IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 March 2024, accepted 28 April 2024, date of publication 6 May 2024, date of current version 20 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3397000

RESEARCH ARTICLE

A Federated Learning Approach for Efficient Anomaly Detection in Electric Power Steering Systems

KIMLEANG KEA[1], (Graduate Student Member, IEEE), YOUNGSUN HAN[1], (Member, IEEE), AND YOUNG-JAE MIN[2], (Member, IEEE)

[1]Department of AI Convergence, Pukyong National University, Busan 48513, South Korea
[2]Department of Electric and Electronic Engineering, Halla University, Wonju 26404, South Korea

Corresponding authors: Youngsun Han (youngsun@pknu.ac.kr) and Young-Jae Min (youngjae.min@halla.ac.kr)

This work was supported by the Semiconductor Research and Development Support Project through Gangwon Technopark (GWTP) funded by Gangwon Province under Grant GWTP 2024-029.

ABSTRACT Electric power steering (EPS) has emerged as a valuable driver-assistance system. In an EPS system, an extensive amount of data collected from various sensors is analyzed to enhance the driving experience. Anomaly detection techniques have shown potential in ensuring the integrity of data patterns and detecting abnormalities to prevent adverse driving incidents, thus improving vehicle safety. However, traditional centralized anomaly detection methods require the collection of data from all EPS sensors, resulting in high communication network overhead. Herein, we propose an approach for anomaly detection using EPS data within a federated learning (FL) framework. Our approach exploits deep learning (DL) on time series data to achieve highly effective anomaly detection. This synergy of FL and DL enables all sensors in an EPS system to collaboratively train the anomaly detection model simultaneously. Our results demonstrate the feasibility of combining FL with DL anomaly detection in EPS systems, thus overcoming the limitations of the traditional centralized approach. The combined FL and DL model performs remarkably well, achieving an F1-Score of 99.01%, outperforming the 98.31%∼98.89% achieved by the centralized approach. It also exhibits high generalizability by incorporating insights from all sensors to comprehensively understand diverse driving scenarios. This results in a significant reduction in error rates, ranging from 20% to 33.25%, compared to centralized methods. Additionally, the proposed model exhibits significant advantages over traditional methods, including reduced training time and communication overhead, while maintaining comparable anomaly detection accuracy performance.

INDEX TERMS Anomaly detection, deep learning, electric power steering (EPS), federated learning.

I. INTRODUCTION

Electric power steering (EPS) is a critical system in vehicles, ensuring smoothness, efficient handling, superior agility, stability, low-temperature resistance, and low power consumption compared to traditional hydraulic steering systems. It also exhibits remarkable endurance and has been gradually replacing conventional steering systems [1]. This is due to the superiority of the EPS system over conventional steering systems across multiple key dimensions, such as safety, cost-effectiveness, energy efficiency, environmental sustainability, and ease of assembly [2]. As the usage of EPS continues to grow, the need to advance related technologies becomes increasingly crucial to ensure the safety and effectiveness of driving assistance. The EPS system is composed of several key sensors, including the steering wheel, intermediate shaft, motor, torque sensors, and reduction gear. These sensors work together to enhance the driver's experience and safety on the road. However, these sensors may exhibit errors and malfunctions. Therefore, identifying the potential errors

The associate editor coordinating the review of this manuscript and approving it for publication was Mehedi Masud.

and malfunctions in EPS sensors can significantly enhance driving safety and reliability. Even minor failures in these sensors could result in severe consequences for drivers. Possible failures in an EPS system are characterized as component failures (e.g., actuator, sensors, ECU, etc.) and incipient failures, such as variation of motor parameters, brush arcing and commutator/brush friction, overload or overheating of the stator coil, damaged or broken bearings resulting in increased friction, and worn steering gear and reduction mechanism [3]. To effectively address failures within the EPS system, a robust understanding is essential. Thus, the current paper primarily targets sensor failures.

The anomaly detection technique is a powerful technique for effectively addressing potential sensor issues, making it a valuable tool for enhancing the dependability and overall performance of the EPS system. An anomaly arises when a particular EPS sensor data point is excessively high or low compared to the rest. Therefore, through the periodic capture and analysis of EPS sensor data, early detection of anomalies takes precedence, especially in preventing incorrect steering due to sensor malfunctions. There are three main categories of anomalies related to sensor data, i.e., point anomalies, contextual anomalies, and collective anomalies [4]. Deep learning (DL) stands out as the preferred choice for such scenarios. Leveraging DL techniques, anomaly detection excels in recognizing deviations and irregular patterns within EPS sensor data, effectively triggering the presence of abnormal behavior that results in system malfunctions [5]. This conventional DL typically requires the accumulation of substantial volumes of EPS sensor data for training the anomaly detection technique effectively [6]. However, collecting extensive EPS sensor data can be challenging due to high communication overheads, which hinders the development of the DL model. Additionally, these methods may not be suitable for cases involving EPS vehicles with confidential data due to privacy concerns.

Federated learning (FL) has recently emerged as a distributed machine learning (ML) technique, offering solutions to the challenges associated with conventional DL methods. FL allows multiple parties to train the DL model collaboratively while maintaining their data locally, thereby ensuring that data privacy is preserved [7]. This stands as a key advantage of FL, as it upholds data privacy while simultaneously reducing network communication overhead [8]. In the FL framework, it consists of a central server and multiple clients. These clients work together to train the DL model using their respective individual datasets, eliminating the need to transmit their data to the central server. For example, in an EPS system, each vehicle possesses its unique local dataset and a DL model for training. After each training round, only the DL model parameters are transmitted to the server for aggregation to update the global DL model, which is then redistributed back to the vehicle clients.

While FL offers improved communication overhead and privacy advantages compared to conventional DL methods, it presents numerous challenges when applied to real-world scenarios, setting it apart from conventional approaches [9]. These challenges encompass various factors, including the persisting communication overhead involved in transmitting model parameters between the server and clients, the computational power and energy demands placed on clients, and the inherent heterogeneity associated with a vast number of local clients participating in the training process.

In this paper, we present an approach that utilizes both FL and unsupervised anomaly detection (USAD) DL models to identify anomalies within EPS system data effectively. This study mainly focuses on point anomalies, which occur when a singular data point of sensors is significantly different from the rest of the dataset in terms of its characteristics. Within this approach, DL models are employed to detect anomalies, while FL is utilized to mitigate communication overhead, ensure privacy preservation, and enhance model generalization. The main contributions of our study are as follows:

- We achieved remarkable accuracy in detecting and predicting anomalies within EPS time series data using a state-of-the-art USAD model.
- We utilized the benefits of FL alongside a USAD model to efficiently create a collaborative and distributed approach for detecting anomalies in EPS time series data. This conjunctive method notably improves the USAD model's ability to generalize for anomaly detection without requiring data exchange between servers and clients, thereby strengthening data privacy.
- We evaluated the effectiveness of our integrated approaches in contrast to centralized learning using EPS time series data, showcasing that our framework accurately detects anomalies with minimal communication overhead.

