Received: 1 April 2022 | Revised: 27 June 2022 | Accepted: 29 June 2022

IET Collaborative Intelligent Manufacturing

DOI: 10.1049/cim2.12057

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Privacy-preserving gradient boosting tree: Vertical federated learning for collaborative bearing fault diagnosis

Liqiao Xia[1] | Pai Zheng[1] | Jinjie Li[2] | Wangchujun Tang[3] | Xiangying Zhang[4]

[1]Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong Special Administrative Region, China

[2]Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong Special Administrative Region, China

[3]Department of Engineering, Institute for Manufacturing, University of Cambridge, Cambridge, UK

[4]School of Mechanical Engineering, Institute of Industrial Engineering, Zhejiang University, Hangzhou, China

Correspondence

Pai Zheng, Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong Special Administrative Region 999077, China.
Email: pai.zheng@polyu.edu.hk

Funding information

Hong Kong Special Administration Region, National Key R&D Programs of Cooperation on Science and Technology Innovation with Hong Kong, Macao and Taiwan, Grant/Award Number: SQ2020YFE020182; Ministry of Science and Technology of the People's Republic of China, Centre for Advances in Reliability and Safety; Mainland-Hong Kong Joint Funding Scheme, Grant/Award Number: MHX/001/20; AIR@InnoHK Research Cluster, and the Shanghai Rising-Star Plan (Yangfan Program); Innovation and Technology Commission; The Science and Technology Commission of Shanghai Municipality, Grant/Award Number: 22YF1400200

Abstract

Data-driven fault diagnosis approaches have been widely adopted due to their persuasive performance. However, data are always insufficient to develop effective fault diagnosis models in real manufacturing scenarios. Despite numerous approaches that have been offered to mitigate the negative effects of insufficient data, the most challenging issue lies in how to break down the data silos to enlarge data volume while preserving data privacy. To address this issue, a vertical federated learning (FL) model, privacy-preserving boosting tree, has been developed for collaborative fault diagnosis of industrial practitioners while maintaining anonymity. Only the model information will be shared under the homomorphic encryption protocol, safeguarding data privacy while retaining high accuracy. Besides, an Autoencoder model is provided to encourage practitioners to contribute and then improve model performance. Two bearing fault case studies are conducted to demonstrate the superiority of the proposed approach by comparing it with typical scenarios. This present study's findings offer industrial practitioners insights into investigating the vertical FL in fault diagnosis.

1 | INTRODUCTION

With the rapid development of the manufacturing industry, fault diagnosis, which detects fault occurrence and its categories, has received much attention [1]. Unexpected failures in manufacturing scenarios can result in unscheduled closure or shutdown of manufacturing equipment, leading to indirect economic losses and even dangerous accidents. Therefore, timely fault identification is crucial in manufacturing, as it prevents inefficiency and economic crises.

Various approaches, including the data-driven, physical-based, and knowledge-based models, have been applied in fault diagnosis [2]. Among the existing approaches, data-driven models have achieved significant results and are implemented broadly. Features have a strong impact on data-driven model performance. However, the feature collection process is not comprehensive enough due to the complicated manufacturing environment. Besides, the production process is multi-staged. One particular stakeholder can only acquire limited data from their own processes. Furthermore, some manufacturers are not aware of collecting data in the manufacturing process, leading to data scarcity.

To compensate for the shortage of data in the fault diagnosis model, the current research focuses on data augmentation, semi-supervised learning, transfer learning, few-shot learning, and other methods [3]. Although these methods have

shown excellent results in accuracy and response time, adding realistic features to the models can outperform the accuracy. As is well known, the more real training data there is, the higher the accuracy of the diagnostic model.

Optimistically, stakeholders should merge their machines or processes' data to train a model collaboratively to achieve a win–win situation in fault diagnosis. In practice, however, data owners are cautious about exposing their information to others due to privacy concerns and potential conflict of interests in market competence. For instance, the revealed manufacturing data can be exploited to derive the original equipment settings reversely, and by increasing legislation and protocols to safeguard the security of industrial data [4]. One approach can be conveying the data in a desensitised and anonymous form beforehand, but the previous research has suggested this approach is vulnerable to re-identification attacks [5].

Federated learning (FL) is a novel technique that seeks to resolve privacy concerns while simultaneously providing a feasible solution for collaborative fault diagnosis. Federated learning requires multiple participants to train a data-driven model collaboratively without data leakage. In FL, multiple decentralised clients train the original base model with their own data, and then local clients send the model information to a central server via a privacy-preserving protocol. After receiving the information from clients, the server securely aggregates the model information to update the model and transmit it back to local clients, as shown in Figure 1. During this learning phase, clients can only provide their model information rather than sharing their raw data.

Generally, FL has three major types: horizontal FL, vertical FL, and federated transfer learning. Figure 2 depicts the difference between horizontal FL and vertical FL. Horizontal FL trains the data with the same features from multiple clients, whereas vertical FL trains the data based on overlapped samples with limited common features. On the other hand, federated transfer learning integrates FL and transfer learning with the aim to train the multi clients' data that fails to contain overlapped samples or features. In many manufacturing scenarios (such as cloud manufacturing), the production of a single product requires the participation of multiple manufacturers, while these manufacturers provide the services without disclosing data information generated from the production. In this case, vertical FL can perfectly match this situation, where a particular product is manufactured by multiple manufacturers. If one manufacturer is faced with feature scarcity in fault diagnosis, it can cooperate with other manufacturers to train a vertical FL model jointly by tracing the features in its production phase.

This work proposes a vertical FL model for fault diagnosis across numerous clients in manufacturing industry contexts that overcomes data silos and privacy concerns by combining a boosting tree with an Autoencoder. Three main contributions are summarised as follows:

(1) Provided a privacy-preserving vertical FL boosting tree model for fault diagnosis. This model allows the active client to collect extra features through the encrypted model information from passive clients, assuring data confidentiality and remarkable performance. To our knowledge, this is the first paper to introduce vertical FL for fault diagnosis.
(2) Integrated the Autoencoder model to increase the feature utilisation rate of passive clients. The Autoencoder creates more significant features based on the existing features, which contributes more features into the boosting tree model and improves performance.
(3) Conducted an experiment to compare different learning modes and FL models. This experiment proves the superiority of the FL scheme, the superiority of the proposed model, and the effectiveness of the Autoencoder.

