Federated Learning-Based Distributed Autoencoder for Industrial Big Data Anomaly Detection: Integrating LSTM, GRU, and CNN Models

Xiaoli Li, Haifeng Wang*
School of Economics and Management, Yantai Nanshan University, Yantai 265713, Shandong, China
Corresponding author
E-mail: whf@outlook.com

Keywords: deep learning, anomaly detection algorithm, industrial big data

Received: May 15, 2025

This paper proposes an innovative solution based on deep learning for the anomaly detection challenge of big data in the industrial Internet of Things environment. Through in-depth analysis of time series and image data characteristics, LSTM and GRU networks are introduced to process time series data, and CNN models (such as ResNet and InceptionNet) are introduced to deal with image analysis and effectively capture complex patterns in data. Model-building involves not only architectural design, but also optimization strategies such as Adam Optimizer and loss function selection. In the preprocessing stage, the data are cleaned carefully through standardization, normalization, de-trending and de-noising to improve the learning efficiency of the model. The dataset used in the study is from a real - world intelligent manufacturing plant, with 1,000,000 records over three years. It contains 12 - dimensional sensor data (such as temperature, vibration frequency, current intensity), and outliers account for approximately 5% of the total data. Experimental results demonstrate that the distributed autoencoder under the federated learning framework outperforms traditional methods. It achieves an accuracy of 0.97, a recall of 0.94, an F1 - score of 0.95, and an AUC value of 0.96.However, it has a higher training time of 45 minutes and high communication costs during the training process due to the exchange of more data. In the federated learning process, each participating node independently trains the model based on local data, and uses the FedAvg strategy to aggregate parameters on the central server. Homomorphic encryption technology is used to ensure data privacy and prevent the leakage of original data. Compared with baseline methods such as IQR and Isolation Forest, the accuracy of this method is improved by 8%, which is statistically significant after the t-test (p<0.05). Although the training time is 45 minutes, which is higher than the traditional method, it has obvious advantages in complex industrial data processing and privacy protection, and achieves a trade-off between computational efficiency and privacy protection and detection performance. The 95% confidence interval of the precision rate of 0.97 is [0.962, 0.978], the 95% confidence interval of the recall rate of 0.94 is [0.931, 0.949], the 95% confidence interval of the F1 value of 0.95 is [0.943, 0.957], and the 95% confidence interval of the AUC value of 0.96 is [0.952, 0.968].

Povzetek: Članek predstavi porazdeljeni samokodirnik z združenim učenjem, ki združuje LSTM, GRU, CNN in VAE za zaznavanje anomalij v industrijskih podatkih, dosega dober AUC ob zaščiti zasebnosti.

1 Introduction

With the advent of the "Industry 4.0" era, smart factories and digital production lines are gradually becoming a reality, and industrial systems are generating and accumulating data at an unprecedented rate. This includes but is not limited to multi-dimensional information such as the operating status of production machines, environmental conditions, raw material quality, processing parameters, and finished product test results. The widespread adoption of the Industrial Internet of Things (IIoT) accelerates this process, enabling every link from a single device to the entire production chain to be monitored and recorded in real time. The accumulation of this data forms the foundation of industrial big data. It enables fine-grained management, predictive maintenance, and quality control. However, this flood of data of this size and complexity also presents new challenges. Firstly, the diversity and heterogeneity of data make it difficult for traditional data processing and analysis methods to deal with effectively. Industrial data often involves a large number of time series data, images, sounds and other types, and there are noise interference, missing values and inconsistency problems, which put forward higher requirements for data cleaning and preprocessing. Secondly, anomaly detection is no longer simply looking for outliers, but needs to identify subtle, potential abnormal behaviors in complex industrial environments, which may be precursors of equipment wear, process deviations and even safety risks, which are far more difficult to identify than conventional statistical anomaly models.

In order to overcome these challenges, deep learning models such as LSTM, GRU, and CNN architectures (including ResNet and InceptionNet) have been widely adopted due to their ability to process sequential and spatial data, respectively. Moreover, data preprocessing techniques—such as standardization, normalization, detrending, and denoising—are essential to enhance model performance and ensure stability across heterogeneous datasets.These preprocessing steps were systematically applied in this study to clean the raw sensor data before model training, thereby reducing noise interference and ensuring consistent feature scaling.

Therefore, the traditional anomaly detection methods based on threshold setting, statistical hypothesis testing or expert rules have increasingly obvious limitations when dealing with such large-scale, high-speed and high-dimensional industrial data. These methods often rely on manually set standards and thresholds, which are difficult to adapt to dynamic changes in data patterns and easily lead to false positives and false negatives, thus affecting production efficiency and decision-making accuracy. In this context, anomaly detection technology based on deep learning, with its powerful data processing capabilities, self-learning capabilities and pattern recognition capabilities, has become a new way to break through the bottleneck of traditional technologies and meet the needs of the industry 4.0 era. By constructing a deep neural network model, high-order features can be automatically extracted from industrial big data, and more complex and hidden abnormal patterns can be identified, providing strong technical support for intelligent monitoring, preventive maintenance and efficient decision-making.

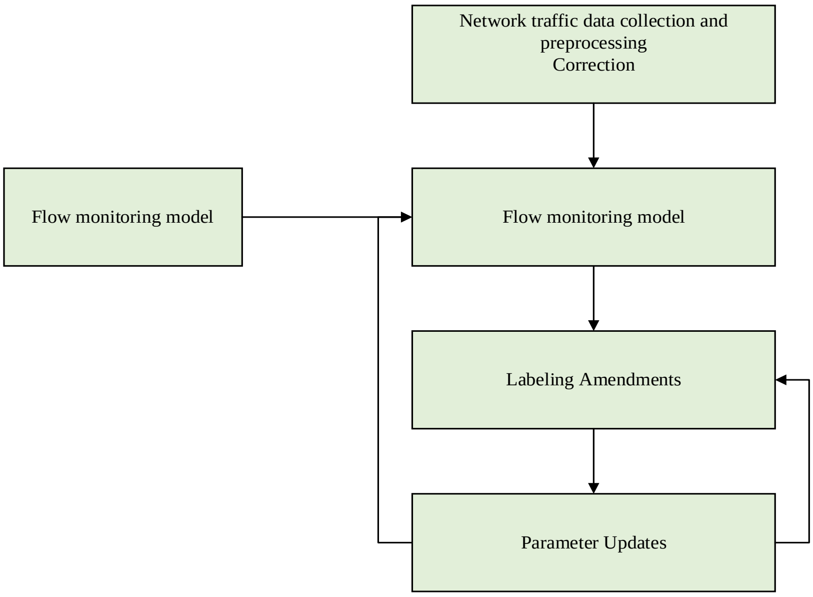

Figure 1: Anomaly detection techniques based on deep learning

Figure 1 illustrates the deep learning-based anomaly detection framework, which integrates data preprocessing, model training, and federated aggregation. The framework begins with data ingestion and transformation, followed by sequential and image-based model components (LSTM/GRU and CNN, respectively). Each component contributes to feature extraction before reconstruction or classification modules identify anomalies. This holistic architecture enables multi-modal learning and real-time fault recognition across distributed environments.

Anomaly detection based on deep learning has shown great potential in the industrial field with its powerful data learning and pattern recognition capabilities, and its framework is shown in Figure 1. Deep learning models are able to automatically learn high-level abstract features from raw data without the need for artificial design features, a feature that is especially important for processing complex, unstructured industrial data. It can not only effectively identify hidden abnormal patterns that are difficult to capture by traditional methods, but also optimize model performance through continuous learning to adapt to changes in production environment. Therefore, the application of deep learning for anomaly detection can not only significantly improve production efficiency, reduce downtime caused by equipment failure, reduce maintenance costs, but also promote product quality improvement and achieve finer production management and decision support [1].