The remainder of this paper is organized as follows: In Section II, we provide the related works conducted on conventional DL- and FL-based anomaly detection. In Section III, we present the methodology, including the DL model architecture and FL training process. In Section IV, we detail the experimental setup, including the dataset, FL simulation setup, and evaluation metrics for comparisons. In Section V, comprehensively present our findings and conduct a thorough comparison with outcomes derived from a centralized learning approach. Finally, in Section VI, we present the conclusions drawn from this work.

II. RELATED WORKS

In this section, we review the relevant DL- and FL-based studies for anomaly detection in sensor data.

A. DL-BASED ANOMALY DETECTION

The DL-based technique is extensively utilized for anomaly detection within sensor data, particularly in cases where labeled anomalies are not present, thereby rendering manual detection challenging. As a result, the utilization of DL

models for EPS applications remains notably limited. However, numerous studies have explored this field. In [10], Alabe et al. introduced a DL model for anomaly detection in EPS sensor data. Their approach involves a two-stage process with an autoencoder (AE) and long short-term memory (LSTM). The results demonstrate that the proposed model excels in anomaly detection with an accuracy of 0.99. However, its implementation entails the collection of data from diverse EPS sensors for training a DL model. This process is time-consuming, demanding high bandwidth over the network, and may potentially raise privacy concerns. Kim and Jung [11] demonstrated anomaly estimation in a rack-type EPS system for a motor using a DL observer. They conducted a comparative analysis, contrasting the estimation performance of the DL observer with model-based approaches. As a result, DL observers showed 84% to 95% of estimation accuracy between estimates and actual anomalies. Ji and Lee [12] presented an anomaly detection approach utilizing a one-class Support Vector Machine (SVM) to validate control functions and streamline the analysis of test data from a hybrid electric vehicle (HEV). However, the results fail to highlight which signals cause anomalies for a comprehensive understanding. In [13], Kavousi-Fard et al. proposed a DL-based approach for cyber attack detection in vehicles, which involves classifying message frames transferred between the electric control unit (ECU) and other hardware in the vehicle. This is important for controlling automated vehicles. Similarly, while this method effectively detects issues, it also raises privacy concerns, as data is still transmitted between the ECUs, potentially compromising user privacy.

B. FL-BASED ANOMALY DETECTION

Here, we present related works that utilized FL with DL, including, but not limited to, EPS sensors. Elbir et al. [14] investigated the usage of FL over centralized learning in vehicular network applications to develop intelligent transportation systems. They also provide a comprehensive analysis of the feasibility of FL for ML-based vehicular applications. Zhang et al. [15] introduced end-to-end FL with DL for autonomous driving vehicles to predict the wheel steering angle. Their findings underscore the enhancement of FL to the quality of local DL models. However, anomaly detection was not addressed in this research. Sater and Hamza [16] developed a federated stacked LSTM for anomaly detection in sensors' time series data in smart buildings. They applied a federated learning approach, using multi-task learning to solve multiple tasks simultaneously. Their experiments on three benchmark datasets demonstrated the federated LSTM effectiveness, outperforming a centralized model. Jithish et al. [17] proposed an FL-based anomaly detection system for smart grids. Their method trains ML models locally in smart meters, ensuring user privacy by avoiding central data sharing. They use a global model downloaded to smart meters for on-device training. Results show FL models match centralized ML performance in anomaly detection while preserving user privacy. In [18], Nguyen et al. introduced a peer-to-peer deep FL approach that trains deep architectures in a fully decentralized manner, eliminating the need for central training. They demonstrated how deep FL can enhance model stability, guarantee convergence, and effectively address issues arising from imbalanced data distribution during training with FL methods. Nonetheless, this study is a general exploration aimed at understanding the effectiveness of FL. Therefore, to the best of our knowledge, our paper is the first to explore anomaly detection in EPS sensors within the automotive vehicle field using FL with the DL model.

III. FEDERATED LEARNING-BASED METHODOLOGY

In this section, we outline the state-of-the-art AE neural network, present the problem statement, introduce an AE-enhanced model designed for detecting anomalies in EPS sensor data, and elaborate on the implementation of the FL architecture adjusted for anomaly detection.

A. AUTOENCODER NETWORK

AE is an unsupervised learning technique that consists of an encoder E and a decoder D. The encoder maps input X to a set of latent spaces Z. In contrast, the decoder maps the latent space Z back into the input data as a reconstruction, as shown in Figure 1. The deviation between the original input X and reconstruction is referred to as the reconstruction error, which is defined as follows:

$$L_{AE} = \|X - AE(X)\|_2 \tag{1}$$

where

$$AE(X) = D(Z) \quad Z = E(X) \tag{2}$$

and $\|\cdot\|_2$ denotes L2-norm.

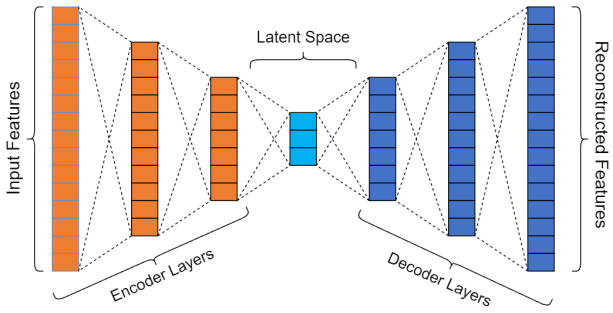


FIGURE 1. Overall architecture of the AE network.

B. PROBLEM STATEMENT

EPS sensors generate data in a time series format at precise intervals, be it in milliseconds, seconds, or minutes. This characteristic directs our attention towards anomaly detection within time series data [19]. It is divided into two primary categories: univariate, which involves a single feature, and multivariate, including multiple features. In the EPS system,

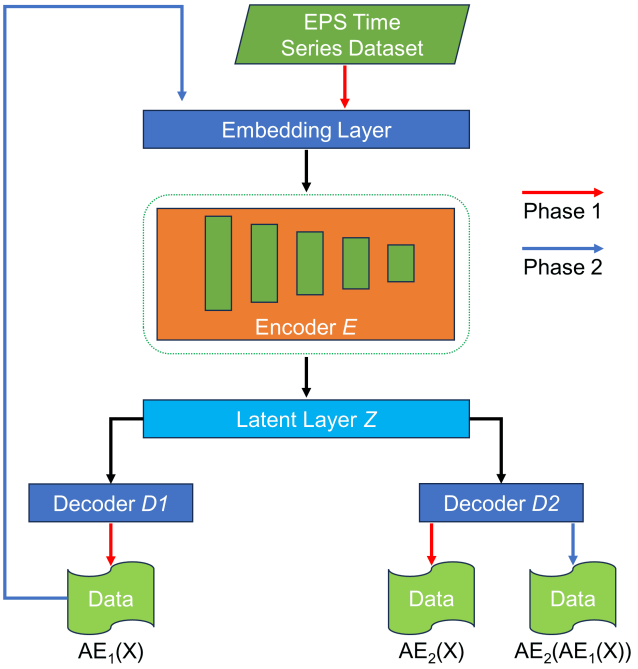


FIGURE 2. Unsupervised anomaly detection architecture illustrating the two-phase training. The red arrows represent the first stage of the reconstruction training phase, and the blue arrows represent the second stage of the adversarial training phase. EPS denotes electric power steering, and AE is a state-of-the-art autoencoder.