The remainder of the paper is laid out as follows: Section 2 reviews the data-driven model's past application with the restricted label, collaborative fault diagnosis, and FL for fault

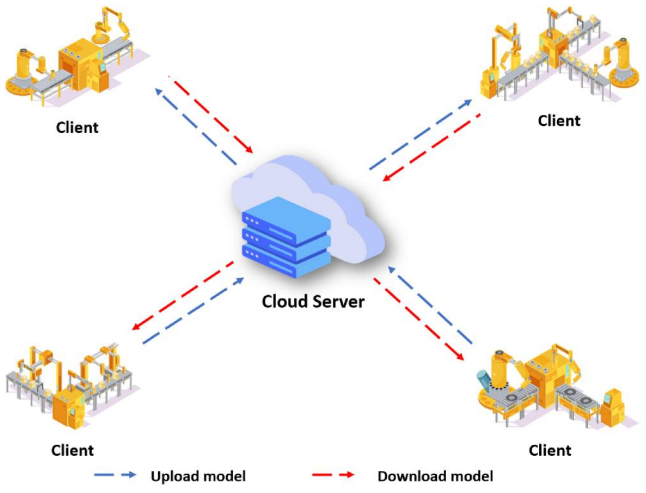

FIGURE 1 Federated learning (FL) framework

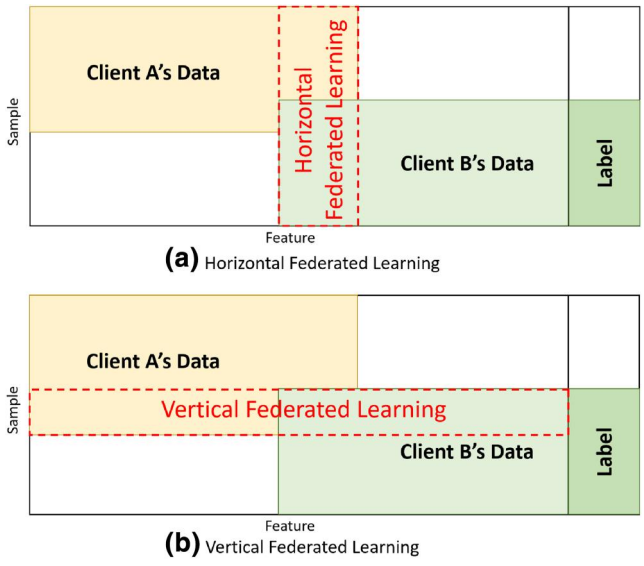

FIGURE 2 Horizontal Federated learning (FL) and vertical FL

diagnostics. The suggested histogram-based gradient boosting tree for diagnosis is introduced in Section 3, and it introduces the architecture of the proposed privacy-preserving boosting tree model. Also, the proposed model's security is examined in Section 4. Section 5 has two case studies to validate the proposed model, and the findings are discussed in Section 6. Section 7 summarizes this work and emphasises future directions.

2 | RELATED WORK

2.1 | Data-driven modelling approaches with limited data

Data-driven modelling approaches are widely used in many disciplines [6–10]. Therefore, data-driven approaches have been effectively applied in predictive maintenance [11, 12], especially fault diagnosis [13, 14]. Nonetheless, many real-world manufacturing tasks related data only have limited labels and features, causing underfitting and overfitting, and weakening the model's generalisation capacity [15]. To tackle this issue, researchers have proposed different methods.

First, data augmentation is a straightforward way to boost the data volume. Upsampling is one of the mainstream methods, which creates new samples by randomly selecting from scarce data [16]. Generative Adversarial Network is also a feasible method, which learns the characteristics of the raw data and then creates new samples by reinforcement learning [17].

Second, few-shot learning leverages the well-trained model to classify unknown new classes based on the limited data. A Siamese neural network-based few-shot learning model was proposed for bearing fault diagnosis by comparing the difference between the input sample pairs (the pairs may involve different label samples) [18]. Besides, zero-shot learning is a special situation, which mainly studies the no-sample situation for the target fault diagnosis task. Feng et al. [19] proposed an attribute zero-shot transfer learning based on a fault description that occurs in the target domain.

Third, transfer learning is a remedy to the inadequacy of data, which improves model performance in the target domain by leveraging the knowledge from the source domain. A typical transfer learning is a fine-tune, which transfers the parameters from the source domain model to the target domain model [20]. Meanwhile, domain adaptation is also an effective transfer learning way. Chen et al. [21] provided a domain adversarial transfer network to shift the source domain knowledge for training the model in rolling bearing fault diagnosis based on the feature similarity.

Furthermore, other approaches have been applied to tackle insufficient data problems, such as semi-supervised learning [22], unsupervised learning [23], self-supervised learning [24].

Despite the fact that the past approaches have obtained outstanding performance, some limitations also exist. First, the limited data restrict the training sufficiently for data-driven approaches. Second, many of them are single-source-based data-driven models, failing to become satisfactory generalised models.

2.2 | Collaborative fault diagnosis

Collaborative fault diagnosis requires different stakeholders to share their data, information, and knowledge to establish the diagnosis model collaboratively. Some diagnosis methods rely on experience knowledge, and a straightforward method is to collect all the rules into the central server for diagnosis [25]. Following a similar logic, a knowledge graph was introduced for all stakeholders to exchange their diagnosis experience, which established semantic associations in diagnosis knowledge [26]. In addition, an assist visualisation tool is developed to facilitate communication and reduce ambiguity among teammates [27].

Meanwhile, data-driven models from multiple participants can combine for ensemble learning [28]. Besides, in a multi-agent system, local models can merge into a global optimal model through reinforcement learning [29] or knowledge graph [30–33]. Moreover, Wang et al. [34] applied a blockchain in collaborative fault diagnosis, where the blockchain provided a decentralised platform and claimed the immutable ownership of data and knowledge of each participant.

Nevertheless, previous collaborative approaches do not consider privacy-preserving, limiting their broad application. It is noted that different manufacturers have their unique data, which can be used to improve the model performance by sharing data. How to overcome the data silos but still retain data security is an important challenge in diagnosis.

2.3 | Federated learning for fault diagnosis

In terms of privacy issues and data gathering costs, FL is a potential tool for using data from various clients in a decentralised and privacy-protected manner. FL allows many clients to use data while maintaining anonymity, ensuring model aggregation without data sharing [35]. Meanwhile, numerous methods are being developed to improve FL's performance [36, 37].

The study of FL in fault diagnosis is still in its early stages. Zhang et al. [38] developed a convolution neural network (CNN) in the FL framework for bearing diagnosis. In addition, CNN also integrates with FL for diagnosing in the Internet-of-Ships scenario, which reduced cryptography calculation and client communication times [39]. In order to identify inter-turn short-circuit defects, a stacked sparse Autoencoder paired with a Siamese network is presented as a solution to the limited label problem [40]. One of the most significant challenges in FL is that various clients have distinct characteristics and duties, which federated transfer learning seeks to address. To solve the domain shift phenomena in various industrial tasks, Zhang et al. [41] presented the CNN in federated transfer learning. In

addition, a previous distribution is used to bridge the domain gap in the early stages of federated transfer learning [42].

Previous methods combine FL with deep learning models for fault diagnosis, which need a mass of data for training, contradicting FL's motivation that each client lacks adequate data. Furthermore, FL with deep learning models necessitates that all clients use the same model architecture, which is difficult to predict in the early stages, and it cannot generalise across clients with variable sample sizes and data distributions. Furthermore, the aforementioned methods are horizontal FL or federated transfer learning, ignoring the vertical FL method, while vertical FL is suitable for many scenarios in the manufacturing industry.

3 | METHODOLOGY

3.1 | Problem statement

In this paper, vertical FL is investigated in fault diagnosis. First of all, only one client has labels in the vertical FL, denoting this client as an active client and other clients without labels as passive clients. Meanwhile, this research has several assumptions.

- Clients are all involved in the manufacture of a specific product or component.
- All the clients have the FL system, and the active client has insufficient data.
- The active client needs the passive clients' extra features to improve its model performance.