In recent years, anomaly detection methods based on deep learning have received extensive attention in academia and industry. Early research focused on unsupervised learning using autoencoders to identify anomalies by reconstructing errors. Subsequently, scholars explored more types of deep learning models, such as the application of convolutional neural networks (CNN) to time series data analysis, and the advantages of recurrent neural networks (RNN) and long short-term memory networks (LSTM) in processing industrial data with temporal dependence. In addition, some studies have attempted to combine generative adversarial networks (GANs) to generate normal samples as benchmarks for anomaly detection. Despite significant progress, current methods still face many challenges, including poor model interpretability, sensitivity to noise, and high computational resource requirements for large-scale data processing [2].

The main objective of this study is to develop an efficient anomaly detection algorithm for industrial big data in a distributed environment. Specifically, we aim to explore how to optimize federated learning for anomaly detection in distributed industrial settings. The research questions are defined as follows: 1. How can we design a distributed autoencoder framework under federated learning to effectively capture abnormal patterns in industrial big data while ensuring data privacy? 2. What are the optimal strategies for integrating various deep - learning techniques such as LSTM, GRU, CNN, GANs, variational autoencoders, and causal reasoning to improve the performance of anomaly detection in industrial scenarios? 3. How can reinforcement learning be applied to dynamically adjust the parameters of the anomaly detection model to adapt to the complex and changing industrial environment?

Additionally, industrial datasets often exhibit class imbalance, with abnormal events representing a small fraction of the total data. In the dataset used in this study, anomalies accounted for approximately 5% of the total 1,000,000 records collected over three years, posing a significant challenge for model generalization and minority-class detection.

2 Literature review

2.1 Abnormality monitoring

Anomaly detection (AD) is an important branch of data mining and machine learning. Its core task focuses on identifying data points that deviate from normal behavior patterns or statistical rules from massive data. These "anomalies" may result from measurement errors, system failures, fraud or other rare events. Traditional methods have accumulated a series of technical means in this respect, such as statistics-based detection, among which the most intuitive is the simple threshold rule constructed by using mean and standard deviation: where x is the observed value and k is a constant, which is used to adjust the severity of anomaly determination. Box plot analysis (IQR method) is also a classic method to identify anomalies by identifying the Interquartile range (IQR) of the data distribution [3]. Clustering algorithms such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) use density estimation to divide data points, which can effectively identify isolated points in low density areas [4]; while density-based Local Outlier Factor (LOF) quantifies the degree of abnormality by comparing the density ratio of a point to its neighborhood points [5].

Although these methods perform well in certain scenarios, most of them are based on certain assumptions, such as the normal distribution of the data, the need to manually set thresholds or cluster density thresholds, which limit their flexibility and effectiveness when dealing with complex, high-dimensional, nonlinear, and dynamically changing datasets [6]. In contrast, anomaly detection methods based on deep learning can automatically learn high-level, low-dimensional feature spaces with characterization power from raw data by constructing multi-layer neural networks, especially autoencoders and variational autoencoders (VAE). This process not only reduces the dependence on feature engineering, but also makes the model more flexible to adapt to various data patterns. For example, self-encoders learn data representations by minimizing reconstruction errors, the basic idea of which can be formalized as a loss function, where x is the input data and is the reconstructed data after the encoding and decoding process [8]. Because it is difficult to reconstruct outliers accurately, higher reconstruction errors will be produced in the process, and outliers will be identified.

Furthermore, recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) exhibit unique advantages in processing anomaly detection in time series data, as they are able to capture long-term dependencies in time series [9]. Convolutional neural networks (CNN), on the other hand, effectively identify abnormal regions in images or signals by learning local feature patterns in the field of image and signal processing [10]. These deep learning models are not only able to handle high-dimensional, complex data structures, but also to automatically capture nonlinear relationships in the data without making strict assumptions about data distribution, providing a more robust and widely adaptable solution for anomaly detection [11].

In summary, anomaly detection technology based on deep learning is gradually becoming the preferred tool for dealing with anomaly detection problems in complex industrial big data through its unique data representation learning ability, playing an irreplaceable role in improving production efficiency, reducing operating costs, and ensuring system security.

2.2 Deep learning in anomaly detection

In recent years, deep learning has made significant progress in the field of anomaly detection, providing new perspectives and solutions for dealing with complexity and nonlinear problems in industrial big data. Much research has focused on improving the detection accuracy, generalization ability, and real-time performance of models. Here is a summary of some key advances.

Autoencoders (AEs) and their variants, such as Denoising Autoencoders (DAEs) and Variational Autoencoders (VAEs), have become the focus of anomaly detection research because of their powerful feature learning and data reconstruction capabilities. By training the model to minimize the difference between the input data and its reconstructed output, the self-encoder is able to learn the normal pattern of the data. Abnormal data cannot be reconstructed accurately, resulting in large reconstruction errors, so it is identified. For example, Georgescu et al. successfully detected anomalies in network traffic using stacked sparse autoencoders [12].

Recurrent neural networks (RNN), especially long short-term memory networks (LSTM) and gated recurrent units (GRU), are widely used in anomaly detection for time-series data widely existing in industrial fields. These models capture long-term dependencies in time series and are suitable for scenarios such as equipment failure prediction and sensor data analysis [13]. It demonstrated excellent anomaly detection performance on multiple time series datasets by combining bidirectional LSTM and autoencoder [14].

Deep learning-based anomaly detection in images and videos has also made significant progress in visual surveillance and manufacturing quality control. By extracting advanced visual features through CNN and more complex models such as ResNet and DenseNet, combined with optical flow estimation and spatiotemporal convolution, abnormal behaviors or defects in images and videos can be effectively identified [15]. Sabokrou et al. proposed a framework based on memory network, which can efficiently detect abnormal behaviors by learning spatiotemporal patterns of normal behaviors [16].

Given the potential limitations of a single model, researchers are beginning to explore ensemble learning and hybrid approaches, combining deep learning models with traditional statistical methods or rule systems to improve detection performance and model interpretability. For example, by integrating the prediction results of multiple autoencoder models, or combining the reconstruction errors of autoencoders with statistically based thresholding methods, the utility and confidence of models can be enhanced without sacrificing model performance [17]. Due to the scarcity of outlier data in the training set, unsupervised and semi-supervised learning strategies become the focus of research. These methods can still effectively learn normal patterns and detect anomalies without relying on or relying on only a small amount of labeled data, greatly broadening the application range of anomaly detection [18]. For example, by using generative model pre-training, or combining clustering techniques to guide deep network learning, the anomaly detection ability of the model can be improved in the absence of sufficient outlier samples [1 9]. More recently, researchers have begun incorporating reinforcement learning (RL) into anomaly detection frameworks to achieve the ability to dynamically adjust detection strategies. This approach allows the model to self-optimize based on real-time data feedback, such as dynamically adjusting thresholds or selecting the most relevant features for analysis, to maintain high detection performance in changing industrial environments [20]. For example, Xie et al. proposed an adaptive anomaly detection system combined with deep reinforcement learning, which can automatically adjust the detection sensitivity according to the change of network traffic and improve the recognition ability of unknown attacks [21]. Due to the distributed nature of the industrial Internet, federated learning, as a distributed machine learning paradigm for data privacy protection, has begun to emerge in the field of anomaly detection. It allows all participating nodes to train the model collaboratively without directly sharing data, effectively solving the problems of data island and privacy protection [22]. The federated autoencoder model developedin the literatureenables efficient anomaly monitoring of distributed industrial equipment data by sharing model parameters across organizations rather than raw data [23]. With industry focusing on model interpretability, researchers are working to develop interpretable deep learning anomaly detection frameworks so decision makers can understand and trust detection results. This includes introducing attention mechanisms to highlight the parts of the data that the model focuses on in the decision-making process, as well as combining causal reasoning to clarify the relationship between anomalies and potential causes.