numerous sensors are utilized, leading us to focus on the analysis of multivariate time series data. A multivariate time series is a sequence of data points continuously collected at equal-space timestamps defined as follows:

$$X = [X_1, X_2, \ldots, X_n], \tag{3}$$

where n is the number of data points. The observation vector of the i^{th} metric with dimensionality d can be defined as follows:

$$X^i = [x_1^i, x_2^i, \ldots, x_d^i]. \tag{4}$$

Anomaly detection utilizes multivariate time series data to determine whether an observation $X_t = [x_d^1, x_d^2, \ldots, x_d^i]$ at time t is normal or abnormal. In an EPS system with FL, we assume N vehicles, where each vehicle has its own dataset X_k^i, which is kept locally. Data X_k^i of the k^{th} vehicle are not shared with the server. The conventional centralized approach collects and has access to all local training data $X = \cup_{k=1}^{N} X_k^i$ obtained from all vehicles. In contrast, the FL system only collects and aggregates updated local models obtained from vehicles to generate and update a global model.

C. DEEP LEARNING FOR ANOMALY DETECTION

In this work, we employ the state-of-the-art AE neural network to implement unsupervised anomaly detection (USAD) using EPS time series data [20]. This novel approach employs adversarially trained AEs for anomaly detection. Its unique architecture enables AEs to learn in an unsupervised manner without the need for labeled data. By combining adversarially trained AEs, the network can effectively detect anomalies while ensuring fast training. The USAD method expands upon the AE architecture through a two-phase adversarial training framework. The AE operates by processing normal data using an encoder and decoder structure, effectively reconstructing the normal data; however, it has difficulties when dealing with abnormal data [21]. The method employs three essential components: an encoder network E and two decoder networks D_1 and D_2. The three components are interconnected with two AEs (AE_1 and AE_2), which share the same encoder network, as shown in Figure 2.

The network is trained in two distinct phases. In the first phase, the two AEs are trained to acquire the ability to accurately reconstruct normal data. In the second phase, the two AEs engage in an adversarial training process, where AE_1 tries to deceive AE_2 by generating reconstruction data similar to real data; AE_2 tries to differentiate the data directly obtained from the normal dataset from the reconstruction data produced by AE_1. The first training phase is defined as follows:

$$AE_1(X) = D_1(E(X)), \quad AE_2(X) = D_2(E(X)), \tag{5}$$

where X is the EPS time series data compressed by encoder E to the latent layer $Z = E(X)$, and reconstructed by each decoder D_1 and D_2.

In the second phase, AE_2 needs to distinguish between real input data and encoded data generated by AE_1. AE_1 is trained to deceive AE_2 by compressing its output with E into Z, then reconstructing it again using $AE_2(AE_1(X))$. AE_1 aims to minimize the difference between real input data X and AE_2 output data, whereas AE_2 aims to maximize this difference in an adversarial training configuration. Furthermore, AE_1 trains based on its success in deceiving AE_2, while AE_2 tries to differentiate between the output reconstructed by AE_1 and the real input data. The second training phase is defined as follows:

$$\min_{AE_1} \max_{AE_2} \|X - AE_2(AE_1(X))\|_2, \tag{6}$$

that produces the following loss functions:

$$\begin{aligned} L_{AE_1} &= +\|X - AE_2(AE_1(X))\|_2, \\ L_{AE_2} &= -\|X - AE_2(AE_1(X))\|_2, \end{aligned} \tag{7}$$

where L is the loss function and $\| \ \|_2$ denotes the L2-norm.

Thus, we employ the USAD technique for analyzing EPS sensor data. This method enables the system to learn and effectively amplify the reconstruction error associated with anomalies present in the EPS sensor data. Notably, it accomplishes the anomaly detection task rapidly while ensuring increased stability in the process.

D. FEDERATED LEARNING (FL)

In this section, we will introduce the FL approach, which is utilized to enhance the USAD model for detecting abnormalities in EPS sensor data collaboratively. In the FL

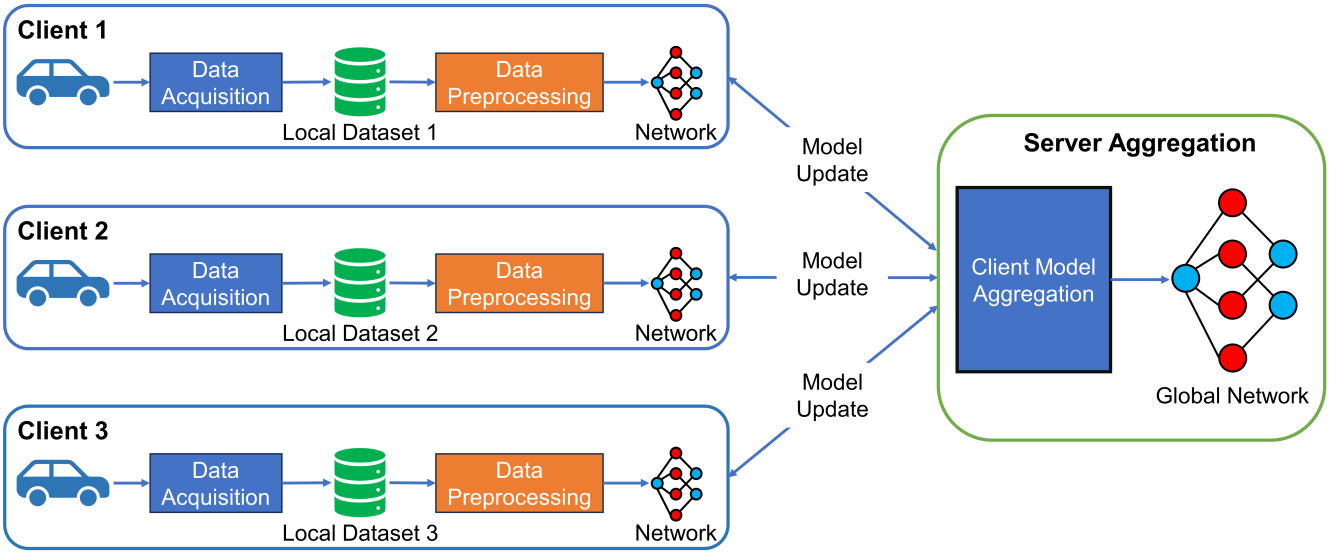


FIGURE 3. Federated learning process framework for anomaly detection using electric power steering multivariate time series data. This framework involves a network that represents the USAD model, which was trained across federated clients and then aggregated within the federated server.

scenario, two primary components are involved: 1) federated clients and 2) a federated server. The overall architecture of FL with USAD is presented in Figure 3. The process entails gathering data from individual federated clients, implementing meticulous data preprocessing techniques to enhance the overall data quality, and subsequently conducting FL training.

