ITC 4/54
Information Technology and Control
Vol. 54 / No. 4/ 2025
pp. 1206-1226
DOI 10.5755/j01.itc.54.4.41998

Robust Industrial Equipment Fault Detection via Bayesian Federated Learning with Channel Importance

Received 2025/06/24 | Accepted after revision 2025/08/26

HOW TO CITE: Tang, Z., Xu, H., Hu, H., Xu, C., Zhang, W. (2025). Robust Industrial Equipment Fault Detection via Bayesian Federated Learning with Channel Importance. *Information Technology and Control*, 54(4), 1206-1226. https://doi.org/10.5755/j01.itc.54.4.41998

Robust Industrial Equipment Fault Detection via Bayesian Federated Learning with Channel Importance

Zhongyun Tang
School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China; School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou, China; Shangyu Institute of Science and Engineering, Hangzhou Dianzi University, Shaoxing, China; e-mail: tangzy@hdu.edu.cn

Hanyi Xu
School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou, China; e-mail: 1811060531@pop.zjgsu.edu.cn

Haiyang Hu*
School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China; Shangyu Institute of Science and Engineering, Hangzhou Dianzi University, Shaoxing, China; e-mail: huhaiyang@hdu.edu.cn

Chonghuan Xu
School of Business Administration, Zhejiang Gongshang University, Hangzhou, China; e-mail: talentxch@zjgsu.edu.cn

Weiyu Zhang
Faculty of Education and Liberal Arts, INTI International University, Nilai, Malaysia; e-mail: i25030576@student.newinti.edu.my

Corresponding author: huhaiyang@hdu.edu.cn

In the era of industrial automation and smart manufacturing, the reliability of industrial equipment is of paramount importance as equipment failures can trigger severe production disruptions and safety hazards. While Bayesian Learning effectively handles uncertainties and enhances fault detection interpretability,

it faces challenges in data privacy and centralized processing, whereas Federated Learning safeguards data privacy. Bayesian Federated Learning (BFL) integrates their strengths for privacy-preserving probabilistic modeling with superior generalization. This paper innovatively proposes BFL-CI (Bayesian Federated Learning with Channel Importance), presenting a comprehensive framework and detection process that analyzes channel parameter distributions and transmits selected channel distributions as prior distributions for subsequent clients. Experiments on bearing and gearbox datasets simulating multi-sensor scenarios verify BFL-CI's effectiveness in improving fault detection accuracy and robustness. This paper fills the gap in applying BFL to industrial equipment fault detection, offering a cutting-edge privacy-preserving solution for reliable intelligent industrial monitoring.

KEYWORDS: Industrial Fault Detection; Bayesian Federated Learning; Interpretability; Channel Importance

1. Introduction

With the rapid development of industrial automation and smart manufacturing, the performance stability and reliability of industrial equipment have become key factors in production efficiency and safety. A fault in the equipment not only leads to production interruptions but can also trigger significant safety incidents, resulting in substantial economic losses [1]. Therefore, effective fault detection methods are crucial for ensuring the continuity of industrial production and enhancing the efficiency of equipment maintenance [25].

Traditional fault detection methods rely on expert experience and periodic inspections, which are often inefficient and struggle to adapt to the complex and changing industrial environments. With the development of big data and machine learning technologies, data-driven fault detection methods have gradually become a research hotspot [2,28]. In addition to most deep learning approaches, Bayesian methods offer a new perspective for fault detection by leveraging their advantages in handling uncertainty and prior knowledge [18].

Bayesian Learning is a machine learning method based on Bayes Theorem, which lies in the updating of posterior probability distributions through prior probability distributions and likelihood functions. It can provide an estimate of the uncertainty in model parameters and allows knowledge to be incorporated into the model in the form of priors. Bayesian Neural Networks (BNNs) are a typical instance of Bayesian Learning. Unlike traditional neural networks, BNNs treat network parameters as random variables rather than fixed values. It makes the model output confidence of its predictions. This characteristic makes BNNs offer a new perspective on model interpretability [24,13,3].

In the application of industrial equipment fault detection, Bayesian Learning can establish fault detection models by learning the data features of equipment under normal operating conditions and fault states. Given that data in industrial environments often contain noise and uncertainties, the uncertainty estimation capability of Bayesian Learning can help us better understand the model's prediction results, thereby leading to more accurate maintenance decisions [33,34].

Traditional Deep Learning or Bayesian Learning typically require data to be centralized at a single location for processing. In industrial environment, the data generated by equipment often contain sensitive information, and directly uploading this data to the cloud for centralized processing may pose risks of data leakage [20]. Due to the wide variety of industrial equipment, each operating in different environments and conditions, a single fault detection model is difficult to adapt to such a diverse range of application scenarios [11]. Bayesian Learning generally require substantial computational resources for training and inference, especially when dealing with large datasets and complex models.

In recent years, Federated Learning has garnered attention as an emerging distributed machine learning technology due to its unique capability to collaboratively train high-quality machine learning models without sharing raw data. Federated Learning allows different entities as clients to jointly train models without centralizing their data, thereby protecting data privacy [30,16]. Federated Learning has also been applied in the industry. Under the premise

of not affecting data privacy, it optimizes production processes and improves product quality and efficiency through cross-organizational collaborative training of machine learning models. It also supports personalized manufacturing and services, enabling various small and medium-sized factories to share data and benefit from industrial big data without disclosing sensitive information [17]. However, traditional Federated Learning methods still have limitations in handling uncertainties, especially in application scenarios that require uncertainty estimation of model parameters. There is also a theoretical risk of privacy leakage if the model parameters are intercepted during transmission [32].

Bayesian Federated Learning (BFL), as a technology that combines Federated Learning with Bayesian statistical methods, can effectively address the challenges. Bayesian Federated Learning not only inherits the advantages of Federated Learning in data privacy protection, but also achieves probabilistic description of model parameters through Bayesian framework. This method allows for the updating of model parameters in the form of probability distributions, thereby enhancing the model's generalization ability and robustness. As a result, it performs excellently in handling limited data and uncertainties [9, 4].

This paper proposes an industrial equipment fault detection method based on Bayesian Federated Learning. The contributions are as follows:

1 Currently, there is limited research applying Bayesian Federated Learning to the fault detection of industrial equipment. This paper innovatively proposes a framework and detection process for industrial equipment fault detection using Bayesian Federated Learning.
2 Leveraging the interpretability of Bayesian methods and the characteristics of equipment fault data, this paper proposes a Bayesian Federated Learning approach for equipment fault detection—BFL with interpretability analysis for Channel Importance. This method analyzes the important channels in the input layer of the model through the interpretability of the Bayesian model, that is, the corresponding important sensors. During the federated learning process, this method only transmits distributions of selected important channels as prior distributions for the next client. It not only implements a new Bayesian Federated Learning method for industrial equipment fault detection but also significantly reduces the information leakage of model parameters during the transmission.
3 Experiments using real-world bearing data sets demonstrate the effectiveness of the proposed method. The performance of the method in this paper has been proved by comparison with other methods. It is suitable for scenarios involving data collection from multiple sensors in industrial equipment.

Section 1 discusses the importance of industrial equipment fault detection. There are many issues in this area, such as insufficient industrial data and data privacy protection. It emphasizes the relevance of using Bayesian Federated Learning in this domain to improve fault detection capabilities. Section 2 introduces the related work of industrial applications using Bayesian Learning and the applications of Bayesian Federated Learning. Section 3 introduces the overall framework and detection process of Bayesian Federated Learning for industrial equipment fault detection. Section 4 introduces how Bayesian Federated Learning, combined with interpretability, identifies important channels in industrial equipment fault detection and then modifies the variance of the prior distributions for corresponding channels on other clients to achieve overall learning and detection. Section 5 introduces the dataset, experimental methods, and experimental results of real industrial equipment experiments. This section analyzes the experimental results compared to other methods. Section 6 provides a summary and discussion of the research.

2. Literature Review

There are several studies on the application of Bayesian Learning to industry-relevant research. Dynamic models based on Bayesian Artificial Neural Networks (BANNs) have been proposed for accurately predicting the thermodynamic efficiency and indoor temperature of direct expansion air conditioning systems, providing a basis for improving system control strategies and enhancing energy efficiency [22]. The use of BNNs has been proposed to quantify uncertainties in data-driven materials modeling, addressing key shortcomings in current machine learning techniques' handling of uncertainties and providing more reliable and precise prediction tools for appli-

cations in material science and other engineering fields [23]. A novel Bayesian neural network-based approach has been developed for the automatic detection of anomalies in retinal optical coherence tomography (OCT) images, improving upon existing anomaly detection methods in sensitivity and specificity [21]. A new Pior-Assistant Random Bayesian Neural Network (PA-RBNN) method has been proposed, combining BNNs with effective priors to address the knowledge uncertainty in online quality prediction in process industries, enhancing the robustness and real-time performance of prediction models [31]. Methods utilizing BNNs for predicting the permeability of industrial drilling have demonstrated the potential of BNNs in dealing with complex industrial data and enhancing prediction accuracy [7]. A novel method combining physical constraints with BNNs for predicting the remaining useful life of batteries has been proposed, aiming to improve prediction accuracy and address challenges in early prediction stages [10].