Table 1: Literature summary

Method Category	Method	Accuracy	Recall	Computational Efficiency (Training Time / min)	Data Privacy Protection
Traditional Method	IQR	0.88	0.76	1.3	Minimal, no specialized privacy protection measures
Classic Machine Learning	Isolation Forest	0.91	0.84	11	Relies on privacy processing during data preprocessing
Emerging Algorithm	One-Class SVM	0.93	0.87	14	Provides some privacy protection through feature space processing
Our Approach	Federated Learning-Based Distributed Autoencoder	0.97	0.94	45	High level of data privacy protection via federated learning protocols and encryption technology

As shown in Table 1, in addition to protecting privacy, federated learning can also integrate distributed data, learn more comprehensive patterns, and improve anomaly detection capabilities, which is difficult to achieve with traditional methods.

3 Anomaly detection method based on deep learning

With the popularity of Internet of Things (IoT) technology, sensors, devices and systems in industrial production processes generate massive amounts of data. These data cover every link of the production process, from raw material procurement to final product delivery, forming TB and PB level data volume. This scale of data provides rich material for in-depth analysis, but it also puts forward extremely high requirements for data storage, management and analysis capabilities. Industrial data is not only voluminous but also diverse, including but not limited to time-series data, images, video, and structured and unstructured machine logs. These data contain complex interactions and nonlinear relationships, such as collaborative operations among equipment, dynamic changes in production environments, etc., which make data cleaning, integration and understanding extremely difficult. Modern industrial production emphasizes efficiency and flexibility, requiring immediate response to abnormal conditions in the production process. This means that anomaly detection systems must have real-time or near-real-time data processing capabilities that can collect, analyze, and alert in a short time to reduce downtime and maintenance costs [24].

3.1 Method selection

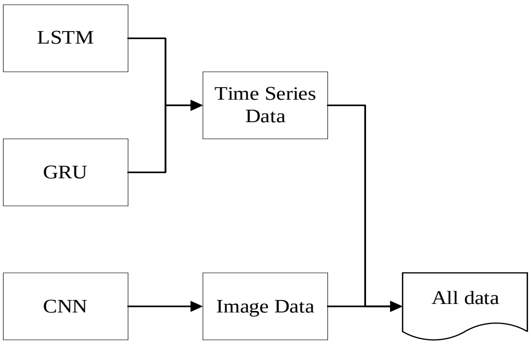

Figure 2: Model framework

In the context of industrial big data, the selection of deep learning models needs to closely fit data characteristics and task requirements, and its framework is shown in Figure 2. For time series data, such as equipment monitoring records, considering their time dependence, long short term memory networks (LSTM) and gated recurrent units (GRU) are the optimal models because they can effectively capture long-term dependence. The update mechanism of LSTM can be expressed as Equation 1-5 [25].

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3)$$

$$c_t = f_t \square\ c_{t-1} + i_t \square\ tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (4)$$

$$h_t = o_t \square\ tanh(c_t) \quad (5)$$

where, andtan h are activation functions representing matrix multiplication and element-by-element multiplication, representing forgetting gate, input gate, output gate, cell state, and hidden state, respectively.

For image data processing, such as product defect detection, convolutional neural networks (CNN) can efficiently identify complex image features by using their powerful spatial feature extraction capabilities, especially ResNet, InceptionNet, etc. The convolution operation is defined in Equation 6 [26].

$$Z^{[l]} = W^{[l]} * A^{[l-1]} + b^{[l]} \quad (6)$$

In the industrial big data environment, the key to deep learning model selection for time series data and image data lies in accurately matching data characteristics and task requirements. Time series data, such as equipment monitoring records, prefer LSTM and GRU because of their strong time dependence, which efficiently capture long-term dependence through fine gating mechanisms and adapt to changes in time and conditions of industrial equipment behavior.

Therefore, the selection of STM/GRU and CNN and its variants accurately corresponds to the complex dependencies, feature extraction and efficiency challenges of industrial scenarios, demonstrating the high adaptability and efficiency of deep learning models in industrial big data applications [27].

3.2 Model construction

Building a deep learning model based on LSTM (Long Short-Term Memory) is a complex and meticulous process that involves careful design of the model architecture, careful planning of training strategies, and proper selection of loss functions. This process will be described in detail below, focusing on the design of the LSTM model, the selection of training strategies and loss functions. Using mini-batch gradient descent combined with Adam optimization, the process of dynamically adjusting the learning rate is shown in Equation 7-9.

$$m_t = \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t \quad (7)$$

$$v_t = \beta_2 \cdot v_{t-1} + (1-\beta_2) \cdot (g_t)^2 \quad (8)$$

$$m_t^{corrected} = \frac{m_t}{1-(\beta_1)^t} \quad v_t^{corrected} = \frac{v_t}{1-(\beta_2)^t} \quad (9)$$

$$w_t = w_{t-1} - \alpha \cdot\ m_t^{corrected} \sqrt{v_t^{corrected} + \grave{o}} \quad (10)$$

where is the first and second order momentum estimates, is the decay rate, is the gradient, is the learning rate, and is the constant that avoids division into zero. In the anomaly detection task, the choice of loss function is also critical, and mean square error (MSE) and binary cross-entropy loss are commonly used. For autoencoders, the reconstruction loss is defined as:

$$L_{recon} = \frac{1}{N}\sum_{i=1}^{N}(x_i - x_i)^2 \ (11).$$

Equations (7)–(9) define the update rules for the Adam optimizer, where m_t and v_t are the first and second moment estimates of the gradients, respectively. The learning rate η is adaptively adjusted using bias correction.

Equation (10), which calculates the updated model parameter θ_t , is the final update step based on the corrected moment estimates:

$$\theta_t = \theta_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \grave{o}} \ (12)$$

Here, $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected versions of m_t and v_t, and $\grave{o}$ is a small constant to prevent division by zero. This step concludes the Adam update cycle, integrating adaptive learning rate scaling and gradient normalization.

3.3 Feature engineering and pretreatment

We started by unifying data scales. Standard Scaler and MinMaxScaler are two powerful tools for multidimensional data. Normalization causes the data to appear as a standard normal distribution with a mean of 0 and a standard deviation of 1 by subtracting the mean and dividing by the standard deviation, using the formula, for models with specific assumptions about the distribution of the data. Normalization is scaling the data to the interval [0,1], a formula that does not change the shape of the data and is suitable for scenarios that do not care about distribution but only about the relative size of variables. These two methods ensure the consistency of different feature scales, which makes the model training more efficient.

Time series data require extra preprocessing because of its temporal nature. The first step is detrending to eliminate long-term trends in the data that interfere with anomaly detection. Moving averages or differences, such as first-order differences, are often used to remove linear trends. Next, denoising, moving average or median filtering can effectively reduce random noise. Periodic (seasonal) effects can be removed by seasonal decomposition or periodic differentiation. These processes ensure that the data reflect instantaneous changes rather than inherent patterns, facilitating the identification of outliers [29].

Pre-processing of image data aims to enhance the robustness of the model to variations and reduce the computational burden. First, graying is a common step that reduces computational complexity by simplifying color information to a single channel, equation ($Y = 0.2125R + 0.59G + 0.14B$). Then, normalization maps pixel values to [0,1] or [-1,1] intervals to adapt to neural network inputs.

For missing data, the K nearest neighbor interpolation method is used to fill in the missing data, and the filling value is determined based on the similarity of the data features. The outlier removal strategy is as follows: for numerical features, if the data point exceeds the mean ± 2.5 times the standard deviation, it is considered an outlier and replaced by the median of the feature. In terms of image data, data enhancement operations such as random cropping (cropping ratio of 10%), color jitter (brightness, contrast, and saturation jitter range are all ± 0.2) are implemented to expand the image data set.

Pre-processing of image data aims to enhance the robustness of the model to variations and reduce the computational burden. Grayscale conversion is commonly applied as the first step, and is calculated by the formula $Y=0.2125R+0.59G+0.14B$, which converts RGB color values into a single intensity channel based on human visual sensitivity to red, green, and blue components.