The active client has the label and it requires the model for fault diagnosis. Therefore, the active client needs to collaborate with passive clients for training together. At the same time, the motivation of passive clients is to make business income from the active client based on the feature contribution. Hence, the passive clients can generate more distinctive features to earn more income. However, creating features blindly may lead to raw feature usage decrease in the inference stage. Motivated by Secureboost [43], this paper combines a privacy-preserving boosting tree with an Autoencoder for vertical FL, as shown in Figure 3 where the Autoencoder creates a few but substantial features for the passive clients.

3.2 | Xgboost

Xgboost model is a kind of ensemble learning model, which combines weak models for regression and classification tasks. In addition, the Xgboost model selects the CART tree (Classification and Regression Trees) [44] as the weak model. The following equation can be used to estimate a final prediction:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \tag{1}$$

where K is the model numbers, $f_k \in \mathrm{F}$, and F represents different CART tree models.

Besides, Xgboost's loss function includes a second-Taylor expansion function in t^{th} iteration times:

$$L^t \simeq \sum_{i=1}^{n} \left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t), \tag{2}$$

where $l\left(y_i, \hat{y}_i^{(t-1)}\right)$ is the predefined loss function of the i^{th} sample between the target value y_i and predicted $\hat{y}_i$, $\Omega(f_t)$ is

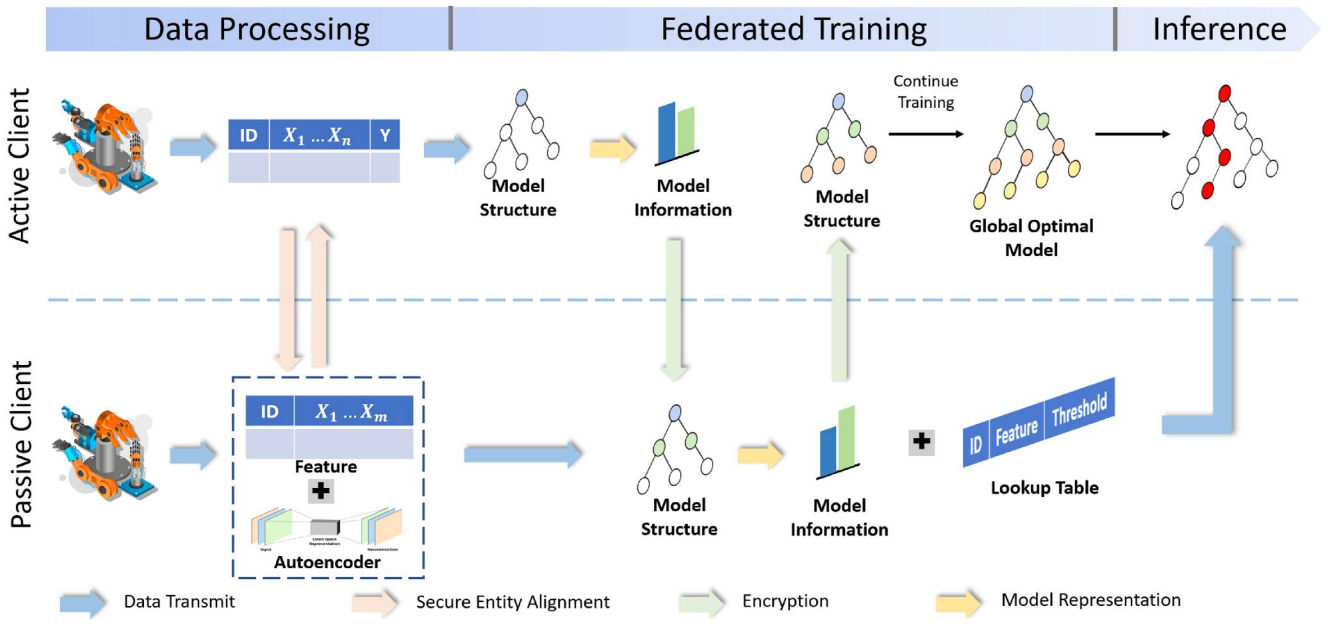

FIGURE 3 Boosting tree vertical Federated learning (FL) framework

the regularisation item, g_i is the gradient of loss function, and h_i represents the hessian of loss function.

Based on obtained gradients g_i and hessian h_i, the optimal weight of j^{th} can be achieved by:

$$w_j = -\frac{G_j}{H_j + \lambda}, \tag{3}$$

where λ is a constant and $G_j = \sum_{i \in I_j} g_i$ and $H_j = \sum_{i \in I_j} h_i$ are the summation of gradient and hessian of each sample indices I_j, respectively.

In addition, the Xgboost training includes the determination of the optimal leaf node split point. A gain score function is used to calculate the optimal weight w_j^*:

$$Gain\_Score = -\frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{(H_L + H_R) + \lambda}\right] - \gamma, \tag{4}$$

where G_L and H_L denote the summation gradient and hessian of the particular index of the left node, respectively; similarly, G_R and H_R correspond to the right node.

3.3 | Paillier cryptosystem with homomorphic encryption

The communication between clients in the vertical FL must be encrypted. This work introduces the homomorphic encryption (HE) paradigm to protect data privacy. The main elements of HE are homomorphic multiplication and addition:

$$[[x_1]] \otimes [[x_2]] = [[x_1 \times x_2]], \tag{5}$$

$$[[x_1]] \oplus [[x_2]] = [[x_1 + x_2]], \tag{6}$$

where [[.]] represents the encryption process. With HE, clients can calculate in encrypted form. In addition, this paper chooses the Paillier cryptosystem to encrypt the information, as shown below:

$$[[m]] = g^m r^n \bmod n^2, \tag{7}$$

where r is a random non-negative integer from $(0, n-1)$. The Paillier cryptosystem is suitable for the situation where a particular cleartext may have many different ciphertexts making it much more resilient.

3.4 | Autoencoder

To increase the feature utilisation rate of passive clients, an Autoencoder model is introduced to generate extra significant features for training collaboratively. An Autoencoder is a special neural network architecture whose input and output are the same architecture. The Autoencoder model involves two parts: the encoder part captures input data and transforms it into a low-dimension representation, and the decoder part tries to reconstruct the previous low-dimension representation as to the original value (Figure 4), and the equation is as shown below:

$$\begin{aligned}\mathcal{L}(x, x') &= \|x - x'\| \\ &= \left\|x - \sigma'\left(W'(\sigma(Wx + b)) + b'\right)\right\|,\end{aligned} \tag{8}$$

where σ denotes the activation function, W represents the weight matrix, b stands for the bias in the encoder process, while σ', W', and b' indicate the same meaning in the decoder process.

The training process minimises the loss function, making x and x' as close as possible. Based on the well-trained model, the middle layer, latent space representation is the output result. The passive clients can generate a few significant features based on the Autoencoder.

3.5 | Active client histogram construction

The active client needs to transmit the gradient information to pass clients in an encrypted manner for collaboratively training after finishing its Xgboost model training. According to Equation (3), the g and h can determine the weight of Xgboost, which can be used to represent the model information. However, it is likely to expose their data if the active client transmits the g and h to the passive clients directly. Therefore, the gradient information needs to be encrypted through Algorithm 1 before being sent to passive clients.