There are several studies on Bayesian Federated Learning and its applications. A novel Federated Learning method is proposed, which uses knowledge distillation techniques on each client to transform predictive posterior distributions (PPDs) into a single deep neural network, effectively addressing model uncertainty issues. This method reduces communication costs, enhances prediction efficiency and accuracy, and is applicable to various scenarios such as classification, active learning, and anomaly detection, holding significant value for advancing the practical application of Federated Learning. However, the method increases computational burden on clients for posterior distribution inference and PPD distillation, and it is relatively sensitive to data heterogeneity [5]. A novel Federated Learning framework is proposed, which uses online Laplace approximation to approximate posteriors on both the client and server sides. This method reduces aggregation errors and mitigates local forgetting by employing a multivariate Gaussian product mechanism on the server side and a prior loss on the client side. Theoretical and experimental results demonstrate that this approach outperforms existing methods in terms of accuracy and convergence speed. However, the method has certain limitations, including increased computational complexity due to the Laplace approximation, higher communication costs during iterative updates, and sensitivity to data heterogeneity, which can affect performance when the data distribution varies significantly across clients [19]. A framework for Bayesian Federated Learning and unlearning in decentralized networks is proposed. The method uses variational inference to collaboratively infer a distribution that approximates the global posterior, allowing for efficient unlearning mechanisms. The approach is based on exponential-family parametrization and local gossip-driven communication, making it suitable for decentralized architectures. Experimental results show that the proposed protocol outperforms existing methods in terms of accuracy and efficiency. However, the method has certain limitations, including increased computational complexity due to the variational inference process, higher communication overhead in large-scale decentralized networks, and sensitivity to the initial conditions and network topology, which can affect convergence and performance [14]. A personalized federated learning method named pFedBayes introduces variational Bayesian inference to address the issue of model overfitting under the non-i.i.d. and limited data conditions. The method introduces weight uncertainty on both clients and the server to balance local data construction errors and the KL divergence with the global distribution, achieving personalized model updates. Theoretical analysis shows that this method reaches an optimal convergence rate in average generalization error. Experimental results indicate that pFedBayes outperforms other state-of-the-art algorithms by 1.25%, 0.42%, and 11.71%, respectively. However, the method has certain limitations in terms of computational complexity, communication cost, and hyperparameter tuning, particularly imposing a heavy computational burden on resource-limited devices and leading to high total communication costs during multiple rounds of iteration [27]. A new perspective on federated learning as a distributed inference problem is proposed, which uses variational inference to find a global variational posterior that better approximates the true posterior. The method introduces a federated learning method based on expectation propagation (FedEP), which iteratively refines the approximation of the global posterior through probabilistic message-passing between the central server and clients. Experimental results show that FedEP outperforms strong baseline methods on standard federated learning benchmarks, achieving

faster convergence and higher accuracy. However, the method has certain limitations in terms of computational complexity, communication cost, and sensitivity to data heterogeneity, particularly in scenarios with large-scale models and numerous clients, where it may require more computational resources and higher communication costs [12].

This paper proposes a Bayesian Federated Learning approach for detecting faults in industrial equipment. There is no research about Bayesian Federated Learning applied in industry. Furthermore, this paper combines the interpretability in Bayesian Learning with the distributed computing in Federated Learning and innovatively proposes an analysis method for important channels. Meanwhile, through experiments and comparisons with other methods on real experimental datasets, the advantages of the method in this paper have been proved.

3. Overall Framework

3.1. Framework of Bayesian Federated Learning

Bayesian Federated Learning is an approach that combines Bayesian Learning and Federated Learning, aiming to address the issues of the singularity and uncertainty of model parameter estimation in traditional Federated Learning. As shown in Figure 1, the BFL architecture consists of a central server and multiple clients.

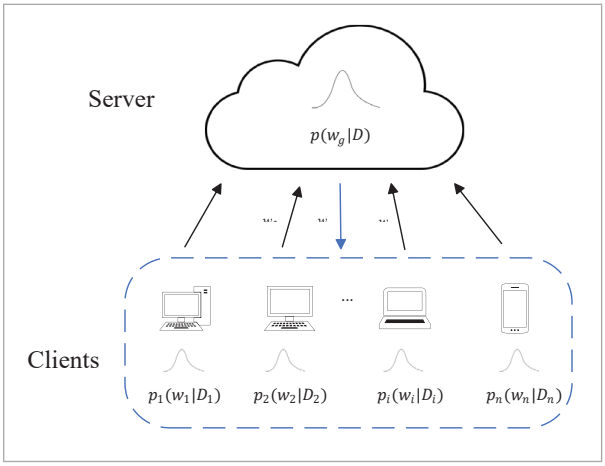
Each client possesses its own private dataset and independently trains a Bayesian model, generating a posterior distribution.

$$p\left(\boldsymbol{w}_i|D_i\right)=\frac{p\left(D_i|\boldsymbol{w}_i\right)p\left(\boldsymbol{w}_i\right)}{p\left(D_i\right)}, \tag{1}$$

where w_i represents the model parameters of client i. These posterior distributions remain private between the clients and only summary information of the model parameters is sent to the server. The global posterior distribution at the server can be expressed as:

$$p\left(\boldsymbol{w}_g|D\right)=\frac{p\left(D|\boldsymbol{w}_g\right)p\left(\boldsymbol{w}_g\right)}{p\left(D\right)}, \tag{2}$$

Figure 1
Bayesian Federated Learning.



where w_g represents the model parameters of the server, while D denotes the entire dataset from all clients in the Federated Learning process. The server collects information from all clients and integrates it into a global posterior distribution $p(w_g \mid D)$. This process not only ensures data privacy but also provides a comprehensive understanding of model uncertainty, enhancing the robustness and adaptability of the model.

Due to the large and complex computation involved in calculating the posterior distribution, variational inference is commonly used. The true posterior distribution $P(w \mid D)$ is approximated by $q(w \mid \theta)$. During training, the parameter θ is continuously optimized to minimize the Kullback-Leibler (KL) divergence between this distribution and the true Bayesian posterior. The optimization process is shown in the following formula:

$$\begin{aligned}
\theta^* &= \underset{\boldsymbol{w}_i}{argmin}\, KL\left[q(\boldsymbol{w}_i \mid \theta)\|p(\boldsymbol{w}_i \mid D_i)\right] \\
&= \underset{\boldsymbol{w}_i}{argmin}\int q\left(\boldsymbol{w}_i|\theta\right)\log\frac{q\left(\boldsymbol{w}_i|\theta\right)}{p\left(\boldsymbol{w}_i\right)p\left(D_i|\boldsymbol{w}_i\right)}d\boldsymbol{\omega}_i \\
&= \underset{\boldsymbol{w}_i}{argmin}\int q\left(\boldsymbol{w}_i|\theta\right)\left(\log\frac{q\left(\boldsymbol{w}_i|\theta\right)}{p\left(\boldsymbol{w}_i\right)}-\log p\left(D_i|\boldsymbol{w}_i\right)\right)d\boldsymbol{w}_i \\
&= \underset{\boldsymbol{w}_i}{argmin}\, KL\left[q(\boldsymbol{w}_i \mid \theta)\|p\left(\boldsymbol{w}_i\right)\right] \\
&-E_{q(\boldsymbol{w}_i|\theta)}\left[\log p(D_i \mid \boldsymbol{w}_i)\right]
\end{aligned} \tag{3}$$

In standard Federated Learning, the goal is to minimize the global loss function through collaboration between the server and the clients. The loss function, ELBO (Evidence Lower Bound), is shown in the following formula:

$$ELBO = \min F(\boldsymbol{w}) = \sum_{i=1}^{1} \frac{1}{|D|} \sum_{a \in D_i} l(\boldsymbol{x}_a, y_a, \boldsymbol{w}). \quad (4)$$

In Bayesian Federated Learning, this objective function is modified to:

$$\begin{aligned} &\underset{\boldsymbol{w}_i}{argmin} \sum_{i=1}^{I} l(\boldsymbol{w}_i; D_i) \\ &= \underset{\boldsymbol{w}_i}{argmin} \sum_{i=1}^{I} KL\left[q(\boldsymbol{w}_i \mid \theta) \| p(\boldsymbol{w}_i)\right] \\ &-E_{q(\boldsymbol{w}_i|\theta)}\left[\log p(D_i|\boldsymbol{w}_i)\right] \\ &= \underset{\boldsymbol{w}_i}{argmin} \sum_{i=1}^{I} \int q(\boldsymbol{w}_i \mid \theta) \log \frac{q(\boldsymbol{w}_i \mid \theta)}{p(\boldsymbol{w}_i)} d\boldsymbol{w}_i \\ &-E_{q(\boldsymbol{w}_i|\theta)}\left[\log p(D_i \mid \boldsymbol{w}_i)\right] \end{aligned}, \quad (5)$$

where $l(w_i; D_i)$ is the loss function of the client i.

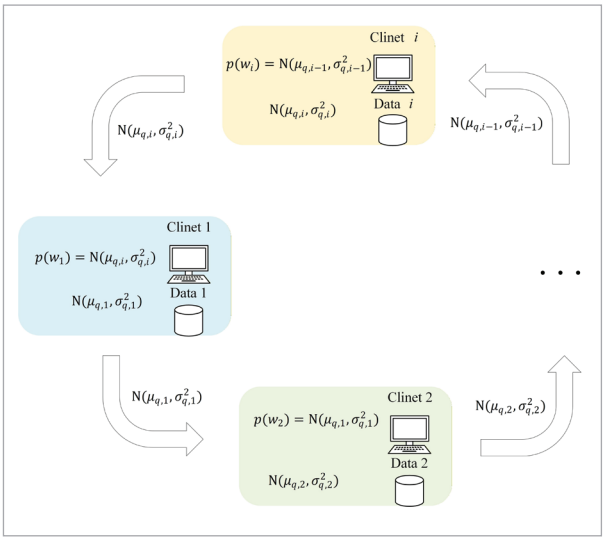
Decentralized Federated Learning is another Federated Learning framework, where model training is performed through direct parameter exchange between clients, reducing reliance on a central server and enhancing system robustness and privacy protection [29]. Decentralized Bayesian Federated Learning (DBFL) combines Bayesian Learning and transforms client models into Bayesian forms, as shown in Figure 2.

For the clients participating in DBFL training, they use the posterior distribution of the previous client as the prior distribution for their current local model to perform Federated Learning. The posterior distribution for these clients is expressed as follows:

$$\begin{aligned} &p(\boldsymbol{w}_i|D_i) \\ &= \frac{p(D_i|\boldsymbol{w}_i) \cdot \frac{p(D_{i-1}|\boldsymbol{w}_{i-1}) \cdot p(\boldsymbol{w}_{i-1})}{p(D_{i-1})}}{p(D_i)} \\ &= \frac{p(D_i \mid \boldsymbol{w}_i) \cdot p(D_{i-1} \mid \boldsymbol{w}_{i-1}) \cdot p(\boldsymbol{w}_{i-1})}{p(D_{i-1}) \cdot p(D_i)} \end{aligned}. \quad (6)$$

Figure 2

Decentralized Bayesian Federated Learning.