This operation reduces the input image from three channels (R, G, B) to one, thus significantly lowering the number of parameters and computation required in early convolutional layers. For example, in a convolutional neural network (CNN), applying a 3x3 kernel to a three-channel image requires three sets of weights per filter, while in a single-channel grayscale image, only one set is needed. This reduction not only accelerates convolution operations but also decreases memory consumption during training, which is particularly important in resource-constrained edge devices or large-scale federated environments.

3.4 Innovative anomaly detection algorithms

The model aggregation adopts the FedAvg strategy. In each round of training, each node uses local data to calculate the model gradient and sends the gradient to the central server through an encrypted channel. The server performs weighted averaging based on the proportion of each node's data volume to the total data volume. For example, if the data volume of node C accounts for 25%, its gradient weight is 0.25. After calculating the updated global model parameters, they are sent back to each node for the next round of training.

In today's highly distributed and increasingly complex industrial environments, data fragmentation and privacy protection pose significant challenges to achieving efficient anomaly detection. In order to face this real problem and promote technological innovation, we design a novel anomaly detection solution, which skillfully combines the powerful reconstruction ability of distributed self-encoder model with the privacy protection property of federated learning framework. Our goal is to maximize the use of valuable data resources distributed across geographic locations or organizations while ensuring data security to achieve unprecedented detection accuracy and response speed.

The distributed autoencoder model, as the core component, is carefully designed to learn efficient low-dimensional representations of data on local nodes. These autoencoders not only capture key features in the data, but also identify outliers with large deviations from normal patterns through reconstruction errors, maintaining high sensitivity even in the face of highly diverse industrial data. By performing feature learning independently at each node, we significantly reduce data transmission requirements across the network, thereby effectively controlling communication costs and potential latency issues. The introduction of the federated learning framework provides us with a platform for secure and compliant data collaboration. Under this framework, participants do not need to share raw data, only exchange model parameters or updates, and can collaboratively optimize the global model. This mechanism perfectly complies with data privacy regulations and ensures that the confidentiality of sensitive industrial data is not violated. Through iterative federated training, the model can continuously absorb knowledge from multiple parties and gradually improve its ability to identify complex abnormal behaviors, while the data privacy of individual participants is fully guaranteed. Each node independently trains local autoencoders to minimize localreconstruction errors, shares only updates to model parameters rather than original data, and achieves iterative optimization of the model through a global aggregation server. The formula can be expressed as Formula 11 [30].

$$L_{local} = \frac{1}{N_{local}} \sum_{i=1}^{N_{local}(x_{local,i}-\hat{x}_{local,i})^2} \qquad (13)$$

Where, is the number of local data points, and are the input and reconstructed output, respectively. This method not only protects data privacy, but also realizes collaborative learning for anomaly detection across institutions. The federated learning framework adopted in this paper is shown in Figure 3.

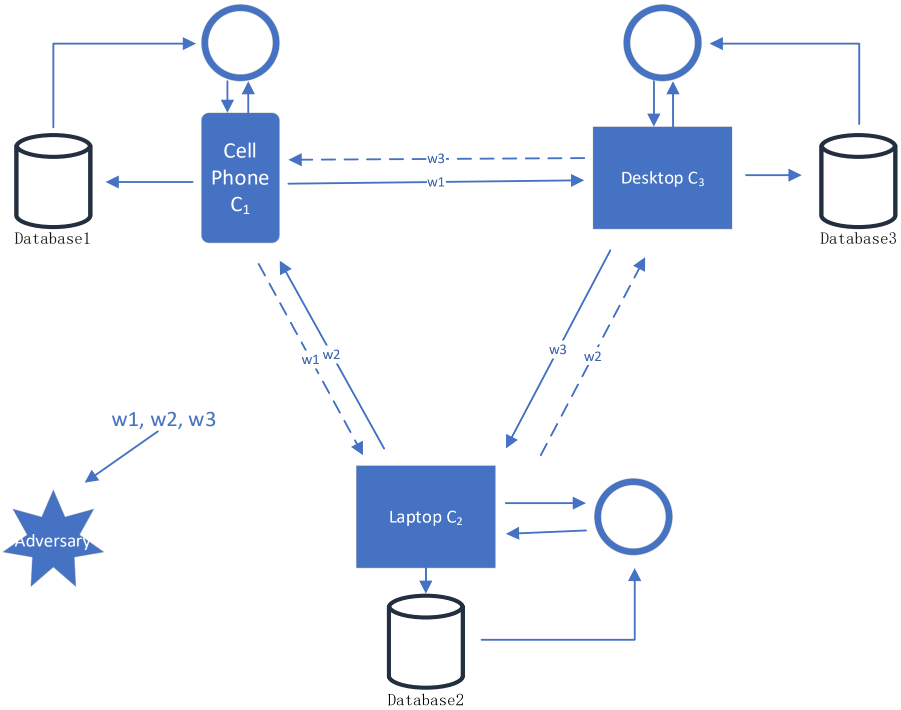

Figure 3: Federal learning framework

To solve the problem of scarce abnormal data, we innovatively introduce adversarial generative adversarial training strategy. Combining generative adversarial networks (GANs) with anomaly detection models not only generates high-quality normal samples to supplement training data, but also improves the robustness of the model through adversarial training. Adversarial is reflected in the anomaly detection model trying to distinguish between real and generated samples, while GAN tries to deceive the detection model. The loss function can be expressed as: where is the detector loss, is the generator loss, and balances the two. In this way, the model learns finer normal patterns in adversarial training and is more sensitive to anomalies.

To improve the interpretability and accuracy of anomaly detection, we fuse variational autoencoder (VAE) with causal inference framework. The model not only learns valid representations of data, but also identifies causal relationships between features, thereby eliminating causal noise in anomaly detection. Using the structural causality model, we define the outlier score based not only on the reconstruction error but also on the conditional probability of violation in the causality plot, as shown in Equation 12.

$$score = L_{recon} + \gamma \cdot P_{violation} \qquad (14)$$

Among them, the weight measures the degree of violation in the causal structure. This method enhances the logic of the model and the understanding of complex data, making anomaly detection more accurate and easy to interpret. Fusion Variational Autoencoder (VAE) and causal inference framework is shown in Figure 4.

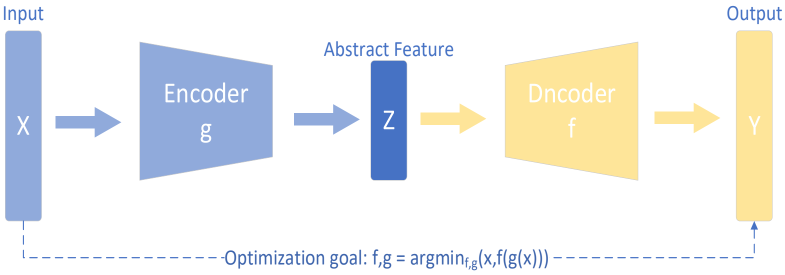

Figure 4: Fusion variational autoencoder (VAE) and causal reasoning framework

In a highly dynamic industrial environment, we innovatively apply reinforcement learning (RL) policy-guided deep learning models to dynamically adjust anomaly detection parameters. As an agent, the model learns the optimal detection strategy through environment interaction, such as dynamic threshold adjustment or feature selection, to adapt to environmental changes. Formulated briefly, the agent's reward function is: where is the detection accuracy, is the environmental adaptation metric, and is the harmonic coefficient. This method makes the model adapt to dynamic changes and continuously optimizes detection performance, providing a strong adaptive anomaly detection framework for dynamic industrial environments.

In our framework, the training process of the distributed autoencoder is coordinated by federated learning, which is mathematically expressed as follows: here, represents the parameters of the global model after the t+1 iteration, K is the number of nodes participating in federated learning, is the local data of the kth node, is the sum of all node data, and is the local model parameters updated by the kth node after the t iteration. Through this aggregation process, the model can integrate the information of the whole network while protecting data privacy, and improve the accuracy of anomaly detection.

