-based digital twin-inspired model for early drift fault detection in wireless sensor networks. *IEEE Sens. J.* **2023**, *23*, 13327–13339. [CrossRef]
28. Hu, S.; Chen, X.; Ni, W.; Hossain, E.; Wang, X. Distributed machine learning for wireless communication networks: Techniques, architectures, and applications. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1458–1493. [CrossRef]
29. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistic, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
30. Cao, K.; Hu, S.; Shi, Y.; Colombo, A.W.; Karnouskos, S.; Li, X. A survey on edge and edge-cloud computing assisted cyber-physical systems. *IEEE Trans. Ind. Informatics* **2021**, *17*, 7806–7819. [CrossRef]
31. Moshawrab, M.; Adda, M.; Bouzouane, A.; Ibrahim, H.; Raad, A. Reviewing federated learning aggregation algorithms; strategies, contributions, limitations and future perspectives. *Electronics* **2023**, *12*, 2287. [CrossRef]
32. Hard, A.; Rao, K.; Mathews, R.; Ramaswamy, S.; Beaufays, F.; Augenstein, S.; Eichner, H.; Kiddon, C.; Ramage, D. Federated learning for mobile keyboard prediction. *arXiv* **2018**, arXiv:1811.03604.
33. Brisimi, T.S.; Chen, R.; Mela, T.; Olshevsky, A.; Paschalidis, I.C.; Shi, W. Federated learning of predictive models from federated electronic health records. *Int. J. Med Inform.* **2018**, *112*, 59–67. [CrossRef]
34. Liu, Y.; James, J.; Kang, J.; Niyato, D.; Zhang, S. Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet Things J.* **2020**, *7*, 7751–7763. [CrossRef]
35. Li, J.; Wang, Y.; Zi, Y.; Zhang, Z. Whitening-Net: A generalized network to diagnose the faults among different machines and conditions. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 5845–5858. [CrossRef] [PubMed]
36. Zhang, W.; Li, X.; Ma, H.; Luo, Z.; Li, X. Federated learning for machinery fault diagnosis with dynamic validation and self-supervision. *Knowl.-Based Syst.* **2021**, *213*, 106679. [CrossRef]
37. Liu, L.; Yan, Z.; Zhang, T.; Gao, Z.; Cai, H.; Wang, J. Data privacy protection: A novel federated transfer learning scheme for bearing fault diagnosis. *Knowl.-Based Syst.* **2024**, *291*, 111587. [CrossRef]
38. Yang, W.; Chen, J.; Chen, Z.; Liao, Y.; Li, W. Federated transfer learning for bearing fault diagnosis based on averaging shared layers. In Proceedings of the 2021 Global Reliability and Prognostics and Health Management (PHM-Nanjing), Nanjing, China, 15–17 October 2021; pp. 1–7.
39. Ma, X.; Wen, C.; Wen, T. An asynchronous and real-time update paradigm of federated learning for fault diagnosis. *IEEE Trans. Ind. Inform.* **2021**, *17*, 8531–8540. [CrossRef]
40. Zhou, R.; Li, Y.; Lin, X. A clustered federated learning framework for collaborative fault diagnosis of wind turbines. *Appl. Energy* **2025**, *377*, 124532. [CrossRef]
41. Wang, X.; Yan, K. Fault detection and diagnosis of HVAC system based on federated learning. In Proceedings of the 2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on

Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Calabria, Italy, 12–15 September 2022; pp. 1–8.
42. Agrawal, S.; Sarkar, S.; Aouedi, O.; Yenduri, G.; Piamrat, K.; Alazab, M.; Bhattacharya, S.; Maddikunta, P.K.R.; Gadekallu, T.R. Federated learning for intrusion detection system: Concepts, challenges and future directions. *Comput. Commun.* **2022**, *195*, 346–361. [CrossRef]
43. Lazzarini, R.; Tianfield, H.; Charissis, V. Federated learning for IoT intrusion detection. *AI* **2023**, *4*, 509–530. [CrossRef]
44. Khan, S.U.; García, C.E.; Hwang, T.; Koo, I. Radio environment map construction based on privacy-centric federated learning. *IEEE Access* **2024**, *12*, 28109–28121. [CrossRef]
45. Mertens, J.S.; Galluccio, L.; Morabito, G. Federated learning through model gossiping in wireless sensor networks. In Proceedings of the 2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Bucharest, Romania, 24–28 May 2021; pp. 1–6.
46. Suthaharan, S.; Alzahrani, M.; Rajasegarar, S.; Leckie, C.; Palaniswami, M. Labelled data collection for anomaly detection in wireless sensor networks. In Proceedings of the 2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Brisbane, Australia, 7–10 December 2010; pp. 269–274.
47. Jan, S.U.; Lee, Y.D.; Koo, I.S. A distributed sensor-fault detection and diagnosis framework using machine learning. *Inf. Sci.* **2021**, *547*, 777–796. [CrossRef]
48. Saeed, U.; Lee, Y.D.; Jan, S.U.; Koo, I. CAFD: Context-aware fault diagnostic scheme towards sensor faults utilizing machine learning. *Sensors* **2021**, *21*, 617. [CrossRef]
49. Xu, W.; An, J.; Li, H.; Gan, L.; Yuen, C. Algorithm-unrolling-based distributed optimization for RIS-assisted cell-free networks. *IEEE Internet Things J.* **2023**, *11*, 944–957. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.