1) FEDERATED CLIENTS

In this scenario, each federated client represents a vehicle within the EPS system. Each federated client has its own dataset, referred to as the local dataset, which is acquired from various sensors installed in the vehicle and is used to construct multivariate time series data. In centralized methods, all these data need to be collected or transferred to a central server, which is a time-consuming and tedious task. However, using the FL technique, the need for data collection is eliminated. Also, each federated client uses its local dataset to train a model known as the local model [22]. Details of the federated client USAD local model were described in the previous section. To ensure the quality of the federated client local dataset and the local model, a data preprocessing step is implemented.

The architecture of the local training process is presented in Figure 4. Initially, a scaling step is performed on the original local dataset using the min-max scaling function. This step normalizes the data and brings them within a specific range. The dataset is then split into train and test subsets, where 70% of the data are allocated for training and the remainder 30% for testing. Both the train and test sets are used for prediction; the prediction results are employed to calculate the threshold value. The threshold value is calculated using the statistical method that is based on extreme value theory (EVT) [23], [24]. This value serves as a reference point for determining whether a data point is normal or an anomaly. Specifically, the loss function value of the local model is compared with the threshold value. If the loss function value is greater than the threshold value, this particular data point is an anomaly. Conversely, if the loss function value is below the threshold value, the data point is considered normal. Finally, the trained local model is uploaded on the federated server for aggregation and evaluation.

2) FEDERATED SERVER

The federated server is crucial in coordinating the training efforts of federated clients. Its main tasks include model initialization and client model aggregation, which are performed to construct the global model and then sent back to federated clients for further training. The aggregation process is performed by the federated averaging (FedAvg) function [7]. The steps performed to achieve communication between the federated clients and the federated server are described below:

1) **Model Initialization:** The federated server initializes global model parameters to set the starting point of the training process. In this step, the model hyperparameters are carefully set to an initial state that enables effective learning. Once the model weights are initialized, this initial model is shared with the federated clients to start the training process. The local model on each federated client is trained for a few epochs, and the updated parameters of the local models on all federated learning clients are then sent to the connected federated server.
2) **Model Aggregation:** Once the federated clients transmit their trained model parameters (w_i) to the federated server, the server initiates the process of model

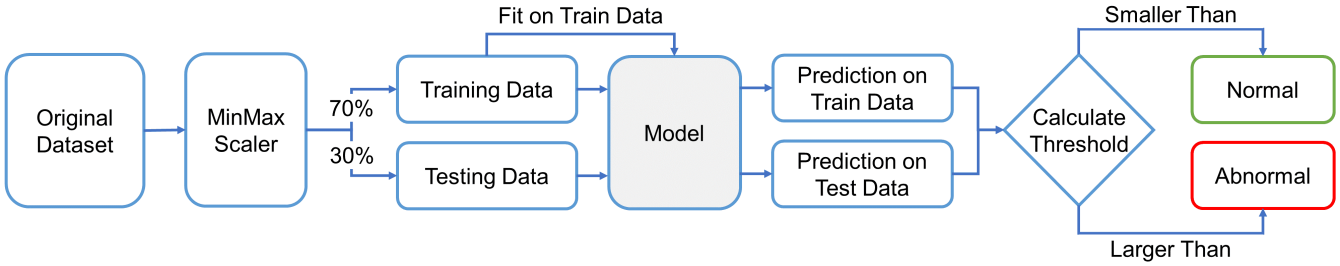


FIGURE 4. Overall process for the proposed federated learning setup. Overall data processing: original datasets, min-max scaling, a split of local datasets, use of local models to predict both train and test sets, and anomaly detection.

aggregation to form the global model parameters (w). The parameters of the global model are employed to evaluate its performance in the federated server. The model parameters aggregation is accomplished using a federated averaging technique and performing careful computation as follows:

$$w = \sum_{k=1}^{N} \frac{1}{k} w_k \tag{8}$$

where w is the USAD model parameters, N represents the number of vehicles or federated clients in the EPS system, and k represents the index k^{th} of the federated clients. The federated clients are required to send trained model parameters to the federated server.

3) **Global Model Broadcasting:** After aggregating and averaging the model parameters, the federated server evaluates the global model and transmits it back to all federated clients. The clients replace their local models with the updated global model to start a new training round. This ensures that all clients benefit from the improved model and can collectively continue the training process.

The training process continues until a predetermined number of iterations (*itrs*) is reached. With each iteration, the global model progressively enhances its detection performance, benefiting from the collaborative efforts of federated clients. This iterative approach improves the robustness and performance of the global model over time. Algorithm 1 presents a high-level pseudocode description of the FL training process, which consists of local data, aggregation of model parameters, federated clients, and the federated server.

Algorithm 1 Federated Learning Training Process

Data: Training data X_i on each federated client
Result: Model parameters w
Initial local and global models
Load and preprocess data on local federated clients
for $i \leftarrow 0$ **to** *itrs* **do**
    Get parameters of global models and update local models on federated clients
    Train local models on federated clients for e epochs
    Connect to federated server and send model parameters to federated server
    Aggregate model parameters from all federated clients on a federated server and update parameters of global model
    Evaluate the performance of the global model
end

IV. EXPERIMENTAL SETUP

This section provides a detailed overview of the experimental process used to validate the performance of the proposed approach. It also describes the characteristics of the dataset used, the methodology for simulating the FL performance, and the evaluation metrics employed.

A. DATA DESCRIPTION

We conducted an extensive experimental validation of the proposed model performance using data obtained from a dedicated test jig employed in the EPS system. The dataset used consists of three essential parameters: vehicle speed, steering angle, and torque data, as shown in Table 1. During the experiments, the vehicle speed was varied between 0 and 60 km/h, and the data were recorded at precise 10-ms intervals. To ensure the robustness of the anomaly detection model, we collected data from a diverse set of four vehicles, resulting in a substantial dataset of 460,000 samples.

TABLE 1. EPS dataset description.

Fields	Description
Date	The timestamp associated with the data collection.
Speed	The speed refers to the velocity of the steering wheel movement over time in EPS system.
Angle	The angle denotes the rotational displacement or orientation of the steering wheel relative to a reference point in EPS system.
Torque	Torque represents the rotational force applied to the steering mechanism by the EPS system in response to driver input.

B. ANOMALY LABEL GENERATION

In this study, we employed unsupervised learning to blindly train our model. However, to perform meaningful

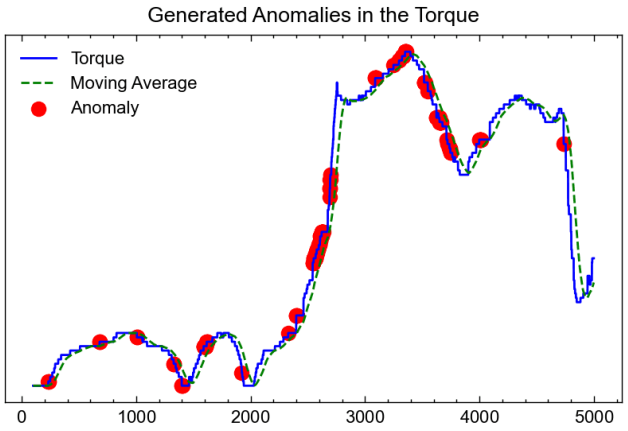


FIGURE 5. Generating anomalies in EPS test datasets with 5000 data points by applying the moving average technique. The true torque data is illustrated by the blue line, whereas the green dashed line represents the moving average values obtained using a specific window value.