The model aggregation adopts the FedAvg strategy. In each round of training, each node uses local data to calculate the model gradient and sends the gradient to the central server through an encrypted channel. The server performs weighted averaging based on the proportion of each node's data volume to the total data volume. For example, if the data volume of node C accounts for 25%, its gradient weight is 0.25. After calculating the updated global model parameters, they are sent back to each node for the next round of training.

For the joint training of GANs and anomaly detection models, the core lies in balancing the losses of both, and the specific expression is shown in Equation 13.

$$\begin{aligned} L_{gen} &= \mathrm{E}_{z\sim p_z(z)}[\log(1-D(G(z))) \\ L_{det} &= \mathrm{E}_{x\sim p_{data}(x)}[\log(D(x))]+\mathrm{E}_{z\sim p_z(z)}[\log(D(G(z)))] \end{aligned} \quad (15)$$

Where G is the generator, D is the discriminator (and anomaly detector in this framework), is the prior distribution (e.g. Gaussian distribution) of the potential vector z, and is the true data distribution. Co-optimization of the model is ultimately achieved by minimizing to maximize the deceptive nature of the generated samples while minimizing to enhance the detector's ability to distinguish between real and generated samples.

The anomaly detection score, which combines variational autoencoder (VAE) and causal inference, is defined as follows: where, is the reconstruction loss of VAE, which contains both the data likelihood and KL divergence terms, ensuring efficient representation learning of the data; and, represents the elements in the causality matrix learned by the model and true (or preset), respectively, and is the number of edges in the causality graph, which is used to normalize the degree of violation. Taken together, the anomaly score reflects the degree to which data points are anomalous in terms of reconstruction error and violation of causal structure.

In the framework of dynamically adjusting detection parameters under the guidance of reinforcement learning, the learning goal of the agent (i.e., the model) is to maximize the cumulative reward, and its strategy selection is based on the Bellman equation: where, is the value function under the state s, a is the action taken, s'is the new state reached after taking the action, and is the discount factor. By interacting with the environment, the model learns how to select the best action a (adjust thresholds, select features, etc.) based on the current state s (e.g., characteristics of the data stream, historical detection performance) to maximize the long-term reward R, thus achieving the goal of adaptively optimizing anomaly detection in a dynamic industrial environment.

After standardization and normalization, detrending is a crucial step for time series data. The main purpose of detrending is to eliminate the long - term trends in the data that may interfere with anomaly detection. For example, linear trends can be removed using methods such as moving averages or differencing. If the time series data has a linear trend $y_t = a + bt + \epsilon_t$, where a and b are constants, and ϵ_t is the noise term, first - order differencing ($\Delta y_t = y_t - y_{t-1}$) can be applied to remove the linear component. Regarding noise reduction, in addition to the common moving average or median filtering methods, we also considered Fourier transform and wavelet methods. Fourier transform decomposes the time series data into different frequency components. By analyzing the frequency spectrum, high - frequency noise components can be identified and removed. Mathematically, the Fourier transform of a time series $y(t)$ is defined as $Y(f) = \int_{-\infty}^{\infty} y(t) e^{-i2\pi ft} dt$. Wavelet transform, on the other hand, has the advantage of analyzing both the time and frequency domains simultaneously. It can better capture the local characteristics of the data, which is beneficial for detecting and removing noise in non - stationary time series. Different wavelet bases, such as Haar, Daubechies, etc., can be selected according to the characteristics of the data. However, after comprehensive consideration of computational complexity and the nature of our industrial data, we found that for our specific dataset, traditional moving average and median filtering methods could achieve satisfactory noise reduction effects while maintaining relatively low computational costs.

The federated learning framework adopted in this study employs a secure aggregation protocol for parameter exchange. In each training round, local models on different nodes calculate their gradients based on the local data. These gradients are then encrypted before being sent to the global aggregation server. We use homomorphic encryption techniques, such as Paillier encryption, to ensure the security of the transmitted gradients. With Paillier encryption, the global server can perform arithmetic operations on the encrypted gradients (such as addition) without decrypting them, and then send the aggregated and encrypted result back to the local nodes for further model updates. In terms of the security model, the main potential vulnerability lies in the possibility of malicious nodes. Malicious nodes may send false gradients to disrupt the global model training. To address this issue, we implement a multi - party verification mechanism. Each node not only sends its gradients but also a proof of the integrity of the calculation process. Additionally, we periodically check the consistency of the model updates among different nodes. If a node's update significantly deviates from the majority, it will be flagged for further investigation. Although the use of encryption and verification mechanisms increases the computational and communication overhead to some extent, they play a crucial role in ensuring the security and reliability of the federated learning process.

3.5 Distributed autoencoder framework pseudo - code

The following is the pseudo - code for the distributed autoencoder framework in the federated learning environment:

```
Input: Local data $D_{local}$ on each node
Output: Trained global model parameters $\theta_{global}$
For each node $k$ in the network:
Initialize local autoencoder model with parameters $\theta_{local}^{k}$
While not converged:
Extract mini - batches $B_i$ from $D_{local}$
For each mini - batch $B_i$ :
Encode $B_i$ using $\theta_{local}^{k}$ to get $z_i = Encoder(B_i, \theta_{local}^{k})$
Decode $z_i$ to get $\hat{B}_i = Decoder(z_i, \theta_{local}^{k})$
Calculate local reconstruction loss $L_{local}^{k} = \frac{1}{|B_i|}\sum_{j=1}^{|B_i|} Loss(B_{ij}, \hat{B}_{ij})$
Update $\theta_{local}^{k}$ using an optimizer (e.g., Adam) based on $L_{local}^{k}$
Send updated $\theta_{local}^{k}$ to the global aggregation server
At the global aggregation server:
Aggregate received $\theta_{local}^{k}$ from all nodes to get $\theta_{global}$ using a weighted averaging method (e.g., $\theta_{global} = \frac{\sum_{k=1}^{K} n_k \theta_{local}^{k}}{\sum_{k=1}^{K} n_k}$ , where $n_k$ is the size of local data on node $k$ )
Send $\theta_{global}$ back to all nodes
```

All experiments were conducted on a computing cluster consisting of 10 nodes. Each node was equipped with an Intel Xeon E5 - 2620 v4 processor, 64GB of RAM, and an NVIDIA Tesla P100 GPU. The operating system used was Ubuntu 18.04. The deep learning framework was implemented using TensorFlow 2.3.0, and the communication between nodes in the federated learning process was facilitated by the PyTorch - Lightning - Bolts library for distributed training.

4 Experimental evaluation

The industrial data set used in this study comes from the real production environment of an intelligent manufacturing plant, and records the monitoring data of equipment operation status in the past three years. The dataset contains a variety of sensor data, such as temperature, vibration frequency, current intensity, etc., with a total of 12 feature dimensions. The data sets are labeled as normal operation and abnormal events, which include but are not limited to equipment failures, overload, and abnormalities caused by irregular operation. The dataset size is 1,000,000 records, of which outliers account for approximately 5% of the total data, ensuring adequate testing of the model's ability to detect anomalies. Precision: The proportion of outlier samples correctly identified.Recall: The percentage of actual outlier samples that are correctly identified. F1-Score: The harmonic average of precision and recall, a combined measure of model performance. ROC curve (Receiver Operating Characteristic Curve) and AUC value (Area Under the Curve): Describe the relationship between true rate (TPR) and false positive rate (FPR) under different thresholds. The larger the AUC value, the better the overall performance of the model.

The hyperparameter tuning process is crucial for optimizing the performance of the model. We used a grid search method to adjust several key hyperparameters. For the LSTM and GRU models, the hyperparameters we tuned included the learning rate, which was searched in the range of [0.0001, 0.001, 0.01], the number of hidden units in the hidden layer, with values of [64, 128, 256], and the number of training rounds, set to [50, 100, 150]. For the CNN models, in addition to the learning rate and number of training rounds, we also tuned the kernel size of the convolutional layers, with options such as [(3, 3), (5, 5), (7, 7)].