Algorithm 1 Active client split-info construction

Input: I: Instance space of current node; d: feature dimension; $[[g_i]], [[h_i]]_{i \in I}$: gradient and hessian;
Output: $G \in \mathbb{R}^{d \times l}, H \in \mathbb{R}^{d \times l}$
1: **for** k from 0 to d **do**
2: Split feature k by percentiles and get $S_k = s_{k,1} \ldots, s_{s,k}$
3: **end for**
4: **for** k from 0 to d **do**
5: **for** v from 0 to last bin index **do**
6: $G_{kv} = \sum_{i \in i | s_{k,v} \geq x_{i,k} > s_{k,v-1}} [[g_i]]$
7: $H_{kv} = \sum_{i \in i | s_{k,v} \geq x_{i,k} > s_{k,v-1}} [[h_i]]$
8: **end for**
9: **end for**

Line 2 indicates the process of splitting each feature in a binning manner, where the binning percentiles were calculated beforehand. In addition, line 4 to line 9 represent the summation of the gradient and hessian of each feature across their binning, respectively. Instead of transmitting $[[g_i]]$ and $[[h_i]]$ directly, the active client aggregates the encrypted gradient information into buckets. After the encryption of model

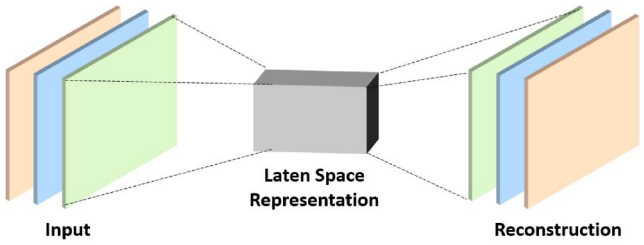

FIGURE 4 Autoencoder architecture

information, the passive client cannot know the raw data and the corresponding label, while the passive clients can conduct encryption calculations based on HE.

3.6 | Passive client histogram construction

After the training process in the active client, passive clients need to integrate their features and the received encrypted gradient information from the active client. Algorithm 2 shows the aggregation process.

Because passive clients do not have access to the raw label, they cannot calculate their features' gradient and hessian. However, the active clients have calculated the gradient and hessian information across the samples. In the first half, the gradient and hessian will be summed up based on each feature's bin index (line 1 to line 6). For example, x_1, the value of vibration is 0.5. The vibration has calculated its split point before, which is (0, 0.7), and (0.7, 1). Therefore, the bin index is (0, 0.7) in this case, and the feature binning part can be accumulated through line 4.

Algorithm 2 Passive client split-info construction

Input: I: Instance space of current node; X: Data sample g, h: gradient and hessian from active client; K: Clients set; F: Feature space
Output: g_i, $h_i \in I$:Encrypted split-info set
1: **for** Instance x_i in X_k, $i \in$ I **do**
2: **for** Feature f_j in F_k **do**
3: bid = x_i[j] //bin index
4: HD[j][bid][0,1]+ = [[g_i]], [[h_i]]
5: **end for**
6: **end for**
7: **for** Feature f_j in F_k **do**
8: **for** bid from 1 to last bin index **do**
9: HD[j][bid][0,1]+=*HD*[j][bid-1][0,1]
10: **end for**
11: [[g_1]], [[h_1]] = HD[j][bid][0,1]
12: **end for**

Afterwards, all the gradient and hessian information of each split index will be aggregated to the largest split index (line 9), which represents the left side information in the tree node. Under this circumstance, the gradient and hessian are encoded, which protects its information from being retrieved by the active client.

3.7 | Federated split finding

Integrating all the components, Algorithm 3 illustrates how the active client estimates the best split depending on the outcomes of the passive clients.

In Algorithm 3, line 2 sums the gradient and hessian from all clients. The split finding process enumerates all the clients (line 3), their features (line 4), and the corresponding split value (line7). Since the encryption was done by the active client before, this encrypted content can be deciphered in the active client as well (line8). Then the deciphered value can be aggregated as the left side gradient and hessian (line 9 and line 10). Correspondingly, the right side of the gradient and hessian can be achieved through the total gradient or hessian minus the left side gradient or hessian (line 11). Based on the gradient and hessian information, the gain score of the current situation can be calculated using Equation (4) (line 12). To obtain the best gain score, line 13 and line 14 represent recording the best score and the corresponding optimal binning index and feature name.

Algorithm 3 Split finding

Input: I: Instance space of current node; g^i, h^i: gradient information from all clients;
Output: k_{opt}, v_{opt}: Best feature with best split point
1: Initialise score = $-\infty$
2: G, $H = \sum_{i \in I} g_i$, $\sum_{i \in I} h_i$
3: **for** n from 0 to N **do**
4: **for** k from 0 to d_i **do**
5: //d_i is the features number of client i
6: $g_l = 0$; $h_l = 0$
7: **for** v from 0 last bin index **do**
8: $g^n_{k,v}$, $h^n_{k,v} \leftarrow \left[\left[g^n_{k,v}\right]\right]$, $\left[\left[h^n_{k,v}\right]\right]$
9: $g_l = g_l + g^n_{k,v}$
10: $h_l = h_l + h^n_{k,v}$
11: g_r, $h_r = G - g_l$, $H - h_l$
12: cur_score = max(score, $\frac{g_l^2}{h_l+\lambda} + \frac{g_r^2}{h_r+\lambda} - \frac{g^2}{h+\lambda}$)
13: **if** cur_score > score **then**
14: score = cur_score, k_{opt} = k, v_{opt} = v
15: **end if**
16: **end for**
17: **end for**
18: **end for**

3.8 | Inference

The nodes in the well-trained boosting tree model come from different clients, while only the active client can provide the raw data in the inference process. Therefore, a secure inference protocol is required. The active client coordinates the inference process because it seeks joint modelling vigorously with its unique label. The nodes in the FL boosting tree involve the client ID and feature ID, and the inference path can refer to the corresponding client for retrieving the nodes. For example, beginning from the root node, the client ID and feature ID are received, then the active client can find the threshold from the lookup table based on these two ID keys. Following this, the active client compares the current value with the threshold to determine which path to go (left or right). The active client continues to search the corresponding threshold from the lookup tables until it gets the final result in the leaf node, as shown in Figure 5.

4 | SECURITY ANALYSIS

4.1 | Encrypted model information

The central server in the proposed model cannot reverse inference of the original data using encrypted histogram-based gradient information, as explained below. The proportion of the major class in a single leaf is defined as leaf purity. The leaf purity of the first tree can be inferred from the weight of the leaves, which leads to data leakage. To simplify the problem, a binary classification is studied, with the following loss function:

$$L = y_i \log\left(1 + e^{-\hat{y}_i}\right) + \left(1 - y_i\right)\log\left(1 + e^{\hat{y}_i}\right), \tag{9}$$

where y_i and $\hat{y}_i$ represent the truth label and estimated label, respectively. Based on Equation (9), the gradient and hessian can be formulated as $g_i = \hat{y}_i^{(0)} - y_i$ and $h_i = \hat{y}_i^{(0)} * (1 - \hat{y}_i^{(0)})$. Equation (3) has a constant λ, which can be rewritten as

$$\begin{aligned} w_j &= -\frac{G_j}{H_j + \lambda} \approx -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i} \\ &= -\frac{\beta_j * n * \left(\hat{y}_i^{(0)} - 1\right) + \left(1 - \beta_j\right) * n * \hat{y}_i^{(0)}}{n * \hat{y}_i^{(0)} * \left(1 - \hat{y}_i^{(0)}\right)} \\ &= \frac{\hat{y}_i^{(0)} - \beta_j}{\hat{y}_i^{(0)}\left(\hat{y}_i^{(0)} - 1\right)}, \end{aligned} \tag{10}$$

where β_j denotes the percentage of major class in leaf j and n is the sample number. Based on Equation (10), the formula of β_j can be adapted as

$$\beta_j = \hat{y}_i^{(0)} - \hat{y}_i^{(0)}\left(\hat{y}_i^{(0)} - 1\right) w_j. \tag{11}$$

As the Equation (11) shows, β_j is affected by $\hat{y}_i^{(0)}$ and w_j, and $\hat{y}_i^{(0)}$ can be derived from h_i. At the same time, β_j indicates the leaf purify (max (β_j, 1 − β_j)). Hence, the first tree's leaf purify can be inferred from the boosting tree model's parameters of w_j and the model information (g_i, and h_i).