Therefore, each client performs local training independently, without the need for global optimization. As a result, the process of optimizing θ for each client becomes:

$$\begin{aligned} &\theta^* = \underset{\boldsymbol{w}_i}{argmin} KL\left[q(\boldsymbol{w}_i \mid \theta) \| P(\boldsymbol{w}_i \mid D_i)\right] \\ &= \underset{\boldsymbol{w}_i}{argmin} \int q(\boldsymbol{w}_i \mid \theta) \\ &\left(\log \frac{q(\boldsymbol{w}_i \mid \theta)}{p(\boldsymbol{w}_i)} - \log p(D \mid \boldsymbol{w}_i)\right) d\boldsymbol{w}_i \\ &= \underset{\boldsymbol{w}_i}{argmin} \int q(\boldsymbol{w}_i|\theta) \\ &\left(\log \frac{q(\boldsymbol{w}_i|\theta)\, p(D_{i-1})}{p(D_{i-1}|\boldsymbol{w}_{i-1})\, p(\boldsymbol{w}_{i-1})} - \log p(D|\boldsymbol{w}_i)\right) d\boldsymbol{w}_i \\ &= \underset{\boldsymbol{w}_i}{argmin} KL\left[q(\boldsymbol{w}_i \mid \theta) \| \frac{p(D_{i-1} \mid \boldsymbol{w}_{i-1}) p(\boldsymbol{w}_{i-1})}{p(D_{i-1})}\right] \\ &-E_{q(\boldsymbol{w}_i|\theta)}\left[\log P(D \mid \boldsymbol{w}_i)\right] \end{aligned} \quad (7)$$

Since data and model parameters are only exchanged between clients without passing through a central server node, the security and privacy of the data are further enhanced. In a decentralized network, each client updates its model's posterior distribution based on information received from other clients, ultimately achieving collaborative optimization of

the global model. However, due to the asynchronous mechanism, the overall training time may be longer. In this paper, the DBFL approach is employed to transmit model channel parameters.

3.2. BFL-CI Framework and Process for Industrial Equipment Fault Detection

This paper applies Bayesian Federated Learning to the domain of industrial equipment fault detection. Devices from factories or production lines located in different geographical areas serve as clients within the federated learning network. Instead of sharing raw data, these devices exchange only model parameters or gradient information, thus collaboratively training a more robust and accurate fault detection model while preserving data privacy. Through a decentralized federated learning approach, each client utilizes locally collected sensor data—such as vibration, temperature, and pressure signals—to train a local model, which is then passed on to the next client. After several rounds of iterative optimization, the model can better generalize across various industrial settings, enhancing the accuracy of fault predictions, minimizing unexpected downtime, and optimizing maintenance planning. Additionally, this paper introduces the method of Bayesian Federated Learning with Channel Importance that reduces the amount of data parameter probability distribution shared among clients, thereby enhancing privacy protection. The specific framework is illustrated in Figure 3.

First, among all the clients involved in Bayesian Federated Learning, select a starting client, preferably the one possessing the largest dataset that encompasses all known fault types. In case this client is deficient in data for a particular fault type, it should engage in small-scale federated Bayesian training in collaboration with another client that holds the most comprehensive data for that specific fault. During this process, the model-passing technique is utilized.

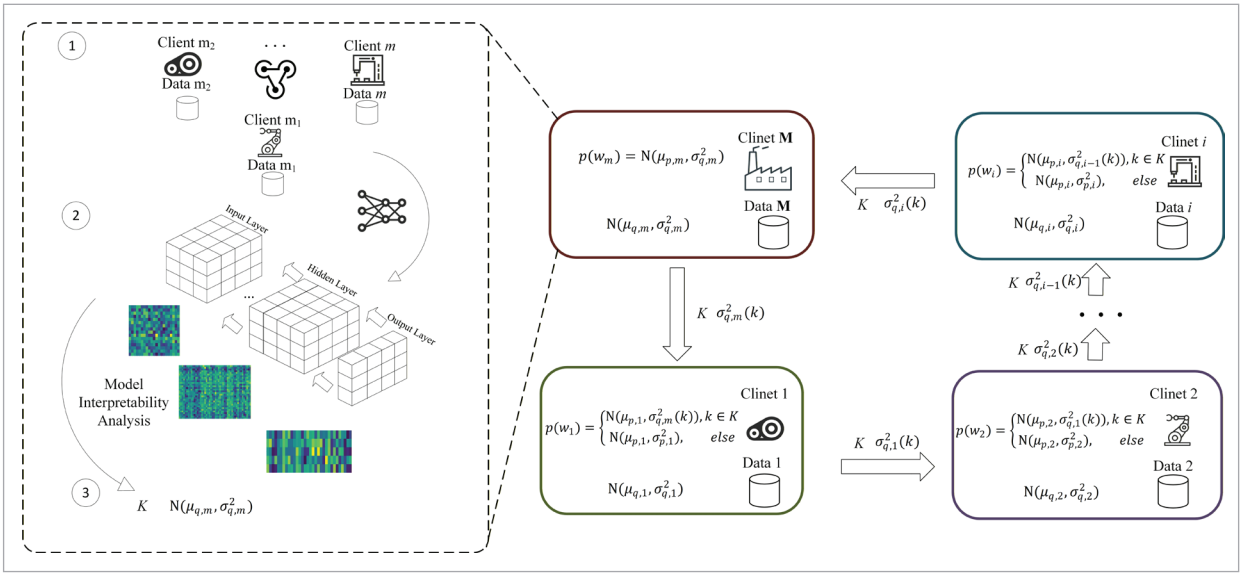
Second, for the model obtained from the preceding step, apply Bayesian principles to carry out an interpretability analysis. By scrutinizing the distribution of model parameters, trace backward from the output layer to identify the crucial channels in the input layer. Detailed explanations can be found in Section 4.

Third, transmit the parameter distributions of the pinpointed input layer channels to other clients to serve as their prior distributions. Subsequently, these clients initiate their training.

Finally, after a predetermined number of training rounds, each client will successively pass the parameter distributions of the trained channels to the next client in a cyclic manner until all clients have completed the training of their models to satisfaction.

Figure 3

BFL-CL framework with Channel Importance for industrial equipment fault detection.



4. Proposed Method

4.1. The Interpretability of Bayesian Learning

Bayesian Learning handles uncertainty by treating model parameters as probability distributions rather than fixed values. The probabilistic nature of Bayesian models makes their internal workings more understandable [8]. It clearly shows how input data and prior information jointly influence the final decision. This means that when detecting potential faults in industrial equipment, it can provide a confidence interval rather than just a definitive result. For maintenance engineers, understanding the degree of uncertainty behind the detection results is very important, because it helps them assess risks and make wiser decisions. By selecting appropriate prior distributions, Bayesian Learning also integrates the experience of domain experts or data from past cases. This not only helps improve the accuracy of the model but also makes the model closer to real-world situations. In the field of industrial equipment fault detection, Bayesian Learning greatly enhances the interpretability of the model through methods such as quantifying uncertainty, integrating professional knowledge, improving predictive accuracy, and increasing model transparency, providing robust support for achieving efficient and reliable industrial maintenance.

In this paper's experiments, a Bayesian Convolutional Neural Network (BCNN) is used as the network architecture, serving as an example to illustrate interpretability.

1 Estimation of Prediction Uncertainty

In industrial equipment fault detection, accurately assessing prediction uncertainty is critical for preventive maintenance and decision support. BCNNs can quantify the uncertainty of model predictions by introducing prior and posterior distributions of parameters. For a given input data point X^* (collected from sensors, representing the operating parameters of the equipment), we can sample multiple weight vectors w from the posterior distribution $P(w \mid D)$ and compute the corresponding predicted fault probabilities $P(Y^* \mid X^*,w)$, where Y^* represents whether the equipment has faults or the specific fault classification.

By computing the mean and standard deviation of these prediction probabilities, we can obtain the expected value of the prediction result along with its associated uncertainty. The calculation formula for the expected value is as follows:

$$E\left[Y^*|X^*\right] = \int Y^* P\left(Y^*|X^*,w\right) P\left(w|D\right) dw. \tag{8}$$

This expected value is calculated by sampling multiple distinct weight vectors w from the model's posterior distribution and averaging the predicted fault probabilities derived from all these samples. It represents the expected probability of equipment faults when accounting for parameter uncertainty. Furthermore, instead of directly determining outcomes solely based on training labels, one can analyze the probability distribution of results from all sampled models and output a measure of uncertainty for unseen data after setting a threshold.

The formula for calculating the standard deviation is as follows:

$$Std\left[Y^*|X^*\right] = \sqrt{\int \left(Y^* - E\left[Y^*|X^*\right]\right)^2 P\left(Y^*|X^*,w\right) P\left(w|D\right) dw}. \tag{9}$$

This standard deviation reflects the uncertainty in the predicted fault probability. A larger standard deviation indicates greater uncertainty in the prediction outcome, which corresponds to a wider confidence interval and suggests significant variability in the model's predictions of equipment faults. This helps engineers determine when to conduct more rigorous equipment inspections or schedule early maintenance. Conversely, a smaller standard deviation leads to a narrower confidence interval, indicating that the model's predictions are more precise and reliable.

2 The impact of various sensors and their related features

In BCNN, c denotes the number of sensors (or input channels), i represents the number of output channels for a given layer, and L is the size of the convolutional kernel, the weight matrix for a one-dimensional convolutional layer in the model is of dimensions (i, c, L).Each weight $\omega_{f,c,l}$ (where l indexes

the position within the convolutional kernel) signifies the weight associated with the feature extracted from the l-th position of input channel c when calculating a particular feature value at a specific position in the current output channel f. This weight determines the degree to which the input feature influences the corresponding output feature.

As discussed earlier, once the posterior probability distribution of the weights is computed, each weight forms a probability distribution—typically a Gaussian distribution, though it may also take a more complex form. In such cases, one can output the mean matrix and variance (or covariance/standard deviation) matrix of these weights. Each weight corresponds to a mean value, resulting in a mean matrix with the same dimensions, which reflects the model's estimation of the "optimal" or "most probable" value for each weight.

Variance (or covariance/standard deviation) is used to quantify uncertainty in the weight probability distributions. If the weights are independently and identically distributed (i.i.d.), the variance (or covariance/standard deviation) matrix is likely diagonal, where diagonal elements represent the variance of each weight. However, when weights are correlated, a non-diagonal covariance matrix is used: off-diagonal elements (non-zero values) indicate the correlation degree between different weights. This covariance matrix shares the same shape as the weight matrix and encapsulates relationships between all possible weight pairs in the model.

After the model training is completed and the posterior distribution $P(w_{f,c,l}|D)$ for the weight $w_{f,c,l}$ has been obtained, the mean of this weight is equal to the mathematical expectation of this posterior distribution. The formula for calculating the mean is as follows:

$$\mu_{f,c,l} = E\left[w_{f,c,l}\,|D\right] = \int w_{f,c,l} P\left(w_{f,c,l}|D\right) dw_{f,c,l}. \tag{10}$$

The expected value of the weight reflects the overall importance of the corresponding sensor and its local features. A high positive expected value indicates that the feature significantly increases the probability of failure. Conversely, a small or negative expected value suggests that the feature either reduces the likelihood of faults or has little impact on the results.