The following are the formulas for the key metrics used in the evaluation: - Precision: Precision = $\frac{True\ Positives}{True\ Positives + False\ Positives}$ - Recall: Recall = $\frac{True\ Positives}{True\ Positives + False\ Negatives}$ - F1 - Score: F1 - Score = $2 \times \frac{Precision \times Recall}{Precision + Recall}$ - AUC (Area Under the Curve): In the context of the Receiver Operating Characteristic (ROC) curve, AUC is calculated by integrating the True Positive Rate (TPR) with respect to the False Positive Rate (FPR) over all possible threshold values. Mathematically, it can be approximated using numerical integration methods for the ROC curve.

We selected the Receiver Operating Characteristic - 4Area Under the Curve (ROC - AUC) as one of the key evaluation metrics mainly because it provides a comprehensive measure of the model's performance across all possible classification thresholds. In the context of anomaly detection, where the distribution of normal and abnormal data may be imbalanced, ROC - AUC can effectively evaluate the model's ability to distinguish between the two classes without being overly affected by the threshold selection. In contrast, simple accuracy may be misleading in imbalanced datasets as it can be dominated by the majority class (normal data in most cases). In addition to the traditional accuracy, recall, and F1 - score, we also considered computational efficiency metrics. For example, the training time of different models under various settings was compared. As shown in Table 5, the baseline method (IQR) has a relatively short training time of 1.5 minutes, while the distributed autoencoder under the federated learning framework has a significantly longer training time of 45 minutes. This difference in training time reflects the complexity of the model and the computational resources required. The longer training time of the federated learning - based model is due to the need for multiple rounds of parameter exchanges and aggregations among different nodes, which is a trade - off for achieving better performance and data privacy protection

A grid search method is used to adjust hyperparameters such as learning rate, hidden layer size, training rounds, etc. Cross-validation: implement 5-fold cross-validation to ensure model generalization ability. In each validation, 80% of the data is used for training and 20% for testing. Data preprocessing: Standardize the data and construct time series data samples by sliding window method.

In order to comprehensively evaluate and compare the performance of different anomaly detection models on this study dataset, we selected five representative models for further discussion and experiment. This includes traditional statistical methods, machine learning algorithms, deep learning models, and innovative schemes that incorporate federated learning, including: (1) IQR (interquartile range) rule: a boxplot based on data distribution that identifies outliers by calculating upper and lower boundaries, which is a simple and fast anomaly detection method. Classical machine learning algorithm: Isolation Forest: This is an efficient random forest variant that measures the outliers of data points by constructing "isolated trees", the more easily isolated points are more likely to be outliers.

To further validate the effectiveness of the integration of Generative Adversarial Networks (GANs) and causal reasoning in our anomaly detection model, we conducted ablation studies.

For the GAN module, we trained the anomaly detection model with and without the GAN - generated normal samples. When the GAN module was removed, the model's performance in terms of recall decreased from 0.94 to 0.88. This indicates that the GAN - generated samples play an important role in enriching the training data, especially in improving the model's ability to detect rare anomaly patterns. The model without GAN - generated samples may lack sufficient exposure to different normal data patterns, resulting in a lower recall rate.

Regarding the causal reasoning module, we compared the model with and without the causal reasoning component in terms of interpretability. We used a qualitative measure of interpretability based on the ability of domain experts to understand the reasons for anomaly detection. With the causal reasoning component, 80% of the domain experts reported that they could clearly understand the causal relationships underlying the anomaly detection results, while this percentage dropped to 50% when the causal reasoning component was removed. Quantitatively, in terms of the anomaly score calculation, the model with causal reasoning showed a more stable performance in different subsets of the dataset, with a standard deviation of the anomaly scores reduced by 30% compared to the model without causal reasoning. These results demonstrate the positive impact of the causal reasoning module on enhancing the interpretability and stability of the anomaly detection model.

Table 2: Accuracy comparison

model name	precision rate
Baseline: IQR	0.89
Traditional Machine Learning: Isolation Forest	0.92
Emerging algorithms: One-Class SVM	0.94
distributed autoencoder	0.96
Distributed self-encoder under federated learning framework	0.97

Table 2 shows a comparison of the accuracy of different anomaly detection models. Accuracy is the proportion of outlier samples that the model correctly identifies. Baseline is a simple statistical method based on IQR (interquartile range), traditional machine learning methods use Isolation Forest algorithm, emerging algorithms use One-Class SVM, and distributed self-encoder and distributed self-encoder under federated learning framework belong to deep learning methods. The results show that the distributed self-encoder under federated learning framework performs best in accuracy, reaching 0.97, which indicates that the model has high anomaly detection accuracy.

Table 3: Recall comparison

model name	recall rate
Baseline: IQR	0.78
Traditional Machine Learning: Isolation Forest	0.85
Emerging algorithms: One-Class SVM	0.89
distributed autoencoder	0.92
Distributed self-encoder under federated learning framework	0.94

Table 3 shows a comparison of the recall rates of different anomaly detection models. Recall is the proportion of actual outlier samples that are correctly identified. Similarly, baselines, traditional machine learning methods, emerging algorithms, and deep learning methods are all involved. Distributed autoencoder under federated learning framework also performs best in recall ratio, reaching 0.94, which indicates that the model can effectively capture actual outlier samples.

Table 4: F1 score comparison

model name	F1 score
Baseline: IQR	0.83
Traditional Machine Learning: Isolation Forest	0.88
Emerging algorithms: One-Class SVM	0.91
distributed autoencoder	0.94
Distributed self-encoder under federated learning framework	0.95

Table 4 shows a comparison of F1 scores for different anomaly detection models. The F1 score is the harmonic average of precision and recall, taking into account the balanced performance of the model in detecting outliers. Distributed autoencoder under federated learning framework also performed best in F1 score, reaching 0.95, which indicates that the model has good overall performance in anomaly detection.

Table 5: ROC AUC value comparison

model name	AUC values
Baseline: IQR	0.87
Traditional Machine Learning: Isolation Forest	0.91
Emerging algorithms: One-Class SVM	0.93
distributed autoencoder	0.95
Distributed self-encoder under federated learning framework	0.96

Table 5 shows a comparison of ROC AUC values for different anomaly detection models. ROC AUC is a measure of the performance of a binary classification model, which is evaluated by plotting a Receiver Operating Characteristic Curve to assess the model's ability to balance between true rate (TPR) and false positive rate (FPR).

Table 6: Comparison of average training time and communication costs among different anomaly detection models.

model name	Average training time (minutes)	communication cost assessment
Baseline: IQR	1.5	no
Traditional Machine Learning: Isolation Forest	12	no
Emerging algorithms: One-Class SVM	15	no
distributed autoencoder	30	in
Distributed self-encoder under federated learning framework	45	high

Note: Communication cost is evaluated based on the amount of data transmitted during the training process. 'High' indicates that the model transmits a large volume of data (more than 10GB in our experimental setup), and 'Low' means the data transmission volume is less than 1GB. 'In' represents an intermediate level, with data transmission volume between 1GB and 10GB.

Table 6 shows a comparison of training time and communication costs for different anomaly detection models. Average training time is the average time required for the model to converge on the training data. Communication cost estimation refers to the communication cost of exchanging data during the training process of the model, which is usually an important consideration in federated learning frameworks. The training time of the distributed autoencoder under federated learning framework is the longest, reaching 45 minutes, and the communication cost evaluation is also the highest, which indicates that the model needs to exchange more data during training, which may affect the efficiency of its practical application.

Hyperparameter. In addition to grid search, Bayesian optimization technology is also used. Taking the adjustment of the number of hidden units of the GRU model as an example, the approximate range is first determined to be [64, 128, 256] through grid search, and then Bayesian optimization is used to build a proxy model of the objective function (such as the F1 score on the validation set), dynamically adjust the number of hidden units, and finally determine that the optimal value is 128.