The first tree parameters, as well as the corresponding leaf purify, can provide sensitive information. As a result, the proposed privacy-preserving boosting tree encodes gradient information, ensuring that the $\hat{y}_i^{(0)}$ cannot be detected for data security purposes.

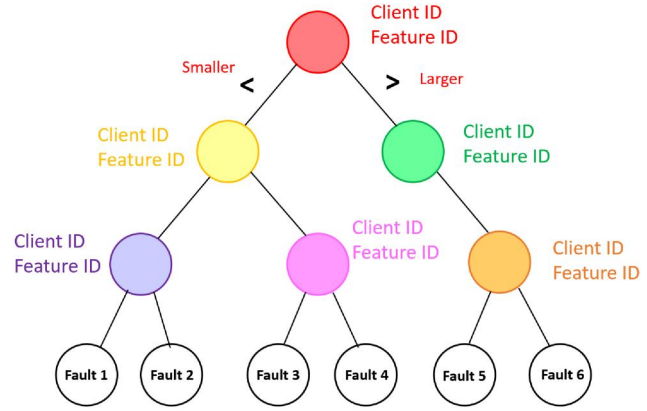

FIGURE 5 FL boosting tree inference process

4.2 | Clients communication

Active clients need to encrypt their histogram-based gradient information using the Paillier cryptosystem before sending it to passive clients. The Paillier cryptosystem is a probabilistic asymmetric algorithm for key cryptography. The key generation process is as follows:

4.2.1 | Key generation

(1) Select two large prime numbers, p and q, ensuring gcd (pq, (p − 1) (q − 1)) = 1, where gcd represents the greatest common divisor.
(2) Calculate n = pq, λ = lcm (p − 1, q − 1), where lcm denotes the lowest common multiple.
(3) Define $L(x) = \frac{x-1}{n}$.
(4) Randomly select a g smaller than n^2, and $\mu = (L(g^{\lambda} \bmod n^2))^{-1}$.
(5) The public key is (n, g), and the private key is (λ, μ).

4.2.2 | Encryption and decryption

Based on the key, the ciphertext can be generated by: $c = g^m r^n \bmod n^2$, where m is a positive integer smaller than n, and 0 <

$r < n$. Besides, the ciphertext can be decoded as $m = L\ (c^{\lambda} \bmod n^2)*\mu \bmod n$. Therefore, any client with g, p, and q can receive the ciphertext. Under this circumstance, the passive clients cannot reconstruct the raw data, guaranteeing the data security of the active client.

5 | CASE STUDY

This paper selects two representative data sets from Case Western Reserve University (CWRU)[1] and the Society for Machinery Failure Prevention Technology (MFPT)[2] to validate the proposed model. These two data sets are collected from authorities and have been adopted in much research, having universality and persuasiveness. Both benchmark data sets relate to bearing fault, and there are four kinds of conditions: healthy, inner race fault, outer race fault, and ball fault, as shown in Figure 6. Each fault in the CWRU data set has three damage levels: primary, intermediate, and severe. To summarise, there are ten labels in the CWRU data set (3*3 defect labels plus one healthy label). Meanwhile, there are just three labels in the MFPT data set: healthy, inner race fault, and outer race fault.

5.1 | Experiment setting

5.1.1 | Validation metrics

This paper adopts the precision, recall, and F1 scores to evaluate the models' performance based on the multi-classification scenario.

$$Precision = \frac{TP}{TP + FP}, \tag{12}$$

$$Recall = \frac{TP}{TP + FN}, \tag{13}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}, \tag{14}$$

where TP is true positive, FP is false positive, and FN is false negative.

5.1.2 | Different learning schemes

In order to validate the superiority of the vertical FL scheme, the non-FL schemes are provided, including global mode and local mode. The global mode represents all the data that are collected together for training without boundaries, including passive client and the active client. Besides, the local mode indicates that the active client trains the model using its own data alone.

It is noted that the vertical FL cannot be compared with the specialising limited data solutions, such as transfer learning and meta-learning. That is because some participants in vertical FL do not have labels, and their features have limited overlap, which restricts their application in many other methods.

All the experiments will be conducted in five trials to avoid the influence of some random factors.

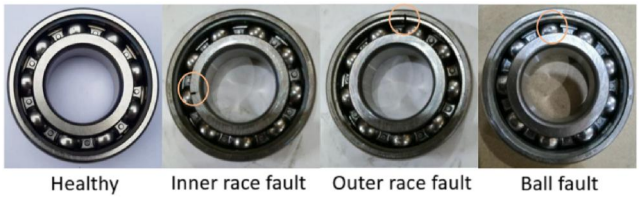

FIGURE 6 Four bearing condition [45]

TABLE 1 Hyperparameter of boosting tree

Hyperparameter	Value	Hyperparameter	Value
Learning rate	0.01	Max tree depth	6
Max iteration	100	Subsample	1
Min samples leaf	6		

5.1.3 | Model hyperparameter

To ensure an unbiased comparison, all the boosting tree models use the same hyperparameter, and the main hyperparameters are shown in Table 1.

5.1.4 | Sample and feature

To validate the robustness and generalisation, the sample numbers of the active client and passive client are 500, 1000, 2000, 4000, 6000, and 8000. Besides, each client has 25 features.

5.1.5 | Experiment environment

This global mode and local mode are conducted with 16G memory and an AMD Ryzen 7 5800H CPU, while the federated mode experiment is carried out in 4G memory with a cloud server.

5.2 | Different learning mode comparison

The federated mode is compared to the global and local modes in this section. The global mode represents the best

[1] https://engineering.case.edu/bearingdatacenter
[2] https://www.mfpt.org/fault-data-sets/

situation (upper limit) because it combines all the clients' data for model training without data privacy protection. At the same time, the local mode indicates that the active client cannot communicate with passive clients in any channel, and the active client can only train the model using its own data. It is noted that the global model and local model can be regarded as the comparison of existing Xgboost model.