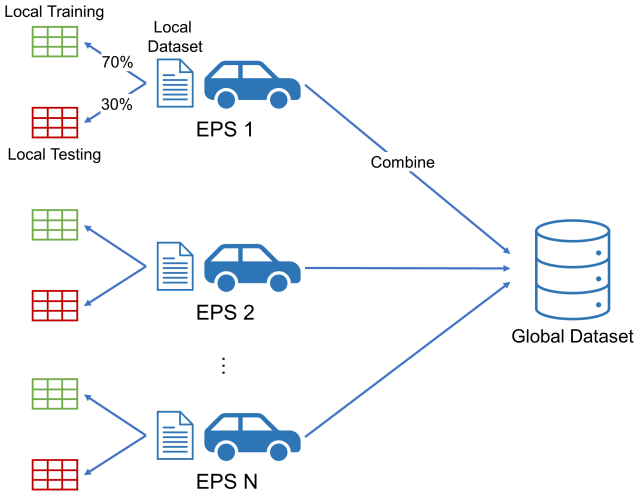


FIGURE 6. Preparation of data combination before starting Federated learning (FL) simulation. Local datasets were employed in each FL client, while the aggregated dataset was subsequently utilized in the federated server by the global model. EPS denotes electric power steering.

comparisons during testing, we utilized the moving average technique to facilitate anomaly detection for metric evaluation [25]. This technique involved two key steps: first, we computed deviations from the dataset, following which we determined the dataset's standard deviation. Notably, the threshold parameter was set to twice the standard deviation of the dataset. This selection implies an encompassing of around 99.7% of the data within three standard deviations following the 68-95-99.7 rule [26]. As a result, we generated a significant set of anomaly labels for the test dataset, aiming to evaluate the anomaly detection performance of our proposed method. In Figure 5, the notable deviations between the actual torque data and the MA values are identified as anomalies. The generated anomalies undergo manual adjustments to eliminate false anomalies, ensuring that only genuine anomalies remain. It is worth noting that these scenarios are not representative of real-world situations; however, they are utilized for comparison to obtain evaluation results.

C. FEDERATED LEARNING SIMULATION

The experiment was conducted utilizing the Windows 11 operating system running on an Intel Core i7-12700F CPU, accompanied by an NVIDIA GeForce RTX 3060 Ti 8–GB GPU, and supported by 64 GB RAM. The setup assumed no network errors between federated clients and the federated server, as all operations were confined within the same machine. To simulate the FL scenario, each client retained its individual dataset, with 70% for training and 30% for testing. These datasets were then combined for testing and evaluating the performance on the server using a global model, as illustrated in Figure 6. The combined dataset was utilized for evaluating models in both centralized and FL. In our experiment, we utilized five simulators; one simulator acted as the federated server, while the other simulators functioned as EPS-equipped vehicles, representing the federated clients [15]. The federated clients and the federated server were trained in parallel, meaning that all federated clients contributed to the improvement of the global model. Given that our experiment was conducted on a single machine, we were unable to directly measure the communication overhead. Nevertheless, we made an informed assumption by considering the dataset size for centralized training and the model weight size for FL training [27].

D. MODEL HYPER-PARAMETER

We configured the input features of the USAD model to include three parameters: speed, angle, and torque. Using the sliding window technique, we set the window size to 5, which was then multiplied with the input features to enhance data processing. The dataset was then normalized using a min-max scaler into a range of 0 to 1. The USAD number of hidden layers and latent are set to 16 and 5, respectively. Within the USAD architecture, we employed the rectified linear unit (ReLU) activation function for both the encoder and decoders. However, at the final layer of both decoders, we opted for the Sigmoid activation function instead. The model was trained using the mean-square-error (MSE) loss function, and Adam optimizer with a learning rate of 0.001 and a step scheduler size of 0.5 [28]. For all models, we set the number of epochs to 20. However, for the FL model, we set the training rounds count to 20 by fine-tuning the parameters to achieve optimal results for the global model.

E. EVALUATION METRICS

We conducted a comprehensive evaluation of our approach, employing a set of four key metrics to evaluate its performance across three distinct model variants. The metrics consist of prediction accuracy, anomaly detection performance, training time, and communication overhead.

- **Torque data prediction performance**: We conducted a rigorous model training using EPS multivariate time series data. We initially focused on the prediction of torque value and then on the prediction of the motor-assistance torque value. To assess the performance of our model, we employed the root mean square error (RMSE) metric, which effectively measures the difference between predicted results and actual ground truth values. Additionally, we performed a comprehensive comparison among all three models to thoroughly evaluate their performance.
- **Anomaly detection performance**: We incorporated the widely used Precision, Recall, F1-Score, and AUC evaluation metric, which is derived from a confusion matrix. This is a standard metric particularly suited to the assessment of classification models. The F1-Score values are in the 0–1 range, where values close to 1 indicate a superior performance. The F1-Score is defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{11}$$

  where true positive (TP) denotes correctly predicted abnormal samples, false positive (FP) denotes normally predicted samples falsely identified as abnormal, and false negative (FN) denotes abnormal samples incorrectly classified as normal.
- **Model training time**: This refers to the time required for the model training process to complete. In the case of the local model and FL training, we calculated the average training time by considering the time required for training in four different vehicles in each training round.
- **Communication overhead**: This metric was used to quantify the amount of data that needs to be transferred during centralized and FL training. However, in local model training, the communication overhead cannot be calculated because there is no communication exchange involved.
- **Sensitivity analysis**: We further investigate the impact of dataset size for training and testing within our proposed approach. This investigation provides insights into the optimal configuration of the dataset size ratio used for training the model.

The three model variants under consideration are the centralized learning model, the client learning model, and the federated learning model, as well as other state-of-the-art methods.

- **Centralized learning model (Central ML)**: This model was trained using a centralized learning approach, where all vehicle local data were combined to create a fully comprehensive training dataset.
- **Client learning model (Client ML)**: This model was also trained using a centralized learning approach, where each vehicle's local dataset was leveraged for training instead of a combination. During the training process, there was no exchange of model parameters, which reduced communication overhead.
- **Federated learning model (FL)**: This model employs FL training, which exploits the potential of each vehicle's local dataset for training. During the training process, only the local model parameters were exchanged between the clients and the server.

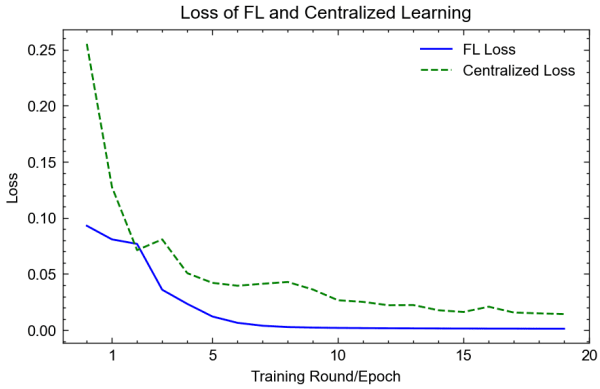


FIGURE 7. Training loss function is evaluated over 20 epochs and 20 rounds for FL.