The formula for the weight variance is as follows:

$$Var[w_{f,c,l}\,|\,D] = E\left[(w_{f,c,l} - E[w_{f,c,l}\,|\,D])^2\,|\,D\right] \tag{11}$$

$$Var[w_{f,c,l}\,|\,D] = \int (w_{f,c,l} - \mu_{f,c,l})^2 P\left(w_{f,c,l}|D\right) dw_{f,c,l}. \tag{12}$$

The variance of a weight quantifies the uncertainty in its contribution to the outcome. A larger variance indicates greater uncertainty in evaluating the importance of the corresponding feature. As the non-negative square root of the variance, the standard deviation also conveys this uncertainty but is expressed in the same units as the weights, making it more intuitively interpretable.

4.2. Layer-by-Layer Analysis of Channel Importance

In a neural network, each layer's weight matrix has its corresponding posterior distribution. For multi-layer networks, the weights of each layer reflect how the neurons in that layer influence the neurons in the next layer, rather than directly indicating their impact on the final output. When a BCNN contains multiple convolutional layers, to analyze the impact of various features on the outcome, one must consider the posterior distributions of weights across all layers and account for the nonlinear transformations that features undergo through different levels.

For each convolutional layer, the posterior mean and variance of its weight matrix can be calculated separately. These reflect the importance and uncertainty of different channels or feature maps within that layer. In a multi-layer structure, the output of one layer serves as the input to the next layer. The influence of initial sensor signals may be amplified, diminished, or even altered after passing through multiple convolutions, activation functions, and other operations. By analyzing the relative magnitude and trends of weights across layers, it is possible to analyze which features gradually become more important at different levels or at which layer their influence begins to diminish.

We can progressively analyze the role of sensor signals in different layers and how they cumulatively

impact the final equipment fault detection results through multiple layers of nonlinear transformations.

Firstly, identify the parameter with the largest weight mean in the output fully connected layer. For each output channel $n(n\in[1,N])$ in the fully connected layer, its output can be expressed as:

$$Z_n = b_n + \sum_{g=1}^{F_j}\sum_{o=1}^{O_j} W_{n,g,o}^{(fc)} \cdot Y_{g,o}^{(j)}, \tag{13}$$

where:

Z_n is the output value of output channel n in the fully connected layer.

b_n is the bias for output channel n.

$W_{n,g,o}^{(fc)}$ is the weight at row g and column o in the weight matrix of output channel n.

$Y_{g,o}^{(j)}$ is the value at position o in the feature map produced by convolution kernel g in the j-th convolutional layer.

the shape of the weight matrix for that layer is $(N, F_j \times O_j)$.

for output channel n, the weight with the maximum mean $w_{\max,n}$ is

$$w_{\max,n} = \max\left\{W_{n,g,o}^{(fc)}\right\} \tag{14}$$

the position of the weight with the maximum mean $w_{\max,n}$ is (g^*,o^*),

where the output for each convolution kernel $g(g\in[1,F_j])$ at each position o in the j-th convolutional layer can be calculated using the following formula(Omit the bias term.):

$$Y_{g,o}^{(j)} = \sum_{f=1}^{F_j-1}\sum_{l=0}^{L_j-1} W_{g,f,l}^{(j)} \cdot Y_{f,o\cdot S_j+l-P_j}^{(j-1)}, \tag{15}$$

where:

$Y_{g,o}^{(j)}$ is the output value of convolution kernel g at position o.

$W_{g,f,l}^{(j)}$ is the l-th weight of the f-th input channel for convolution kernel g.

$Y_{f,o\cdot S_j+l-P_j}^{(j-1)}$ is the value from the (j-1)-th convolutional layer's output $Y^{(j-1)}$ for convolution kernel f at position $(o\cdot S_j+l-P_j)$, S_i is the stride, P_j is the padding. The number of input channels in the current convolutional layer equals the number of output channels from the previous convolutional layer, which is the number of convolution kernels in that layer, denoted here as f.

the shape of the weight matrix for that layer is (F_j, F_{j-1}, L_j). Since the position of the weight with the maximum mean found in the fully connected layer is at g^*, this layer needs to find the weight with the maximum mean at g^* in the output layer (at the convolution kernel g^*).

$$w_{\max,g^*} = \max\left\{W_{g^*,f,l}^{(j)}\right\}. \tag{16}$$

The position of the weight with the maximum mean weight $w_{\max,g}$ is (f^*,l^*).

And so on, until the first convolutional layer, which is the input layer. The output for each convolution kernel f at each position o in the first convolutional layer can be calculated using the following formula.

$$Y_{f,o}^{(1)} = \sum_{c=1}^{C}\sum_{l=0}^{L_1-1} W_{f,c,l}^{(1)} \cdot X_{c,o\cdot S_1+l-P_1}, \tag{17}$$

where:

$Y_{f,o}^{(1)}$ is the output value of convolution kernel f at position o.

$W_{f,c,l}^{(1)}$ is the l-th weight of the c-th input channel for convolution kernel f.

$X_{c,o\cdot S_1+l-P_1}$ is the value of the input signal X at channel c and spatial position $(o\cdot S_1+l-P_1)$, where c denotes the channel index, o is the output position, S_1 is the stride, l is the kernel index, and P_1 is the padding.

the shape of the weight matrix for that layer is $(F1,C,L)$, The position of the weight with the maximum mean found in the subsequent layer is at channel f^*. Then, the weight with the maximum mean is found at output channel f^* in this layer.

$$w_{\max,f^*} = \max\left\{W_{f^*,c,l}^{(1)}\right\}. \tag{18}$$

Thus, the corresponding input channel c^* is identified.

In addition, each time the position of the weight with the maximum mean is identified, the variance at that position must also be checked. If the variance is large, it indicates a high level of uncertainty. If it exceeds the corresponding threshold, the maximum value should be discarded, and the next maximum value should be selected, and so on. The overall algorithm is as follows:

Algorithm 1 Layer-by-layer analysis of the Bayesian model for important channels

Input: The mean of the weight distribution in the output layer $W_{n,g,o}^{(fc)}$, the set of output channels N in the input layer $W_{g^*,f,l}^{(j)}$ the mean of the weight distribution of each layer ,the set J of all layers in the model (excluding the output layer).
output: The set of important channels C^*, the corresponding variance $\sigma^2_{f^*,c^*,l^*}$
for n in $[1, N]$
  $\left(g_n^*, o_n^*\right)$=max_index $\left\{W_{n,g,o}^{(fc)}\right\}$
  if $\sigma^2_{g^*,f^*,l^*} < 1$
    for j in $[J,2]$
      $\left(f^*, l^*\right)$=max_index $\left\{W_{g^*,f,l}^{(j)}\right\}$
      if $\sigma^2_{g^*,f^*,l^*} < 1$
        $g^* \leftarrow f^*$
      else $\left\{W_{g^*,f,l}^{(j)}\right\}$ remove w_{g^*,f^*,l^*}
      Break
    else $\left\{W_{n,g,o}^{(fc)}\right\}$ remove w_{g^*,o^*}
    Break
  $\left(c^*, k^*\right)$=max_index $\left\{W_{f^*,c,l}^{(1)}\right\}$
Output $(\boldsymbol{C}^*, \boldsymbol{\sigma}^2_{f^*,c^*,l^*})$

4.3. Channel Distribution Transfer for BFL-based Industrial Equipment Fault Detection

After identifying the important channels in the input layer, this paper transmits the posterior distributions of these channels from the model trained by client 1 to the corresponding channels of other clients as their prior distributions. This paper uses a Gaussian distribution and only transmits the variance. This approach effectively reduces the variance of the corresponding prior distributions based on the results from Client 1, rather than using the default standard Gaussian distribution, thereby enhancing the importance of the corresponding sensors.

For each client i (e.g., different production lines or equipment), we assume that the variational posterior distribution $q(w_i \mid \theta)$ of its model parameters follows a one-dimensional Gaussian distribution $N\left(\mu_{q,i}, \sigma^2_{q,i}\right)$, and its prior distribution $p(\omega_i)$ also follows a Gaussian distribution $N\left(\mu_{p,i}, \sigma^2_{p,i}\right)$.

For the set of parameters K corresponding to the important channels identified through analysis:

When $k \in K$, the prior distribution variance for these parameters is set to the variational posterior variance from the previous client's important channels: $\sigma^2_{p,i}(k) = \sigma^2_{q,i-1}(k)$. This adjustment reduces the variance.

For parameters where $k \notin K$, the prior distribution remains the standard Gaussian distribution with: $\sigma^2_{p,i}(k) = 1$

This approach ensures that the prior distributions for important parameters are informed by previous clients' results, while maintaining a standard prior for other parameters.

For each client i, the KL divergence is given by:

$$\begin{aligned} &KL\left[q(\boldsymbol{w}_i \mid \theta) \| \, p(\boldsymbol{w}_i \mid D_i)\right] \\ &= KL\left[q(\boldsymbol{w}_i \mid \theta) \| \, p\left(\boldsymbol{w}_i\right)\right] \\ &- E_{q(\boldsymbol{w}_i \mid \theta)}\left[\log p\left(D_i \mid \boldsymbol{w}_i\right)\right] \\ &= KL\left[q(\boldsymbol{w}_i \mid \theta) \| \, q(\boldsymbol{w}_{i-1} \mid \theta)\right] \\ &- E_{q(\boldsymbol{w}_i \mid \theta)}\left[\log p\left(D_i \mid \boldsymbol{w}_i\right)\right] \end{aligned} \tag{19}$$

The data likelihood term $E_{q(\omega_i \mid \theta)}\left[\log p(D_i \mid \boldsymbol{w}_i)\right]$ is unaffected by the prior distribution and can be solved and trained using methods such as reparameterization. When the prior distribution is a standard Gaussian distribution (the mean is 0), the KL divergence term simplifies to:

$$\begin{aligned} &KL\left[q(\boldsymbol{w}_i \mid \theta) \| \, p\left(\boldsymbol{w}_i\right)\right] \\ &= \frac{1}{2}\left(\frac{\sigma^2_{q,i} + \mu^2_{q,i}}{\sigma^2_{p,i}} - 1 + \log\left(\frac{\sigma^2_{p,i}}{\sigma^2_{q,i}}\right)\right). \end{aligned} \tag{20}$$

When the variance in the prior distribution for some important channels is replaced, then:

$$\begin{aligned} &KL\left[q(\boldsymbol{w}_i \mid \theta) \| \, p\left(\boldsymbol{w}_i\right)\right] \\ &\begin{cases} = \frac{1}{2}\left(\frac{\sigma^2_{q,i} + \mu^2_{q,i}}{\sigma^2_{p,i-1}} - 1 + \log\left(\frac{\sigma^2_{p,i-1}}{\sigma^2_{q,i}}\right)\right), \text{parameters } k \in K \\ = \frac{1}{2}\left(\sigma^2_{q,i} + \mu^2_{q,i} - 1 + \log\left(\frac{1}{\sigma^2_{q,i}}\right)\right), \text{parameters } k \notin K \end{cases} \end{aligned} \tag{21}$$

For important channels parameters, range the above equation and remove the constant terms:

$$\frac{\mu_{q,i}^2}{\sigma_{p,i-1}^2}+\frac{\sigma_{q,i}^2}{\sigma_{p,i-1}^2}+\log\left(\frac{\sigma_{p,i-1}^2}{\sigma_{q,i}^2}\right). \tag{22}$$

For the first term, to minimize it, the model will try to make $\mu_{q,i}^2$ as close to 0 as possible. As $\sigma_{p,i-1}^2$ decreases, the impact of this term becomes more significant if $\mu_{q,i}^2$ is large.