In the CWRU data set, the FL mode performs significantly better than the local mode, as shown in Figure 7a–c. However, this huge advantages narrows in the MFPT data set (Figure 7d, e, f). Though the precision is high enough, the recall does not clearly distinguish it from the local mode. Hence, affected by the recall, the advantage of F1 values over the local mode is not as great as that in the CWRU data set (Figure 7f). One possible explanation is that the label types in MFPT are relatively small. As a result, the recall rate may fluctuate. On the contrary, the global mode achieves the best performance in all the metrics in these two data sets. It is comprehensible that the global mode can leverage all the raw data to establish the boosting tree model. In the CWRU data set, the FL mode is close to the global mode. Besides, the precision of FL mode in MFPT is promising while the recall stays at a low level, leading to the F1 not staying at the same level as the global mode. Meanwhile, each subfigure has a bar chart in the lower part, indicating the corresponding standard deviation of five trials. All the bar charts show no significant difference in the standard deviation between each group. Overall, the FL mode achieves the expected result in these two benchmark data sets, proving its generalisation ability.

5.3 | Different models comparison

After illustrating the superiority of FL, this paper compares the proposed model with other cutting-edge models, CNN, and deep neural network (DNN). In the experiment, one active client and one passive client participate in this vertical FL, where each client has 6000 samples.

According to Figure 8, the proposed model achieved the best results compared with CNN and DNN in the CWRU data set. Nonetheless, the proposed model performs slightly differently in the MFPT data set, as shown in Figure 9. Though the proposed model obtained the best result in both precision and F1 scores, it failed to obtain a promising result in the recall metrics. In addition, overall, both these bar charts have verified the superiority of the proposed model in vertical FL for fault diagnosis. The possible explanation is both the active client and the passive client have 6000 samples, which is insufficient for the deep learning model generally. Because the deep learning model has a relatively complex structure, it needs a large amount of data to determine the parameters.

5.4 | Autoencoder validation

In the vertical FL, the passive clients need to transmit their features to the active client in an encrypted manner. This paper applied the Autoencoder to generate extra significant features in passive clients. In this experiment, the Autoencoder generates three extra features for each sample (a total of 28 features).

Figure 10 and Figure 11 illustrate the proposed model with and without the Autoencoder in both the CWRU data

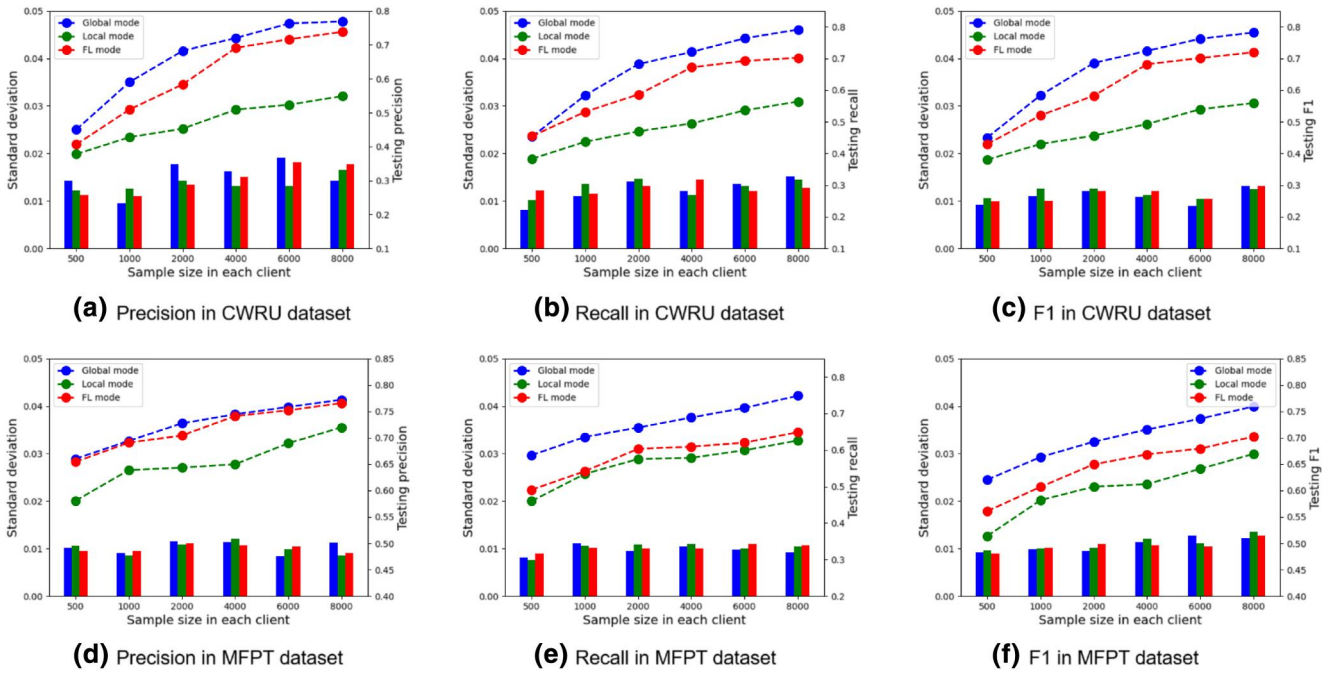

FIGURE 7 Precision, recall, and F1 scores in Case Western Reserve University (CWRU) and MPFT

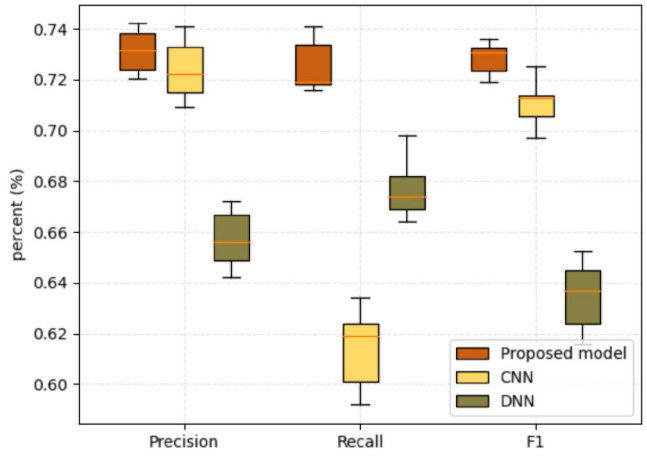

FIGURE 8 Different models comparison in Case Western Reserve University (CWRU)

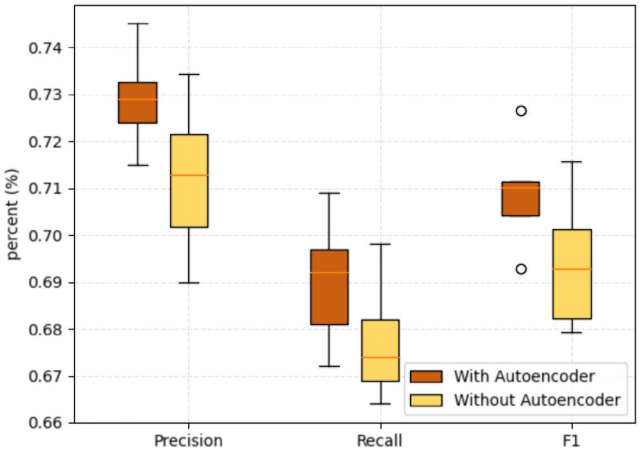

FIGURE 10 Model performance with/without Autoencoder in Case Western Reserve University (CWRU)