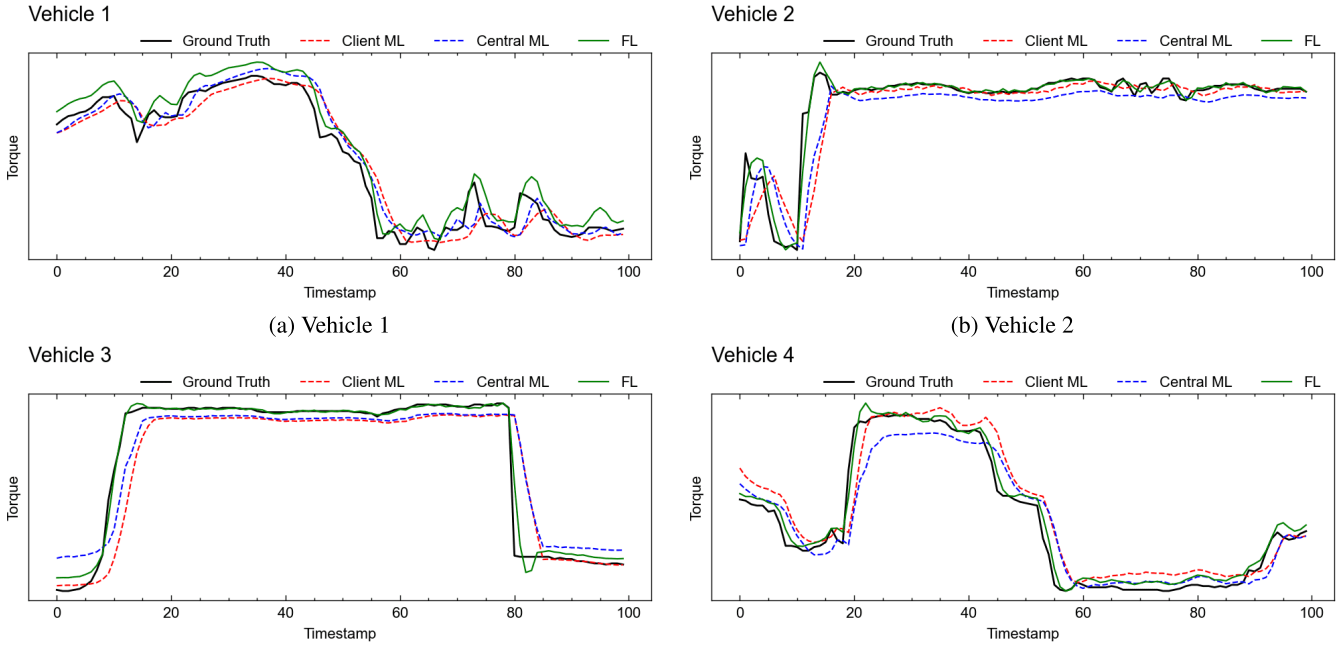
V. EVALUATION

In this section, we present the experimental results of the proposed FL technique for anomaly detection.

A. REGRESSION PERFORMANCE

We conducted a regression evaluation using the proposed FL model and other models. The evaluation was based on the RMSE for all vehicles. In Table 2, we present the RMSE values for all three models across the entire vehicle dataset. The proposed FL approach achieved significantly reduced error rates for different vehicles compared with Central ML and Client ML. FL reduced the RMSE of Vehicle 1 by 24% compared with Client ML and by 40% compared with Central ML. The corresponding reductions in RMSE for Vehicle 2 were 1% and 18%, respectively. For Vehicle 3, the reductions were 53% and 32%, respectively, and for Vehicle 4, the reductions were 22% and 43%, respectively. Overall, FL reduced the error rate by 25% compared to Client ML and by 33.25% compared to Central ML. These results demonstrate the robustness of the proposed FL approach for anomaly detection systems.

Figure 7 demonstrates that the FL loss function stabilizes after 5 training rounds, showcasing its ability to achieve stability. In contrast, centralized learning does not appear to reach the lowest loss function on the same dataset. One



(a) Vehicle 1 (b) Vehicle 2

(c) Vehicle 3 (d) Vehicle 4

FIGURE 8. Performance comparison of client learning models (Client ML), centralized learning models (Central ML), and federated learning models (FL) in terms of torque sensor data prediction using four local vehicle test datasets. The ground truth torque values are represented by the black line, and the values predicted by the three models are represented by the other colored lines. The lines close to the black line indicate high prediction accuracy.

TABLE 2. Torque sensor data regression error (RMSE) on the test set of each vehicle and model.

Models	Vehicle 1	Vehicle 2	Vehicle 3	Vehicle 4
Client ML	0.0164	0.0301	0.0764	0.0526
Central ML	0.0143	0.0362	0.0525	0.0733
FL	0.0141	0.0298	0.0354	0.0412

possible explanation is that FL trains using local data from all federated clients, leveraging information from multiple sources simultaneously, as these clients undergo 20 epochs and 20 rounds within the federated server. This collaborative approach contributes to the observed stability of FL compared to centralized learning.

B. PREDICTION PERFORMANCE

We have presented detailed results that demonstrate the low RMSE error rate of our model. The low error rate in regression confirms that our model performs remarkably well in making accurate predictions for the test set. Here, we evaluate the prediction performance of our FL approach. In Figure 8, we illustrate the torque prediction performance for three different models using 100 timestamp data points. The results demonstrate that FL achieves the same level of accuracy as other centralized methods. Moreover, FL significantly improves model generalizability when applied to all vehicle datasets, making the model superior in terms of prediction performance compared to both Central ML and Client ML in some instances. The results indicate that the potential of FL extends beyond privacy-preserving and data transfer reduction for various applications.

C. ANOMALY DETECTION PERFORMANCE

The proposed FL model performance in anomaly detection was evaluated using a confusion matrix and compared with two other centralized models. The classification performance metric employed was the F1-Score. The results showed that the FL model performance is comparable to that of centralized approaches in accuracy, demonstrating its potential for anomaly detection while eliminating the need for data transfer. A comparison of the results is illustrated in Figure 9. We observe that for Vehicle 4, the F1-Score for the Central ML model is affected by the Vehicle 4 data being used for testing. However, for the FL model, all local models of each vehicle are aggregated into the global model, making it more generalized for that particular vehicle. The performance of the FL model was comparable to the Central ML and Client ML models, achieving higher F1-Scores: 98.55% for Vehicle 1, 99.35% for Vehicle 2, 98.96% for Vehicle 3, and 99.18% for Vehicle 4. Client ML scored 98.04%, 98.88%, 99.66%, and 98.98%, respectively, and Central ML scored 97.65%, 98.75%, 99.73%, and 97.09%, respectively. Overall, the FL model achieved 99.01%, Central ML achieved 98.31%, and Client ML achieved 98.89% on average in terms of F1-Score. However, FL often faces challenges in achieving high accuracy in particular scenarios, primarily attributed to the diversity of datasets, variations in dataset quality, the