For the second and third terms, to minimize them, the model will attempt to reduce $\sigma_{q,i}^2$ and bring it closer to $\sigma_{p,i-1}^2$. This ultimately enhances the model's sensitivity to these parameters, resulting in more accurate and stable parameter estimation for these important channels.

The above method can more accurately capture changes in critical sensors in industrial fault detection and can also improve the model's robustness and generalization ability through stronger regularization effects.

5. Experiment and Result

5.1. Experiment Preparation

5.1.1. Datasets and Processing

The data used in this paper is sourced from the SEU gearbox datasets [26]. The dataset consists of two subsets: gear data and bearing data. It was collected using a Dynamic Drive System (DDS) simulator, aiming to investigate the gearbox performance under two operating conditions: 20Hz - 0V and 30Hz - 2V. Each dataset includes five distinct operating states: four fault modes and one normal state. Thus, the fault diagnosis task based on DDS can be viewed as a five-class classification problem. The specific fault class and their corresponding data labels are shown in the Table 1.

Table 1
The specific fault class and their corresponding data labels.

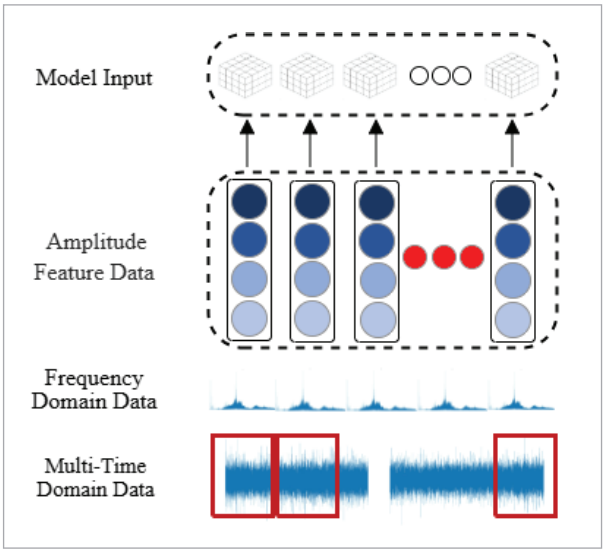
Location	Type	Description	Label
Bearing	Ball	Crack occurs in the ball	1
	Combination	Crack occurs in both inner and outer ring	2
	Inner	Crack occurs in the inner ring	3
	Outer	Crack occurs in the outer ring	4
	Health	No Crack	0
Gearbox	Chipped	Crack occurs in the gear feet	1
	Miss	Missing one of feet in the gear	2
	Root	Crack occurs in the root of gear feet	3
	Surface	Wear occurs in the surface of gear	4
	Health	No Crack	0

Each data sample consists of signals from 8 channels: Channel 1 represents the motor vibration frequency, Channels 2-4 represent the vibration frequencies of the planetary gearbox in the x, y, and z directions, respectively, Channel 5 represents the motor torque, and Channels 6-8 represent the vibration frequencies of the parallel gearbox in the x, y, and z directions, respectively. To minimize the interference from vibrations during startup and shutdown, we select a stable time segment from the middle of the data for analysis. Subsequently, a Fourier Transform is applied to the data to obtain the frequency-domain representation x_{fft}. Given the symmetry of the non-negative frequency part in the Fourier Transform results, only frequencies up to half of the sampling frequency need to be computed. A frequency array is constructed that includes all frequencies from 0 to half of the sampling frequency. Furthermore, the magnitude of x_{fft} ,i.e., the amplitude of each frequency component, is calculated and stored in the list $\text{abs}_{x_{fft}}$.

Data under the operating condition of 20Hz - 0V, the reference frequency for this condition is 20Hz. Based on this reference frequency, target frequencies of 10Hz, 20Hz, 30Hz, 40Hz, and 50Hz are chosen, corresponding to $0.5f$, $1f$, $1.5f$, $2f$, and $2.5f$ times the reference frequency. The other operating condition is similar. For each target frequency, the closest frequency value in the frequency array and its corresponding $\text{abs}_{x_{fft}}$ amplitude value are identified. This process en-

sures that we can accurately locate and analyze the vibration characteristics at each target frequency. Finally, the amplitude values corresponding to these target frequencies are aggregated to form the final dataset for experimental analysis. The above data processing procedure is shown in Figure 4.

Figure 4
Sensor data processing procedure.



5.1.2. Experimental Environment

This paper utilizes a Dell PowerEdge R740 server with advanced hardware configurations as the experimental platform. The server is powered by a dual Intel Xeon Gold 5218R processor setup and possesses 256GB of memory. Although the server is also provided with two NVIDIA Tesla T4 GPUs to manage large-scale parallel computing and deep learning tasks. The server operates on the Ubuntu 22.04 LTS operating system and runs version 1.3.0 of the PyTorch deep learning framework, which is utilized to construct and execute the experiments.

In the implementation of Bayesian Neural Networks, this paper utilizes Pyro, a contemporary probabilistic programming library [6]. Pyro is a flexible framework built atop Python and PyTorch specifically designed for the development of Bayesian deep learning models and other intricate probabilistic models, first released in 2017 by Uber AI Labs. Constructed upon the foundation of the PyTorch deep learning framework, Pyro harnesses PyTorch's dynamic computation graphs and automatic differentiation capabilities, offering users a highly adaptable and powerful platform to design, simulate, and infer complex probabilistic models. Pyro comes equipped with a rich set of tools to define random variables, construct implicit or explicit probabilistic models, and perform efficient and modular inference through methods such as variational inference and Markov Chain Monte Carlo algorithms. This integration allows researchers and developers to seamlessly blend deep learning architectures with Bayesian principles, enabling them to tackle problems with uncertainty quantification and probabilistic reasoning effectively. The BCNN of this Experiment consists of two convolutional layers and one fully connected output layer.

In this paper, a Federated Learning environment was established, involving three clients. A deep learning environment was built using PyTorch on a shared server, with PySyft simulating the federated learning environment [15] and Pyro constructing the Bayesian Learning framework. Three training programs were implemented to simulate distributed data training scenarios for the three clients.

To reflect the varying amounts of data held by each client, the data distribution across the three clients was set to a ratio of 2:1:1. Specifically, the first client has the most extensive data resources, while the second and third clients each have half the amount of data compared to the first client. Additionally, the training datasets of the three clients are completely independent, with no overlap. This data distribution design aims to simulate real-world scenarios where different clients may possess varying amounts and diversity of data, thus providing a challenging environment for evaluating the performance of Federated Learning algorithms.

In this experiment, Algorithm 1 is implemented in Python, enabling automatic analysis and integration of distribution information from the training results of multiple clients for interpretability analysis and information transfer. For clients requiring layer-by-layer analysis, the client with the largest data volume is selected by default (in this paper, Client 1). After training, Algorithm 1 uses interpretability analysis results to identify key channels and their corresponding variances and transfers this information to the target client. For other clients, Algorithm 1 directly extracts the local distribution parameters

for the same channels and transmits the channel and distribution information to the target client.

To comprehensively evaluate the performance of different methods in a Bayesian Federated Learning environment, comparative experiments are designed in this paper. The four methods are as follows:

1 Local Bayesian Learning (Local BL): Bayesian Learning with local training on all data (no federated collaboration).

2 Federated Average Bayesian Learning- Gaussian (FEDAG)[17]: Bayesian Federated Learning with centralized parameter distribution averaging.

3 Bayesian Federated Learning (BFL) [17]: Bayesian Federated Learning with full-channel distribution transmission.

4 Bayesian Federated Learning with Channel Importance (BFL-CI, ours): Bayesian Federated Learning with important-channel distribution transmission.

For all models, the learning rate is set to 0.01, the number of training epochs is 500, and the batch size is 8. When evaluating the Bayesian model, the parameters are sampled 100 times to account for uncertainty estimation and provide a probabilistic interpretation of the model's predictions. The training sets and test sets are consistent across all models.

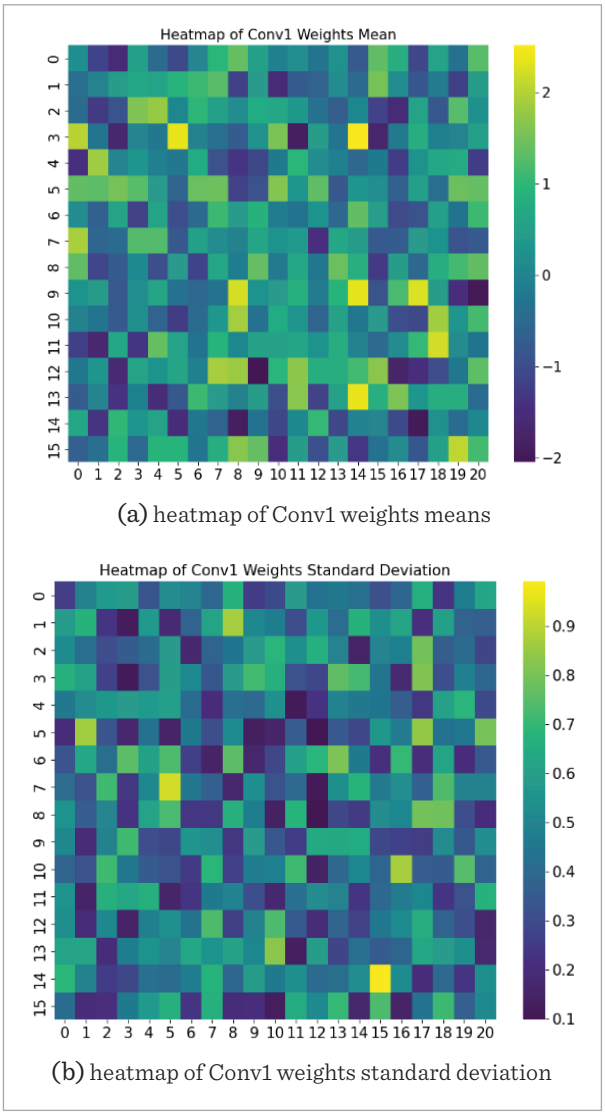
To objectively evaluate the performance of the models from each algorithm, comparisons will be made across the following dimensions: Accuracy, Receiver Operating Characteristic (ROC) curve, and confidence interval. The accuracy table presents results from both datasets; however, due to the similar performance of the two datasets in other evaluations, only the results from the bearing dataset are shown.