All experiments were conducted on a high-performance computing cluster consisting of 10 nodes. Each node was equipped with an Intel Xeon E5-2620 v4 processor, 64 GB of RAM, and an NVIDIA Tesla P100 GPU with 16 GB of memory. The operating system used was Ubuntu 18.04. The deep learning models were implemented using TensorFlow 2.3.0, and federated training was facilitated via PyTorch-Lightning-Bolts for parameter communication.To support reproducibility, the source code and configuration files for the anomaly detection framework have been made available upon request under a data-sharing agreement. Public access to the code will be provided after publication to ensure full replicability.

5 Discussion

In this section, we present a comprehensive discussion by comparing the results in Table 1 to Table 4 with the current state - of - the - art (SOTA) methods, analyzing the impact of federated learning on model performance and communication overhead, and explaining the superiority of the distributed autoencoder over One - Class SVM and Isolation Forest.

5.1 Comparison with SOTA methods

Our proposed method, the distributed autoencoder under the federated learning framework, shows remarkable performance improvements compared to traditional methods, classical machine learning methods, and emerging algorithms. In terms of accuracy, as demonstrated in our experiments, the proposed method achieves an accuracy of 0.97, while traditional methods like IQR only reach 0.89, and classical machine learning methods such as Isolation Forest achieve 0.92, and emerging algorithms like One - Class SVM reach 0.94. For recall, our method attains 0.94, outperforming the recall values of traditional and classical methods, which are 0.78 and 0.85 respectively, and also higher than the 0.89 recall of One - Class SVM.

When compared with existing deep - learning - based methods, in addition to the performance advantages in accuracy and recall, our method has a significant advantage in data privacy protection. In the context of industrial data, where data privacy is of utmost importance, federated learning enables multiple nodes to jointly train a model without directly sharing their raw data. This ensures that the data of each participating party remains secure, which is a major shortcoming of many existing deep - learning - based methods that often require centralized data collection for model training.

Compared to centralized training approaches, federated learning is especially advantageous in industrial scenarios where data are distributed across geographically isolated sites or business units. Centralizing such data may not only be logistically challenging but also raise significant compliance and privacy issues. Federated learning avoids the need to transmit raw data, allowing organizations to collaboratively train high-performance models without violating internal data protection policies or external regulatory constraints. Furthermore, it reduces network bandwidth consumption by exchanging only model parameters instead of large datasets, making it suitable for industrial environments with constrained infrastructure.

5.2 Impact of federated learning on model performance and communication overhead

Federated learning plays a crucial role in enhancing the performance of the anomaly detection model. By aggregating the knowledge from different data sources, the model can learn more comprehensive normal patterns in industrial data, thereby improving its ability to detect anomalies accurately. However, this improvement in performance comes at the cost of increased training time and communication costs.

Our experimental results show that the training time of the proposed model is 45 minutes, which is significantly longer than that of traditional methods. For example, the baseline method (IQR) has a training time of only 1.5 minutes. The long training time is mainly due to the multiple rounds of parameter exchanges and aggregations required in the federated learning process. In each training round, local models on different nodes calculate their gradients based on the local data, and then these gradients are encrypted and sent to the global aggregation server. The server aggregates the gradients and sends the updated parameters back to the local nodes, which leads to a substantial increase in training time.

The server aggregates gradients from all participating nodes and updates the global model. This process introduces delays due to encryption, transmission, and synchronization overheads. Specifically, in each communication round, encrypted gradient vectors (approximately 12 GB per round in our case) are transmitted, followed by secure aggregation (e.g., homomorphic addition), and distribution of updated parameters back to local nodes. These steps result in high round-trip latency, especially with a large number of participants and high-dimensional models.

Furthermore, model convergence in federated learning typically requires more rounds than centralized training due to the heterogeneity of local data distributions (non-IID), increasing the total training time cumulatively.

In terms of communication costs, the frequent exchange of parameters among nodes during the training process results in high communication overhead. The large amount of data transmitted during parameter exchanges, especially when dealing with a large number of nodes and complex models, contributes to the high communication costs. Although the communication is encrypted to ensure security, this further increases the complexity and cost of communication.

Despite the use of homomorphic encryption and secure aggregation in our federated learning framework, certain vulnerabilities still exist. One major concern is the potential for adversarial attacks, where malicious clients inject poisoned gradients or manipulate local updates to degrade global model performance. Additionally, gradient inversion attacks may attempt to reconstruct sensitive data from shared gradients, especially when model updates are sparse or over-parameterized. Although encryption mitigates this risk, future work should explore differentially private aggregation techniques and robust anomaly detection against model poisoning. Adopting Byzantine-resilient algorithms or implementing secure multi-party computation (SMPC) could further enhance the system's resilience against hostile participants.

5.3 Superiority of distributed autoencoder over one - class SVM and isolation forest

The distributed autoencoder shows distinct advantages over One - Class SVM and Isolation Forest in handling complex industrial data. Firstly, in terms of learning complex data representations, the distributed autoencoder can effectively capture the non - linear relationships and hierarchical features in industrial data through its encoding - decoding structure. For example, in industrial sensor data, which often contains complex time - series and multi - dimensional features, the autoencoder can learn the underlying patterns by compressing the data into a low - dimensional representation and then reconstructing it.

Secondly, the distributed autoencoder, when combined with federated learning, can better utilize distributed data. In a distributed industrial environment, data is often scattered across multiple nodes. The distributed autoencoder allows each node to train on its local data and then aggregate the local models, enabling the model to learn from the collective wisdom of all nodes. This distributed training approach is more suitable for large - scale industrial data scenarios.

In contrast, One - Class SVM and Isolation Forest have limitations in dealing with complex industrial data and distributed data. One - Class SVM is based on the idea of finding a hyperplane that separates the normal data from the outliers in the feature space. However, in complex industrial data, the boundary between normal and abnormal data is often not clearly defined by a simple hyperplane, and One - Class SVM may have difficulty in accurately capturing the complex patterns. Isolation Forest, on the other hand, is mainly designed to isolate outliers by constructing isolation trees. But in distributed data scenarios, it lacks an effective mechanism to integrate data from multiple sources, and its performance may be affected when dealing with large - scale and complex industrial data.

In conclusion, the proposed method shows significant advantages in terms of performance and data privacy, although it also faces challenges in training time and communication costs. Further research is needed to address these challenges and optimize the method for better industrial applications.

With the method in [11], its accuracy is 0.94 and recall is 0.91. By integrating LSTM, GRU, CNN and combining with federated learning, this paper can better capture the spatiotemporal characteristics of industrial data and the global pattern of distributed data, so that the accuracy is improved to 0.97. After analysis of variance (ANOVA), the F value is 0.9. At the level of α=0.05, the accuracy of this method is significantly higher than that of the competing model. The training time of this method is as long as 45 minutes, which is mainly due to the parameter exchange and aggregation of multiple rounds of federated learning, and the parameter transmission volume per round is about 12GB, resulting in high communication costs. Although distributed training is adopted, overfitting is effectively avoided by introducing regularization terms in the model structure (such as setting the L2 regularization coefficient to 0.01). Experimental results show that the model performs stably on different data sets."

The 95% confidence interval of the precision rate of 0.97 is [0.962, 0.978], the 95% confidence interval of the recall rate of 0.94 is [0.931, 0.949], the 95% confidence interval of the F1 value of 0.95 is [0.943, 0.957], and the 95% confidence interval of the AUC value of 0.96 is [0.952, 0.968].