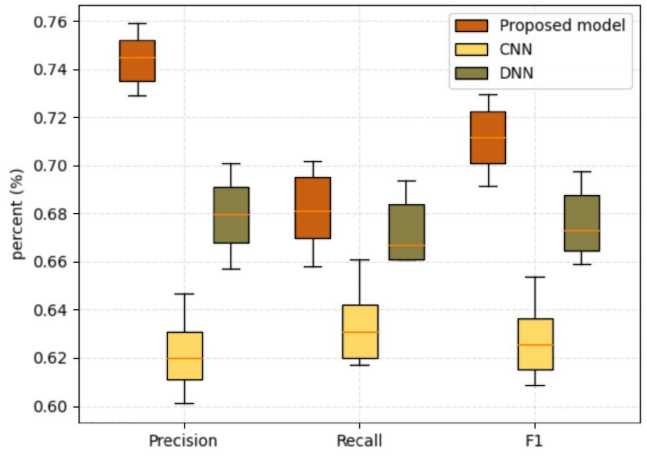

FIGURE 9 Different models comparison in Machinery Failure Prevention Technology (MFPT)

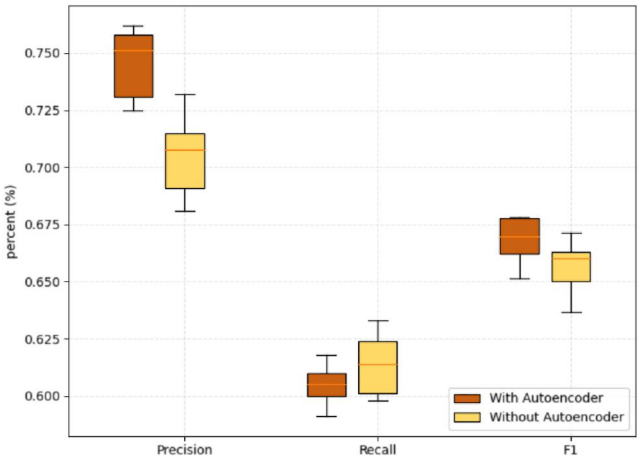

FIGURE 11 Model performance with/without Autoencoder in Machinery Failure Prevention Technology (MFPT)

set and MFPT data set, respectively. It is noted that the model with the Autoencoder can achieve better results in both data sets, which shows the Autoencoder can generate benefit features for the boosting tree model in the vertical FL schema.

The Xgboost model can generate the feature importance based on the tree structure. This paper adopts the gain score as the criteria, where the higher the information gain, the more important the feature. Based on the obtained feature importance, the average score is used to evaluate the effectiveness of the Autoencoder. Table 2 displays the feature importance in active client and passive client with or without the Autoencoder, where **Active S** indicates the average feature importance score from the active client, and vice versa. The average feature importance with the Autoencoder for passive clients is 71.45, slightly higher than that without the Autoencoder (70.86) in the CWRU data set. Meanwhile, the feature importance in MFPT has the same finding (61.81 vs. 61.55). The feature importance indicates the features coverage in the boosting tree, and the coverage is larger, representing a higher possibility for the active client to retrieve the feature.

TABLE 2 Feature importance in Boosting tree

	With autoencoder		Without autoencoder	
Data set	**Active S**	**Passive S**	**Active S**	**Passive S**
CWRU	82.21 ± 1.04	71.45 ± 1.49	82.06 ± 2.04	70.86 ± 1.57
MFPT	75.19 ± 1.21	61.81 ± 0.72	75.11 ± 1.17	61.55 ± 0.87

6 | DISCUSSION

The proposed model allows active client to cooperate with passive clients regarding different sample sizes and feature numbers to train a boosting tree jointly with data security. The experiment result reveals that the proposed model can diagnose the fault more accurately by integrating passive clients' features.

In spite of the above achievements, some limitations also exist in this research due to the low computation power, for instance: (1) the default hyperparameter. This research adopts the recommended hyperparameter settings rather than searching for the optimal one. (2) The sampling number. Each client sample number ranges from 500 to 8000 in the experiment, while the diagnosis results could be improved if the client data sample number is expanded.

7 | CONCLUSION

Data-driven models have been widely implemented in fault diagnosis, while the limited features constrict them to achieve more accurate results. Many previous researchers have focussed on leveraging limited features to diagnose the fault. Nevertheless, no work has yet attempted to break down the data silos for obtaining more numbers of realistic features. This paper proposed a privacy-preserving boosting tree with an Autoencoder to enlarge the features for fault diagnosis with vertical FL.

It is believed that this research can help other industrial scenarios to break down the data silos. In the meantime, future research can include 1) effectively searching for suitable hyperparameters and 2) applying boosting tree to additional FL schemas (such as federated transfer learning).

ACKNOWLEDGEMENTS

This research is partially funded by the Mainland-Hong Kong Joint Funding Scheme (MHX/001/20), Innovation and Technology Commission (ITC), Hong Kong Special Administration Region, National Key R&D Programs of Cooperation on Science and Technology Innovation with Hong Kong, Macao and Taiwan (SQ2020YFE020182), Ministry of Science and Technology (MOST) of the People's Republic of China, Centre for Advances in Reliability and Safety (CAiRS) admitted under AIR@InnoHK Research Cluster, and the Shanghai Rising-Star Plan (Yangfan Program) from the Science and Technology Commission of Shanghai Municipality (22YF1400200).

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest that could be perceived as prejudicing the impartiality of the research reported.

DATA AVAILABILITY STATEMENT

CWRU: https://engineering.case.edu/bearingdatacenter.
MFPT: https://www.mfpt.org/fault-data-sets/.

ORCID

Liqiao Xia https://orcid.org/0000-0001-8878-7490
Pai Zheng https://orcid.org/0000-0002-2329-8634