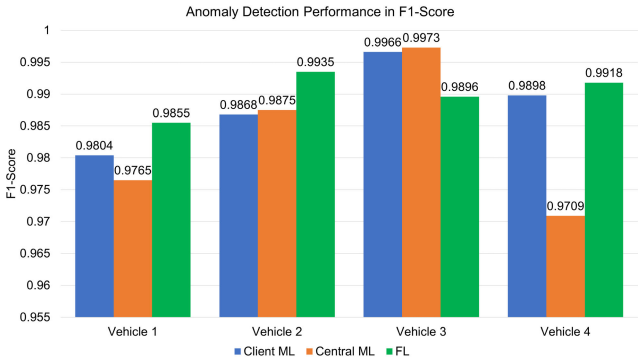


FIGURE 9. F1-Score comparison of the three models used for the local datasets of four vehicles. A high F1-Score value indicates better performance. However, the performance of our method is lower than that of other models in the case of Vehicle 3. This may be due to the varying data distributions in the federated learning setup. Client ML denotes a client learning model; Central ML is a centralized learning model; and FL is a federated learning model.

TABLE 3. Anomaly detection performance comparison of our proposed FL with other centralized methods in terms of precision, recall, F1-score, and AUC score.

Models	Precision	Recall	F1-Score	AUC Score
AE	0.9469	0.9783	0.9525	0.9639
DAGMM	0.9621	0.7421	0.8379	0.8707
OmniAnomaly	**0.9999**	0.9078	0.9516	0.9539
Central ML	0.9796	0.9533	0.9662	0.9764
Our FL	0.9660	**0.9953**	**0.9804**	**0.9972**

TABLE 4. Training time and communication overhead with different models of all vehicles in total.

Models	Training Time (Sec)	Communication Overhead (MB)
Client ML	970	-
Central ML	4,587	9.00
FL	1,260	0.82

number of datasets across federated clients, and the choice of aggregation functions. However, through the execution of our proposed FL method, wherein all federated clients collaboratively utilize a shared dataset, maintain an equivalent number of datasets, and undergo consistent data processing to uphold uniform dataset quality, we can achieve a commendable level of accuracy comparable to centralized learning. This allows us to leverage the advantages fundamental in FL characteristics.

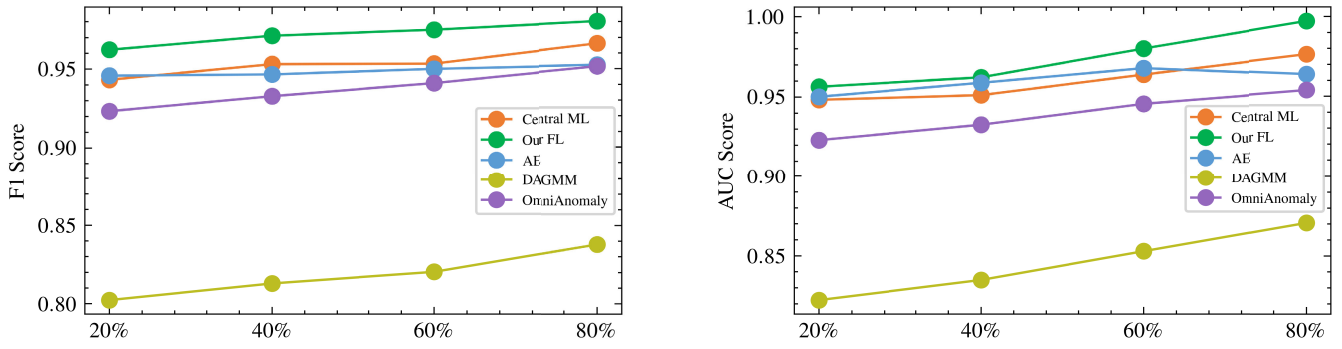
Furthermore, to demonstrate the overall performance of our proposed model, we compare Central ML and FL with other state-of-the-art unsupervised models for the detection of EPS sensor anomalies: AE, DAGMM [29], and OmniAnomaly [19]. These are renowned for their exceptional performance in anomaly detection, compared to the original USAD methods, due to their superior capabilities. However, none of these methods provide a threshold value to detect anomalies. Therefore, we implemented a mechanism to automatically and dynamically find the anomaly threshold value using the peaks-over-threshold technique [23] and conduct a comparison using Precision, Recall, F1-Score, and AUC score. Table 3 details the obtained anomaly detection performance results for all models. This outperformance can be attributed to FL access to comprehensive data from federated clients, contrasting with centralized models, which are limited to only 70% of the combined data. Moreover, for time series analysis, temporal information is not just important but indispensable. Observations within a time series are inherently interdependent, making historical data crucial for reconstructing current observations accurately. In the context of USAD, temporal relationships are vital for both training and detection. The input comprises a sequence of observations, ensuring the retention and utilization of this temporal context throughout the analysis.

D. TRAINING TIME AND COMMUNICATION OVERHEAD

Here, we compared the total training time and communication overhead of the FL model with those of two baseline models. The FL model significantly reduces training time compared with Central ML, though slightly increases training time compared with Client ML due to model exchange during training. Table 4 shows a performance comparison between the FL model and the other two models in terms of training time and communication overhead. We observe a significant reduction in training time for the FL model compared with that of Central ML because the FL model performs simultaneous training. In terms of communication overhead, often expressed as a function of data volume (e.g., megabytes or MB), the proposed FL model demonstrates high efficiency. This is attributed to its practice of only transferring model parameters, as opposed to Central ML, which necessitates collecting data from all vehicles, leading to increased communication overhead. We quantify the communication overhead in FL by computing the number of model parameters for each federated client and multiplying it by the number of training rounds. In contrast, for Central ML, we calculate the communication overhead based on the size of the dataset file that needs to be collected from various clients. However, the communication overhead for Client ML cannot be precisely determined due to local data in the client vehicles. Nevertheless, the FL model achieves a remarkable 72.5% reduction in training time and a 90% reduction in communication overhead compared with Central ML.

E. SENSITIVITY ANALYSIS

Figure 10 illustrates the variation in F1 and AUC scores across all models as the ratio of training data used for model training varies, spanning from 20% to 100%. As the dataset size grows, we can clearly see an improvement in anomaly prediction performance. Notably, across all ratios, the FL



(a) F1 Score (b) AUC Score

FIGURE 10. F1 and AUC scores dataset size ratio.

model consistently outperforms others with higher F1 and AUC scores.

VI. CONCLUSION

In this paper, we presented a highly successful integration of the USAD model with the FL framework for anomaly detection using time series data, specifically focusing on an EPS sensor dataset. Our experiments involved three distinct models: centralized (Central ML), on-device (Client ML), and FL models. The experimental results demonstrated the remarkable performance of the proposed FL approach in accurately predicting the time series data, outperforming the other two variant models with a notably lower error rate between 25% and 33.25%. This improvement can be attributed to the aggregation of local models from all federated clients, which significantly enhances the overall generalizability of the model. Furthermore, the FL model proved its ability to achieve improved anomaly detection results compared with the other two baseline models. Specifically, its anomaly detection performance was 99.01% in terms of F1-Score. One of the key strengths of the FL model is its ability to reduce training time by training local models simultaneously while also minimizing communication overhead because the data remain decentralized and are not transferred to a centralized server. The FL model achieved a 72.5% reduction in training time and a 90% reduction in communication overhead. Additionally, the FL framework ensures data anonymity and eliminates the need for data transfer, thus adding an extra layer of security. Nevertheless, it remains crucial to select trusted federated clients to uphold the system integrity against potential security risks. As part of our future work, we aim to investigate and implement additional techniques, such as blockchain, to improve further the connection security between the federated server and the clients.