5.2. Experimental Result

5.2.1. of Channel Importance

The mean distribution and standard deviation distribution of the weights of each layer of the BCNN after the training of client *m* are shown in Figures 5-7, respectively. In each figure, there are two sub-figures, (a) and (b). Figure (a) represents the mean heatmap of the weights of this layer, and Figure (b) represents the variance heatmap of the weights of this layer. The abscissa in the figure corresponds to the input channels of each layer in the model. For the convolutional layers conv1 and conv2, due to the particularity of their structures, the input channels are flattened, and the abscissa represents the product of the input channels and the convolution kernels. And the ordinate corresponds to the output channels of each layer in the model. Each square in the heatmap corresponds to the value of the mean or variance corresponding to a neuron respectively. The shade of the color represents the magnitude of the value. The darker the color, the lower the value, and the lighter the color, the higher the value. In the mean heatmap, a higher value indicates a higher degree of importance. In the variance heatmap, a higher value indicates a lower degree of confidence.

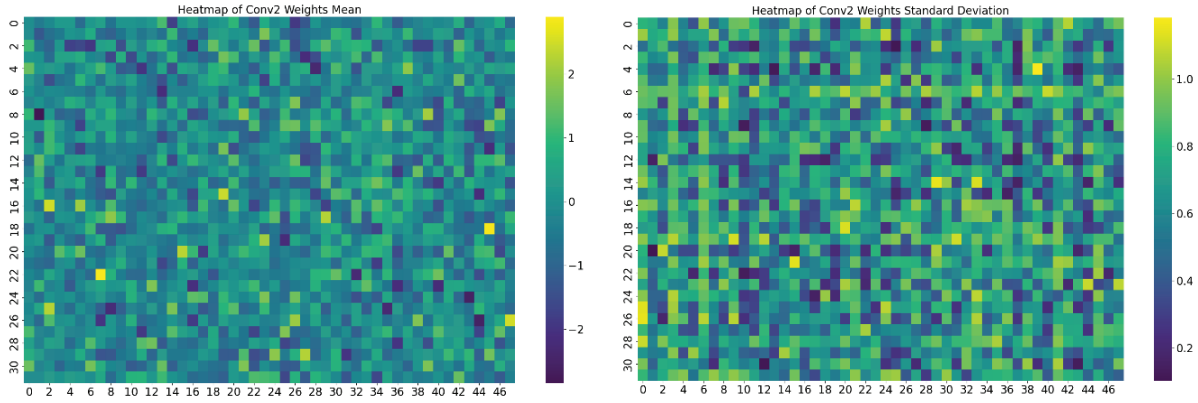
Figure 5
Weights posterior distribution of Conv1.



(a) heatmap of Conv1 weights means

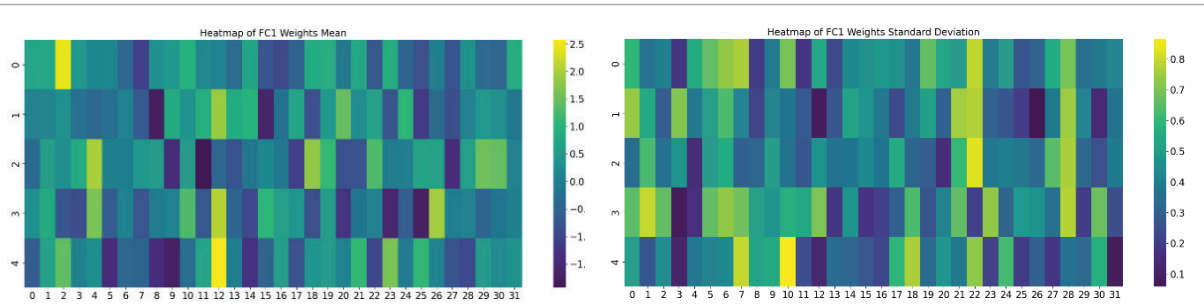
(b) heatmap of Conv1 weights standard deviation

Figure 6
Weights posterior distribution of Conv2.



(a) heatmap of Conv2 weights means (b) heatmap of Conv2 weights standard deviation

Figure 7
Weights posterior distribution of FC.



(a) heatmap of FC weights means (b) heatmap of FC weights standard deviation

In this paper, Algorithm 1 is implemented using Python. Starting from the output layer, important channels are identified based on the mean and standard deviation of the parameters in each layer, ultimately leading to the discovery of important input channels (which also correspond to sensors). The program's output indicates that Channel 2 is the significant channel in the experiments conducted in this paper. This result indicates that the data from vibration sensors in the x direction of the planetary gearbox plays an important role in the results of fault detection.

The variance of client m's channel 2 and its corresponding distribution is passed to the next client as the variance of the prior distribution of channel 2 of the next client. And so on. Eventually, the training process of all clients is completed, and then the performance test is carried out using the test set.

5.2.2. Accuracy

As shown in the following table, it presents the accuracy obtained by various methods under different operating conditions for each training set. From the data in Table 2, it can be clearly observed that, compared with other methods, the method proposed in this paper demonstrates a remarkable advantage of high accuracy under different working conditions for both bearings and gearboxes.

Bearing data under the operating condition of 20Hz - 0V is taken as an example to show the results of the

Table 2
Accuracy obtained by various methods under different operating conditions for each training set.

Methods	Client	Client2				Client3			
	Dataset	Bearing		Gearbox		Bearing		Gearbox	
	Operating conditions	20Hz 0V	30Hz 2V	20Hz 0V	30Hz 2V	20Hz 0V	30Hz 2V	20Hz 0V	30Hz 2V
Local BL		89.13%	90.17%	93.14%	94.56%	87.71%	87.9%	94.21%	94.13%
FEDAG		89.03%	92.43%	95.32%	95.07%	89.14%	90.23%	97.11%	95.83%
BFL		90.23%	92.13%	95.72%	95.67%	89.28%	89.5%	97.22%	95.94%
BFL-CI		92.84%	93.26%	96.92%	97.23%	94.84%	91.97%	98.73%	97.35%

accuracy changing with the training epoch. Figures 8-9 depict the accuracy results on the test set for Client 2 and Client 3, respectively, after training for a total of 200 epochs. It is evident that the proposed method in this paper outperforms other methods in terms of accuracy. The accuracy achieved on Client 2 is 90%, and on Client 3, it is 95%. Moreover, compared to the FEDAG and Local BL methods, the BFL method demonstrates superior performance.

5.2.3. ROC

Figures 10 and Figures 11 show the ROC curves for Client 2 and Client 3, respectively. In these figures, the Local BL ROC curve is represented by a yellow line, the FEDAG ROC curve by a green line, the BFL ROC curve by a blue line, and the BFL-CI ROC curve by a red line. All results demonstrate that the ROC curve of the proposed BFL-CI model in this paper is closest to the top-left corner, with the highest AUC value, indicating superior performance compared to other methods. For the third class of faults, all methods achieve an AUC of 1, indicating that all methods can accurately identify this class, likely due to the data features being distinctly different from other classes.

5.2.4. Confidence Interval

To clearly and intuitively illustrate the distribution of uncertainty trends associated with different classification results across the entire test set, this paper first categorizes the test set by class. Curve plots are then constructed with classes on the x-axis and the log probability distribution of model predictions for each class on the y-axis. The blue curve represents the trend of the average posterior probability for the samples, while the shaded area indicates the 95% confidence interval for each sample, highlighting the model's varying levels of uncertainty for different classification decisions. Figures 12 and 13 show the confidence interval plots for each fault category using the proposed method for Clients 2 and 3, respectively.

Figure 8
The accuracy of Client 2.

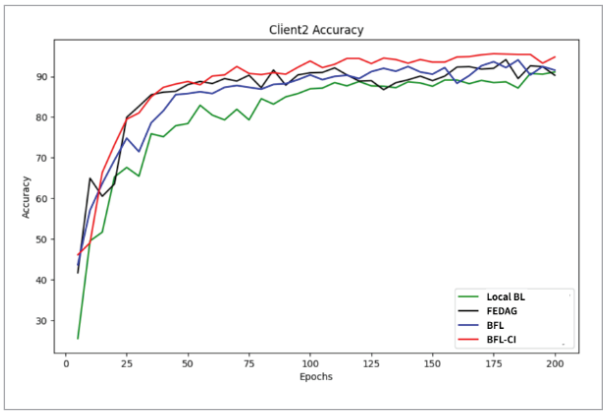


Figure 9
The accuracy of Client 3.