REFERENCES

1. Jing, T., et al.: Cloud-Edge collaboration framework with deep learning-based for remaining useful life prediction of machinery. In: IEEE Transactions on Industrial Informatics (2021)
2. Gao, Z., Cecati, C., Ding, S.X.: A survey of fault diagnosis and fault-tolerant techniques—Part I: fault diagnosis with model-based and signal-based approaches. In: IEEE Transactions on Industrial Electronics, 62 6, pp. 3757–3767. (2015)
3. Zhang, T., et al.: Intelligent fault diagnosis of machines with small & imbalanced data: a state-of-the-art review and possible extensions. In: ISA Transactions, vol. 119, pp. 152–171. (2022)
4. Voigt, P., Von dem Bussche, A.: The eu general data protection regulation (gdpr). In: A Practical Guide, vol. 10, 1st ed, pp. 10–5555. Springer International Publishing Cham 3152676. (2017)
5. Keith, B., et al.: Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1175–1191. (2017)
6. Zhang, X., et al.: Promoting employee health in smart office: a survey. In: Advanced Engineering Informatics, vol. 51, 101518. (2022)
7. Fan, J., Zheng, P., Li, S.: Visionbased holistic scene understanding towards proactive human–robot collaboration. In: Robotics and Computer- Integrated Manufacturing, vol. 75, 102304. (2022)
8. Keung, K.L., Lee, C.K.M., Ji, P.: Industrial internet of things-driven storage location assignment and order picking in a resource synchronization and sharing-based robotic mobile fulfillment system. In: Advanced Engineering Informatics, vol. 52, 101540. (2022)
9. Li, C., et al.: AR-assisted digital twin-enabled robot collaborative manufacturing system with human-in-theloop. In: Robotics and Computer-Integrated Manufacturing, vol. 76, 102321. (2022)
10. Lok Keung, K., Lee, C.K.M., Ji, P.: Datadriven order correlation pattern and storage location assignment in robotic mobile fulfillment and process automation system. In: Advanced Engineering Informatics, vol. 50, 101369. (2021)
11. Xia, L., et al.: A novel hypergraph convolution network-based approach for predicting the material removal rate in chemical mechanical planarization. J. Intell. Manuf., 1–12 (2021). https://doi.org/10.1007/s10845-021-01784-1
12. Xia, L., Zheng, P., Liu, C.: Predicting the material removal rate in chemical mechanical planarization process: a hypergraph neural network-based approach. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. 85376. American Society of Mechanical Engineers (2021).V002T02A057
13. Lyu, P., et al.: A novel RSG-based intelligent bearing fault diagnosis method for motors in high-noise industrial environment. In: Advanced Engineering Informatics, vol. 52, 101564. (2022)
14. Zhu, F., et al.: Methodology for important sensor screening for fault detection and classification in semiconductor manufacturing. In: IEEE Transactions on Semiconductor Manufacturing, vol. 34 1, pp. 65–73. (2020)
15. Chen, Z., Gryllias, K., Li, W.: Intelligent fault diagnosis for rotary machinery using transferable convolutional neural network. In: IEEE Transactions on Industrial Informatics, vol. 16 1, pp. 339–349. (2019)
16. Cheng, P., et al.: Research on fault diagnosis of wind power generator blade based on SC-SMOTE and kNN. In: Journal of Information Processing Systems, vol. 16 4, pp. 870–881. (2020)
17. Wan, W., et al.: QSCGAN: an un-supervised quick self-attention convolutional GAN for LRE bearing fault diagnosis under limited label-lacked data. In: IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1–16. (2021)
18. Zhang, A., et al.: Limited data rolling bearing fault diagnosis with few-shot learning. In: IEEE Access, vol. 7, pp. 110895–110904. (2019)
19. Feng, L., Zhao, C.: Fault description based attribute transfer for zero-sample industrial fault diagnosis. In: IEEE Transactions on Industrial Informatics, vol. 17 3, pp. 1852–1862. (2020)
20. Wen, L., Gao, L., Li, X.: A new deep transfer learning based on sparse auto-encoder for fault diagnosis. In: IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49 1, pp. 136–144. (2017)
21. Chen, Z., et al.: Domain adversarial transfer network for cross-domain fault diagnosis of rotary machinery. In: IEEE Transactions on Instrumentation and Measurement, vol. 69 11, pp. 8702–8712. (2020)
22. Zhang, K., et al.: Fault diagnosis of planetary gearbox using a novel semi-supervised method of multiple association layers networks.

In: Mechanical Systems and Signal Processing, vol. 131, pp. 243–260. (2019)
23. Zhang, Z., et al.: General normalized sparse filtering: a novel unsupervised learning method for rotating machinery fault diagnosis. In: Mechanical Systems and Signal Processing, vol. 124, pp. 596–612. (2019)
24. Wang, H., et al.: Self-supervised signal representation learning for machinery fault diagnosis under limited annotation data. In: Knowledge-Based Systems, 107978. (2021)
25. Wang, S., et al.: Knowledge reasoning with semantic data for real-time data processing in smart factory. In: Sensors, vol. 18 2, p. 471. (2018)
26. Xu, F., et al.: Ontology-based method for fault diagnosis of loaders. In: Sensors, vol. 18 3, pp. 729. (2018)
27. Pan, D., et al.: Use of collaborative concept mapping in team diagnosis. In: Human Factors and Ergonomics in Manufacturing & Service Industries, vol. 31 5, pp. 469–483. (2021)
28. Montero Jimenez, J.J., et al.: Towards multi-model approaches to predictive maintenance: a systematic literature survey on diagnostics and prognostics. J. Manuf. Syst. 56, 539–557 (2020). https://doi.org/10.1016/j.jmsy.2020.07.008
29. Belhadi, A., et al.: Reinforcement learning multi-agent system for faults diagnosis of mircoservices in industrial settings. In: Computer Communications, vol. 177, pp. 213–219. (2021)
30. Zheng, P., et al.: Towards Self-X cognitive manufacturing network: an industrial knowledge graph-based multiagent reinforcement learning approach. J. Manuf. Syst. 61, 16–26 (2021). https://doi.org/10.1016/j.jmsy.2021.08.002
31. Zheng, P., et al.: A visual reasoning-based approach for mutual-cognitive human-robot collaboration. In: CIRP Annals (2022)
32. Xia, L., et al.: Toward cognitive predictive maintenance: a survey of graph-based approaches. J. Manuf. Syst. 64, 107–120 (2022). https://doi.org/10.1016/j.jmsy.2022.06.002
33. Liu, H., et al.: A knowledge model-based BIM framework for automatic code-compliant quantity take-off. In: Automation in Construction, vol. 133, 104024. (2022)
34. Wang, H., Qi, Y., Wang, J.: Blockchainsecured multi-factory production with collaborative maintenance using Q learning-based optimisation approach. Int. J. Prod. Res., 1–18 (2021). https://doi.org/10.1080/00207543.2021.2002968
35. Yang, Q., et al.: Federated learning. In: Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 13 3, pp. 1–207. (2019)
36. McMahan, B., et al.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
37. Li, T., et al.: Fair resource allocation in federated learning. In: arXiv preprint arXiv:1905.10497 (2019)
38. Zhang, W., et al.: Federated learning for machinery fault diagnosis with dynamic validation and self-supervision. In: Knowledge-Based Systems, vol. 213, 106679. (2021)
39. Zhang, Z., et al.: Adaptive privacy preserving federated learning for fault diagnosis in internet of ships. In: IEEE Internet of Things Journal (2021)
40. Zhang, J., et al.: Diagnosis of interturn short- circuit faults in permanent magnet synchronous motors based on few-shot learning under a federated learning framework. In: IEEE Transactions on Industrial Informatics, vol. 17 12, pp. 8495–8504. (2021)
41. Zhang, W., Li, X.: Federated transfer learning for intelligent fault diagnostics using deep adversarial networks with data privacy. In: IEEE/ASME Transactions on Mechatronics (2021)
42. Zhang, W., Li, X.: Data privacy preserving federated transfer learning in machinery fault diagnostics using prior distributions. In: Structural Health Monitoring, 14759217211029201. (2021)
43. Cheng, K., et al.: Secureboost: a lossless federated learning framework. In: IEEE Intelligent Systems, vol. 36 6, pp. 87–98. (2021)
44. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, pp. 785–794. (2016)
45. Faysal, A., et al.: Noise eliminated ensemble empirical mode decomposition scalogram analysis for rotating machinery fault diagnosis. Sensors, 21(23), 8114 (2021). https://doi.org/10.3390/s21238114

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Xia, L., et al.: Privacy-preserving gradient boosting tree: vertical federated learning for collaborative bearing fault diagnosis. IET Collab. Intell. Manuf. 1–12 (2022). https://doi.org/10.1049/cim2.12057