CODE AVAILABILITY

The code that supports the findings of this study is openly available in the Github repository, https://github.com/QCL-PKNU/FL-AD-EPS.

REFERENCES

[1] L. W. Zhang, R. Du, and Y. Cheng, "The analysis of the fault of electrical power steering," in *Proc. MATEC Web Conf.*, vol. 44, 2016, p. 02003.

[2] W. Kim, Y. S. Son, and C. C. Chung, "Torque-overlay-based robust steering wheel angle control of electrical power steering for a lane-keeping system of automated vehicles," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 4379–4392, Jun. 2016.

[3] W.-C. Lin and Y. A. Ghoneim, "Model-based fault diagnosis and prognosis for electric power steering systems," in *Proc. IEEE Int. Conf. Prognostics Health Manag. (ICPHM)*, Jun. 2016, pp. 1–8.

[4] K. Shaukat, T. M. Alam, S. Luo, S. Shabbir, I. A. Hameed, J. Li, S. K. Abbas, and U. Javed, "A review of time-series anomaly detection techniques: A step to future perspectives," in *Advances in Information and Communication*. Springer, 2021, pp. 865–877.

[5] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines," *IEEE Access*, vol. 9, pp. 120043–120065, 2021.

[6] A. A. Cook, G. Misirli, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020.

[7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[8] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecný, S. Mazzocchi, B. McMahan, and T. Van Overveldt, "Towards federated learning at scale: System design," in *Proc. Mach. Learn. Syst.*, vol. 1, 2019, pp. 374–388.

[9] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-IID data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, Nov. 2021.

[10] L. W. Alabe, K. Kea, Y. Han, Y. J. Min, and T. Kim, "A deep learning approach to detect anomalies in an electric power steering system," *Sensors*, vol. 22, no. 22, p. 8981, Nov. 2022.

[11] S. Kim and D.-Y. Jung, "Fault estimation of rack-driving motor in electrical power steering system using an artificial neural network observer," *Electronics*, vol. 11, no. 24, p. 4149, Dec. 2022.

[12] Y. Ji and H. Lee, "Event-based anomaly detection using a one-class SVM for a hybrid electric vehicle," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 6032–6043, Jun. 2022.

[13] A. Kavousi-Fard, M. Dabbaghjamanesh, T. Jin, W. Su, and M. Roustaei, "An evolutionary deep learning-based anomaly detection model for securing vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4478–4486, Jul. 2021.

[14] A. M. Elbir, B. Soner, S. Çöleri, D. Gündüz, and M. Bennis, "Federated learning in vehicular networks," in *Proc. IEEE Int. Medit. Conf. Commun. Netw. (MeditCom)*, Sep. 2022, pp. 72–77.

[15] H. Zhang, J. Bosch, and H. H. Olsson, "End-to-end federated learning for autonomous driving vehicles," in *Proc. IJCNN*, Jul. 2021, pp. 1–8.

[16] R. A. Sater and A. B. Hamza, "A federated learning approach to anomaly detection in smart buildings," *ACM Trans. Internet Things*, vol. 2, no. 4, pp. 1–23, Nov. 2021.

[17] J. Jithish, B. Alangot, N. Mahalingam, and K. S. Yeo, "Distributed anomaly detection in smart grids: A federated learning-based approach," *IEEE Access*, vol. 11, pp. 7157–7179, 2023.
[18] A. Nguyen, T. Do, M. Tran, B. X. Nguyen, C. Duong, T. Phan, E. Tjiputra, and Q. D. Tran, "Deep federated learning for autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2022, pp. 1824–1830.
[19] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2828–2837.
[20] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "USAD: Unsupervised anomaly detection on multivariate time series," in *Proc. KDD*, 2020, pp. 3395–3404.
[21] H. Chen, M. Liu, Y. Chen, S. Li, and Y. Miao, "Nonlinear Lamb wave for structural incipient defect detection with sequential probabilistic ratio test," *Secur. Commun. Netw.*, vol. 2022, pp. 1–12, Mar. 2022.
[22] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowl.-Based Syst.*, vol. 216, Mar. 2021, Art. no. 106775.
[23] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 387–395.
[24] X. Yang, E. Howley, and M. Schukat, "ADT: Agent-based dynamic thresholding for anomaly detection," 2023, *arXiv:2312.01488*.
[25] S. Hansun, "A new approach of moving average method in time series analysis," in *Proc. Conf. New Media Stud. (CoNMedia)*, Nov. 2013, pp. 1–4.
[26] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: Local outlier probabilities," in *Proc. 18th ACM Conf. Inf. Knowl. Manag.*, 2009, pp. 1649–1652.
[27] X. Wang, Y. Wang, Z. Javaheri, L. Almutairi, N. Moghadamnejad, and O. S. Younes, "Federated deep learning for anomaly detection in the Internet of Things," *Comput. Electr. Eng.*, vol. 108, May 2023, Art. no. 108651.
[28] Z. Zhang, "Improved Adam optimizer for deep neural networks," in *Proc. IEEE/ACM 26th Int. Symp. Quality Service (IWQoS)*, Jun. 2018, pp. 1–2.
[29] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–19.

KIMLEANG KEA (Graduate Student Member, IEEE) received the bachelor's degree in computer science and engineering from the Royal University of Phnom Penh, Phnom Penh, Cambodia, in 2020. He is currently pursuing the M.S. degree with the Department of AI Convergence, Pukyong National University, Busan, South Korea. His research interests include deep learning, embedded systems, and quantum computing.

YOUNGSUN HAN (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2003 and 2009, respectively. He was a Senior Engineer with the System LSI, Samsung Electronics, Suwon, South Korea, from 2009 to 2011. He was an Assistant/Associate Professor with the Department of Electronic Engineering, Kyungil University, Gyeongsan, South Korea, from 2011 to 2019. He is currently a Professor with the Department of Computer Engineering, Pukyong National University, Busan, South Korea. His research interests include quantum computing, compiler construction, microarchitecture, and high-performance computing.

YOUNG-JAE MIN (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2006, 2008, and 2011, respectively. In 2011, he was a Postdoctoral Researcher with the Semiconductor Research Institute, Korea University. In 2012, he joined the Memory Division, Samsung Electronics Corporation, Hwaseong, South Korea. In 2016, he founded SENTOUS Company Ltd., Seoul. He is currently an Assistant Professor with the Department of Electric and Electronic Engineering, Halla University. His research interests include high-speed CMOS transceivers and mixed-signal integrated circuits, including sigma-delta data converters and Nyquist-rate data converters.