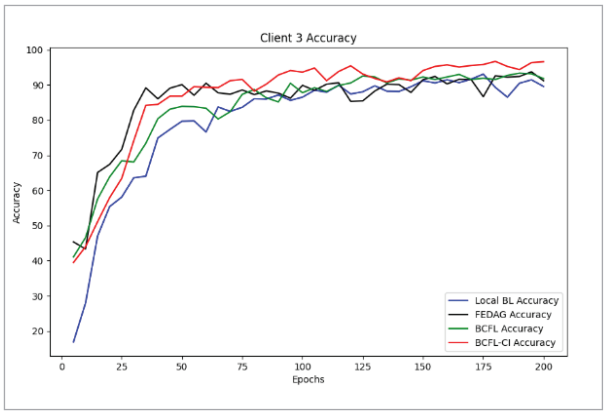
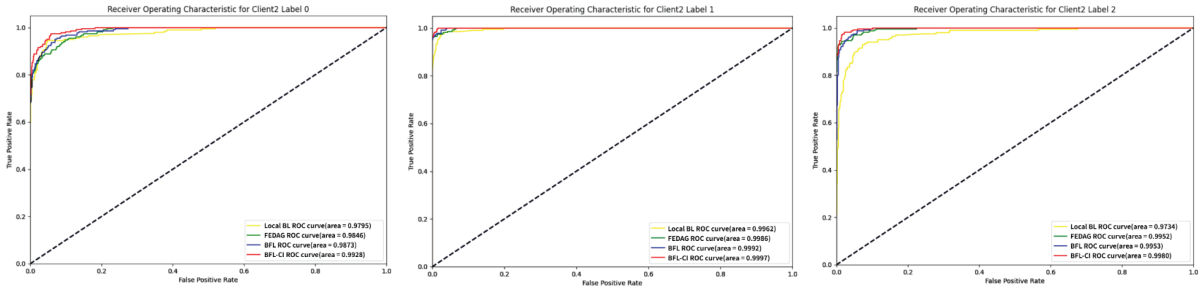


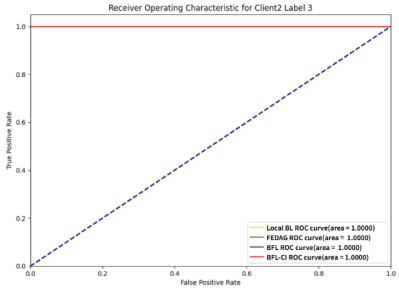
Figure 10
ROC of Client 2.



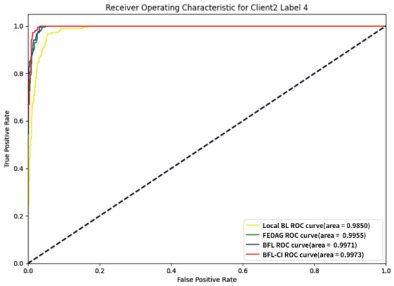
(a) ROC for label 0

(b) ROC for label 1

(c) ROC for label 2

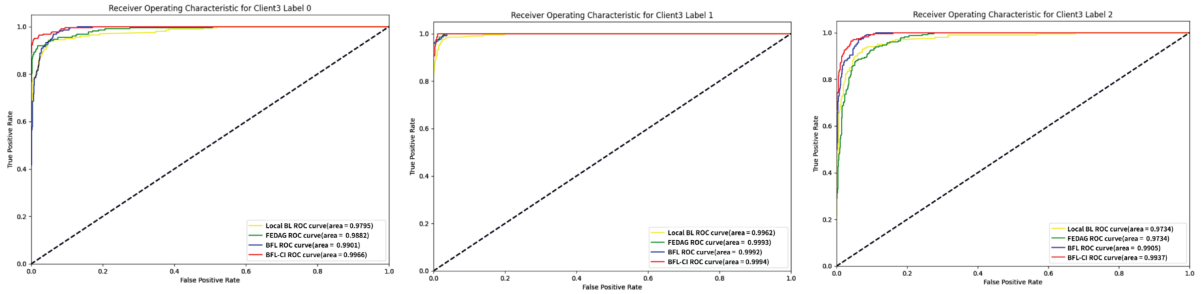


(d) ROC for label 3



(e) ROC for label 4

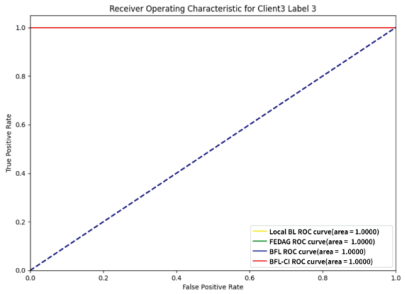
Figure 11
ROC of Client 3.



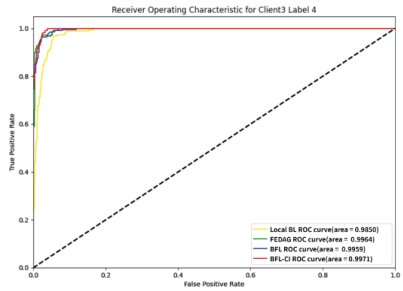
(a) ROC for label 0

(b) ROC for label 1

(c) ROC for label 2

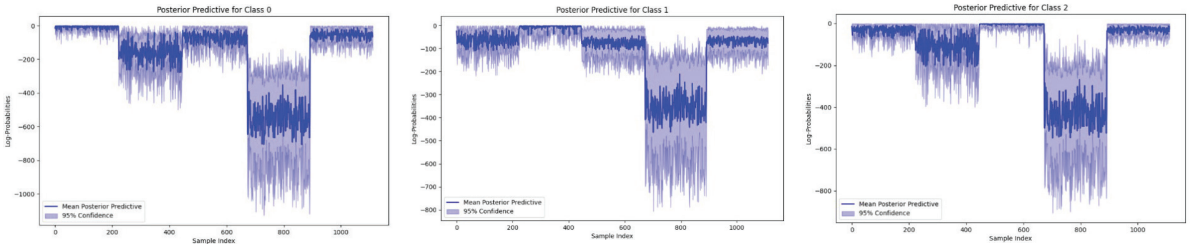


(d) ROC for label 3

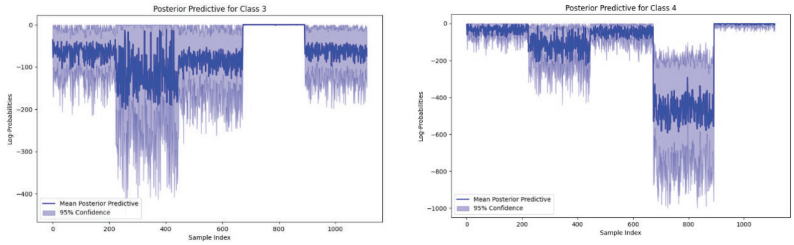


(e) ROC for label 4

Figure 12
Confidence interval of Client 2.

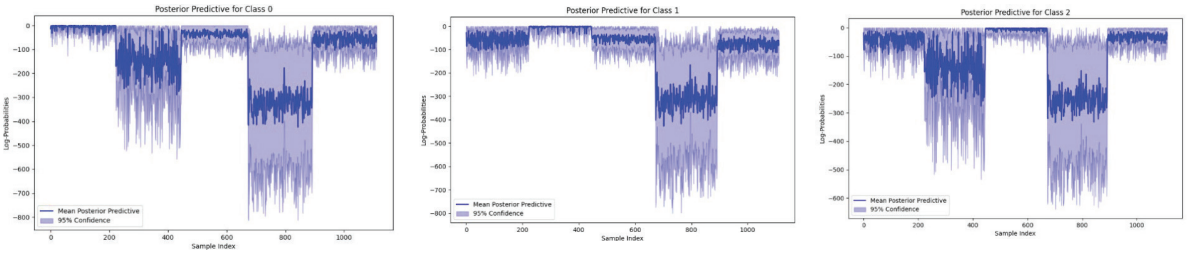


(a) confidence interval for Class 0 (b) confidence interval for Class 1 (c) confidence interval for Class 2

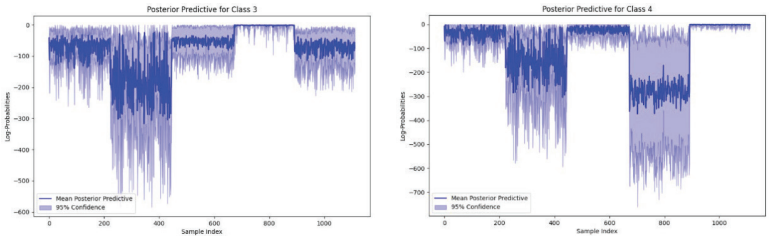


(d) confidence interval for Class 3 (e) confidence interval for Class 4

Figure 13
Confidence interval of Client 3.



(a) confidence interval for Class 0 (b) confidence interval for Class 1 (c) confidence interval for Class 2



(d) confidence interval for Class 3 (e) confidence interval for Class 4

From the figures, for each fault class, the model generates mean log-probabilities that reach their highest levels for the correct class, and these log-probabilities are accompanied by relatively narrow confidence intervals. This phenomenon strongly suggests that the model shows a high level of accuracy and strong certainty when predicting various fault classes.

In the confidence interval plot for Class 3 faults, the uncertainties for other classes are very evident, while Class 3 exhibits a smooth curve with the narrowest interval. This indicates that the model's detection of Class 3 is both the most accurate and the most confident, which is consistent with the ROC curve results showing that the model can identify this class with complete certainty, achieving a 100% recognition rate. Furthermore, it is observed that the curves for Classes 0, 2, and 4 are relatively closer to each other, suggesting potential confusion between these classes due to their similar features. Therefore, greater attention should be paid to the detection results of these three classes during subsequent testing processes.

5.2.5. Analysis of Experimental Results

Although faults can be detected through vibration signals and it can be known from ordinary frequency analysis that the data in the radial direction is more sensitive to the detection results. However, this paper obtains mowre accurate channel importance results through interpretability analysis. Since the inner ring of the bearing is tightly fitted with the shaft, the rotational speed and power transmission of the shaft directly act on the inner ring. The impact of inner ring faults on vibration is more obvious, so the detection accuracy of the model reaches 100%. However, the phenomena of outer ring faults and simultaneous inner and outer ring faults are relatively similar, and the signal characteristics are rather complex. Whether viewed from the results of channel importance analysis or from the detection performance results such as accuracy and ROC, the experimental results in this paper are consistent with the theoretical concept results, which proves the feasibility of the method in this paper. Then, the variance of the important channels is transmitted to other clients, which not only realizes Bayesian Federated Learning but also improves the detection performance.

6. Discussion and Conclusion

This paper proposes an industrial equipment fault detection method based on Bayesian Federated Learning (BFL), combining the uncertainty modeling advantages of Bayesian Learning with the distributed training characteristics of Federated Learning to effectively address data privacy and model generalization issues in industrial equipment fault detection. The proposed method of passing channel distributions based on explainability analysis not only ensures the learning capabilities of each industrial node but also significantly reduces the transmission of parameters across the network, thereby further protecting data privacy. Through experiments, this paper demonstrates the feasibility of the proposed method and its performance in detecting faults. Therefore, this approach not only quantifies the uncertainty in industrial equipment fault detection but also allows the model to learn from a broader range of data sources, adapting to cross-regional industrial environments.

In Federated Learning, although the order of client participation does not affect the final detection results, optimizing the scheduling sequence by considering factors such as each client's computing power and network conditions can effectively improve the training efficiency of BFL. This direction deserves further in-depth research in the future. Industrial equipment typically generates various types of data, including images, sounds, vibration signals, etc. Current BFL methods primarily focus on processing single-type sensor data, lacking effective integration of multimodal data. Future research can aim to develop BFL frameworks capable of handling multimodal data to fully utilize complementary information among different types of data. This not only requires designing algorithms that can fuse multiple data types but also necessitates considering synchronization and calibration issues between different modalities. Although BFL excels in handling data uncertainties and dynamics, its interpretability regarding the decision-making process, especially when the model makes incorrect predictions, still needs improvement. Future research should explore how to combine existing explainability methods (such as SHAP values, LIME, etc.) with BFL to enhance model transparency.

Data Sharing Agreement

The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, author-ship, and publication of this article.

Acknowledgments

The authors acknowledge the Zhejiang Provincial Key Science and Technology 'LingYan' Project Foundation (2023C01145), Zhejiang Gongshang University Higher Education Research Projects (Xgy22028).

Authors' contributions

Conceptualization, Zhongyun Tang and Hanyi Xu; methodology, Hanyi Xu; software, Zhongyun Tang; validation, Haiyang Hu, Chonghuan Xu and Weiyu Zhang; formal analysis, Chonghuan Xu and Weiyu Zhang; investigation, Haiyang Hu; resources, Hanyi Xu; data curation, Chonghuan Xu and Weiyu Zhang; writing—original draft preparation, Zhongyun Tang; writing—review and editing, Zhongyun Tang; visualization, Chonghuan Xu; supervision, Hanyi Xu and Weiyu Zhang; project administration, Haiyang Hu; funding acquisition, Haiyang Hu. All authors have read and agreed to the published version of the manuscript.

References

1. Arnaiz-González, Á., Ramírez-Sanz, J. M., Maestro-Prieto, J., Bustillo, A. Semi-supervised Learning for Industrial Fault Detection and Diagnosis: A Systemic Review. ISA Transactions, 2023, 143, 255-270.https://doi.org/10.1016/j.isatra.2023.09.027
2. Atzmueller, M., Hoogen, J. V. D., Hudson, D., Bloemheuvel, S. Hyperparameter Analysis of Wide-Kernel CNN Architectures in Industrial Fault Detection: An Exploratory Study. International Journal of Data Science and Analytics, 2024, 18, 423-444. https://doi.org/10.1007/s41060-023-00440-6
3. Balaji, V., Clare, M. C. A., Sonnewald, M., Lguensat, R., Deshayes, J. Explainable Artificial Intelligence for Bayesian Neural Networks: Toward Trustworthy Predictions of Ocean Dynamics. Journal of Advances in Modeling Earth Systems, 2022, 14(11), e2022MS003162. https://doi.org/10.1029/2022MS003162
4. Bhatt, S., Gupta, A., Rai, P. Bayesian Federated Learning via Predictive Distribution Distillation. Asian Conference on Machine Learning, 2023.
5. Bhatt, S., Gupta, A., Rai, P. Federated Learning with Uncertainty via Distilled Predictions. Proceedings of Machine Learning Research, 2024, 222, 153-168.
6. Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., Goodman, N. D. Pyro: Deep Universal Probabilistic Programming. Journal of Machine Learning Research, 2019, 20(28), 1-6. https://doi.org/10.1145/3315508.3329974
7. Bizhani, M., Kuru, E. Towards Drilling Rate of Penetration Prediction: Bayesian Neural Networks for Uncertainty Quantification. Journal of Petroleum Science and Engineering, 2022, 219, 111068. https://doi.org/10.1016/j.petrol.2022.111068
8. Bykov, K., Höhne, M. M.-C., Creosteanu, A., Müller, K.-R., Klauschen, F., Nakajima, S., Kloft, M. Explaining Bayesian Neural Networks. arXiv Preprint arXiv:2108.10346, 2021.
9. Cao, L., Chen, H., Fan, X., Gama, J., Ong, Y., Kumar, V. Bayesian Federated Learning: A Survey. Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, 2023, 10, 7233-7242.
10. Chadha, M., Najera-Flores, D.A., Hu, Z., Todd, M.D. A Physics-Constrained Bayesian Neural Network for Battery Remaining Useful Life Prediction. Applied Mathematical Modelling, 2023, 122, 42-59. https://doi.org/10.1016/j.apm.2023.05.038
11. Deng, X., Shi, Y., Yao, D. Outlier Detection Toward High-Dimensional Industrial Data Using Extreme Tensor-Train Learning Machine with Compression. Journal of King Saud University - Computer and Information Sciences, 2023, 35, 101576. https://doi.org/10.1016/j.jksuci.2023.101576
12. Gelman, A., Guo, H., Greengard, P., Wang, H., Kim, Y., Xing, E. P. Federated Learning as Variational Inference: A Scalable Expectation Propagation Approach. ICLR, 2024.
13. Ghahramani, Z., Gal, Y. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. International Conference on Machine Learning, 2016, 3, 1651-1680.
14. Gong, J., Simeone, O., Kang, J. Bayesian Variational Federated Learning and Unlearning in Decentralized Networks.

IEEE International Workshop on Signal Processing Advances in Wireless Communications, 2021, 5, 216-220. https://doi.org/10.1109/SPAWC51858.2021.9593225

15. Kaissis, G., Ziller, A., Trask, A., Lopardo, A., Szymkow, B., Wagner, B., Bluemke, E., Nounahon, J., Passerat-Palmbach, J., Prakash, K., Rose, N., Ryffel, T., Reza, Z. N. PySyft: A Library for Easy Federated Learning. Federated Learning Systems, 2021, 965, 111-139. https://doi.org/10.1007/978-3-030-70604-3_5
16. Kaur, H., Rani, V., Kumar, M., Sachdeva, M., Mittal, A., Kumar, K. Federated Learning: A Comprehensive Review of Recent Advances and Applications. Multimedia Tools and Applications, 2024, 83, 54165-54188. https://doi.org/10.1007/s11042-023-17737-0
17. Leng, J., Li, R., Xie, J., Zhou, X., Li, X., Liu, Q., Chen, X., Shen, W., Wang, L. Federated Learning-Empowered Smart Manufacturing and Product Lifecycle Management: A Review. Advanced Engineering Informatics, 2025, 65, PA. https://doi.org/10.1016/j.aei.2025.103179
18. Li, T., Li, W., Li, H., Gu, S. Process Fault Diagnosis with Model-and Knowledge-Based Approaches: Advances and Opportunities. Control Engineering Practice, 2020, 105, 104637. https://doi.org/10.1016/j.conengprac.2020.104637
19. Liu, L., Jiang, X., Zheng, F., Chen, H., Qi, G.-J., Huang, H., Shao, L. A Bayesian Federated Learning Framework with Online Laplace Approximation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(1), 1-16. https://doi.org/10.1109/TPAMI.2023.3322743
20. Liu, L., Li, J., Lguensat, J., Wang, J., Zhao, S., Lu, Q. Privacy-Preserving and Secure Industrial Big Data Analytics: A Survey and the Research Framework. IEEE Internet of Things Journal, 2024, 11(11), 18976-18999.https://doi.org/10.1109/JIOT.2024.3353727
21. Mou, L., Liang, L., Gao, Z., Zhang, X., Shao, Y. A Multi-Scale Anomaly Detection Framework for Retinal OCT Images Based on the Bayesian Neural Network. Biomedical Signal Processing and Control, 2022, 75, 103619. https://doi.org/10.1016/j.bspc.2022.103619
22. Ohno, K., Sholahudin, Giannetti, N., Yamaguchi, S., Saito, K. Dynamic Modeling of Room Temperature and Thermodynamic Efficiency for Direct Expansion Air Conditioning Systems Using Bayesian Neural Network. Applied Thermal Engineering, 2019, 158, 113809. https://doi.org/10.1016/j.applthermaleng.2019.113809
23. Olivier, A., Shields, M. D., Graham-Brady, L. Bayesian Neural Networks for Uncertainty Quantification in Data-Driven Materials Modelling. Computer Methods in Applied Mechanics and Engineering, 2021, 386, 114079. https://doi.org/10.1016/j.cma.2021.114079
24. Rasmussen, C. E., Williams, C. K. I. Gaussian Processes for Regression. Advances in Neural Information Processing Systems, 1995, 8, 514-520.
25. Sabry, A. H., Ungku Amirulddin, U. A. B. A Review on Fault Detection and Diagnosis of Industrial Robots and Multi-Axis Machines. Results in Engineering, 2024, 23, 102397. https://doi.org/10.1016/j.rineng.2024.102397
26. SEU Gearbox Datasets. [Online]. Available: https://github.com/cathysiyu/Mechanical-datasets (Accessed September 2019).
27. Shao, Y., Zhang, X., Li, Y., Li, W., Guo, K. Personalized Federated Learning via Variational Bayesian Inference. International Conference on Machine Learning, 2022, 162, 26293-26310.
28. Varalakshmi, B. D., Lingaraju, G. M. Enhancing Industrial Anomaly Detection with Auto Encoder-Based Temporal Convolutional Networks for Motor Fault Classification. SN Computer Science, 2024, 5, 1067. https://doi.org/10.1007/s42979-024-03425-9
29. Yuan, L., Wang, Z., Sun, L., Yu, P. S., Brinton, C.G. Decentralized Federated Learning: A Survey and Perspective. IEEE Internet of Things Journal, 2024, 11(21), 34617-34638. https://doi.org/10.1109/JIOT.2024.3407584
30. Yurdem, B., Kuzlu, M., Gullu, M. K., Catak, F. O., Tabassum, M. Federated Learning: Overview, Strategies, Applications, Tools and Future Directions. Heliyon, 2024, 10(19), e38137. https://doi.org/10.1016/j.heliyon.2024.e38137
31. Zhang, X., Zou, Y., Li, S. Bayesian Neural Network with Efficient Priors for Online Quality Prediction. Digital Chemical Engineering, 2022, 2, 100008. https://doi.org/10.1016/j.dche.2021.100008
32. Zhao, Z., Luo, M., Ding, W. Deep Leakage from Model in Federated Learning. Conference on Parsimony and Learning, 2024, 234, 324-340.
33. Zhou, T., Han, T., Droguett, E.L. Towards Trustworthy Machine Fault Diagnosis: A Probabilistic Bayesian Deep Learning Framework. Reliability Engineering & System Safety, 2022, 224, 108525. https://doi.org/10.1016/j.ress.2022.108525
34. Zhou, T., Zhang, L., Han, T., Droguett, E. L., Mosleh, A., Chan, F.T.S. An Uncertainty-Informed Framework for Trustworthy Fault Diagnosis in Safety-Critical Applications. Reliability Engineering & System Safety, 2023, 229, 108865. https://doi.org/10.1016/j.ress.2022.108865