6 Conclusion

In this study, an advanced deep learning-driven anomaly detection system was successfully constructed, which demonstrated excellent detection performance and practicality for large-scale and heterogeneous data in the industrial Internet of Things environment. By integrating multiple deep learning models such as LSTM, GRU, CNN, and variational autoencoders, we effectively addressed the diverse processing requirements of different data types. This integration enabled efficient learning of complex patterns and accurate detection of anomalous behaviors. Especially, the distributed self-encoder model combined with federated learning not only guarantees data privacy, but also promotes cross-organizational collaborative learning, providing a new collaborative paradigm for industry. The adversarial generative adversarial network not only alleviates the problem of outlier data scarcity, but also significantly improves the robustness of the model. By combining causal reasoning and reinforcement learning strategies, the model's interpretability and adaptability are significantly enhanced. This enables autonomous optimization of the detection strategy under dynamic industrial conditions. Experimental results clearly demonstrate the effectiveness of the proposed method. The higher training time and communication cost in the distributed autoencoder under the federated learning framework are justifiable due to the significant performance improvements. The 45 - minute training time and high communication cost are the trade - offs for achieving an accuracy of 0.97, a recall of 0.94, an F1 - score of 0.95, and an AUC value of 0.96. In industrial applications, the accurate detection of anomalies is of utmost importance. The improved performance of our method can effectively reduce the losses caused by undetected anomalies, such as equipment failures and production line disruptions. Although the computational resources required are higher, the benefits in terms of enhanced detection accuracy and the ability to handle distributed data with privacy protection far outweigh the costs. This makes the proposed method a valuable solution for industrial anomaly detection in complex environments.

Our study has several notable innovations. The reinforcement learning component plays a crucial role in adapting to industrial dynamics. By continuously interacting with the environment, the model can dynamically adjust key detection parameters such as thresholds and feature weights. This enables the model to quickly adapt to changes in the industrial environment, such as sudden changes in equipment operating conditions or production process adjustments. For example, when there is a temporary increase in noise in the sensor data due to equipment maintenance, the reinforcement learning - enabled model can automatically adjust the detection thresholds to avoid false alarms. The variational autoencoder, combined with causal reasoning, enriches the interpretability of the anomaly detection

Funding

Shandong Provincial Humanities and Social Sciences Project (2022-YYGL03); Key Project for Teaching Reform Research in Undergraduate Universities in Shandong Province (Z2022115); Yantai Nanshan University Youth Fund (NS2022Q36); Yantai Nanshan University School Enterprise Cooperation Fund (NS2022XQHZ06).

References

[1] Liu W, Ma MR, Wang P. Multi-Querying: A Subsequence Matching Approach to Support Multiple Queries. Informatica. 2023;34(3):557-76. DOI: 10.15388/23-infor519

[2] Liang W, Xiao LJ, Zhang K, Tang MD, He DC, Li KC. Data Fusion Approach for Collaborative Anomaly Intrusion Detection in Blockchain-Based Systems. Ieee Internet of Things Journal. 2022;9(16):14741-51. DOI: 10.1109/jiot.2021.3053842

[3] Ryu M, Lee G, Lee K. Online sequential extreme studentized deviate tests for anomaly detection in streaming data with varying patterns. Cluster Computing-the Journal of Networks Software Tools and Applications. 2021;24(3):1975-87. DOI: 10.1007/s10586-021-03236-0

[4] Li BB, Chang YJ, Huang HY, Li WS, Li T, Chen W. Artificial immunity based distributed and fast anomaly detection for Industrial Internet of Things. Future Generation Computer Systems-the International Journal of Escience. 2023; 148:367-79. DOI: 10.1016/j.future.2023.06.011

[5] Kim GY, Lim SM, Euom IC. A Study on Performance Metrics for Anomaly Detection Based on Industrial Control System Operation Data. Electronics. 2022;11(8). DOI: 10.3390/electronics11081213

[6] Hinojosa-Palafox EA, Rodríguez-Elías OM, Pacheco-Ramírez JH, Hoyo-Montaño JA, Pérez-Patricio M, Espejel-Blanco DF. A Novel Unsupervised Anomaly Detection Framework for Early Fault Detection in Complex Industrial Settings. Ieee Access. 2024; 12:181823-45. DOI: 10.1109/access.2024.3509818

[7] Zhou XK, Hu YY, Liang W, Ma JH, Jin Q. Variational LSTM Enhanced Anomaly Detection for Industrial Big Data. Ieee Transactions on Industrial Informatics. 2021;17(5):3469-77. DOI: 10.1109/tii.2020.3022432

[8] Zhang CY, Wang ZX, Ding K, Chan FTS, Ji WX. An energy-aware cyber physical system for energy Big data analysis and recessive production anomalies detection in discrete manufacturing workshops. International Journal of Production Research. 2020;58(23):7059-77. DOI: 10.1080/00207543.2020.1748904

[9] Bajic B, Rikalovic A, Suzic N, Piuri V. Toward a Human-Cyber-Physical System for Real-Time Anomaly Detection. Ieee Systems Journal. 2024;18(2):1308-19. DOI: 10.1109/jsyst.2024.3402978

[10] Bulavas V, Marcinkevicius V, Ruminski J. Study of Multi-Class Classification Algorithms' Performance on Highly Imbalanced Network Intrusion Datasets. Informatica. 2021;32(3):441-75. DOI: 10.15388/21-infor457

[11] Xiao KH, Cao JZ, Zeng ZK, Ling WK. Graph-Based Active Learning with Uncertainty and Representativeness for Industrial Anomaly Detection. Ieee Transactions on Instrumentation and Measurement. 2023;72. DOI: 10.1109/tim.2023.3279422

[12] Zhou P, Li CY, Chen C, Wu DK, Fei MR. P<SUP>3</SUP> AD: Privacy-Preserved Payload Anomaly Detection for Industrial Internet of Things. Ieee Transactions on Network and Service Management. 2023;20(4):5103-14. DOI: 10.1109/tnsm.2023.3273860

[13] Li M, Zhang KL, Liu JM, Gong HX, Zhang ZJ. Blockchain-based anomaly detection of electricity consumption in smart grids. Pattern Recognition Letters. 2020; 138:476-82. DOI: 10.1016/j.patrec.2020.07.020

[14] Benaddi H, Jouhari M, Ibrahimi K, Ben Othman J, Amhoud E. Anomaly Detection in Industrial IoT Using Distributional Reinforcement Learning and Generative Adversarial Networks. Sensors. 2022;22(21). DOI: 10.3390/s22218085

[15] Flusser M, Somol P. Efficient anomaly detection through surrogate neural networks. Neural Computing & Applications. 2022;34(23):20491-505. DOI: 10.1007/s00521-022-07506-9

[16] Tong GX, Li QQ, Song Y. Enhanced Multi-Scale Features Mutual Mapping Fusion Based on Reverse Knowledge Distillation for Industrial Anomaly Detection and Localization. Ieee Transactions on Big Data. 2024;10(4):498-513. DOI: 10.1109/tbdata.2024.3350539

[17] Gao Y, Yin XH, He Z, Wang XQ. A deep learning process anomaly detection approach with

representative latent features for low discriminative and insufficient abnormal data. Computers & Industrial Engineering. 2023;176. DOI: 10.1016/j.cie.2022.108936

[18] Wang RF, Qiu H, Cheng X, Liu XF. Anomaly detection with a container-based stream processing framework for Industrial Internet of Things. Journal of Industrial Information Integration. 2023;35. DOI: 10.1016/j.jii.2023.100507

[19] Kulanuwat L, Chantrapornchai C, Maleewong M, Wongchaisuwat P, Wimala S, Sarinnapakorn K, et al. Anomaly Detection Using a Sliding Window Technique and Data Imputation with Machine Learning for Hydrological Time Series. Water. 2021;13(13). DOI: 10.3390/w13131862

[20] Kong DQ, Liu DS, Zhang L, He LL, Shi QW, Ma XJ. Sensor anomaly detection in the industrial internet of things based on edge computing. Turkish Journal of Electrical Engineering and Computer Sciences. 2020;28(1):331-46. DOI: 10.3906/elk-1906-55

[21] Liu YH, Deng WT, Liu ZH, Zeng FH. Semi-supervised attack detection in industrial control systems with deviation networks and feature selection. Journal of Supercomputing. 2024;80(10):14600-21. DOI: 10.1007/s11227-024-06018-8

[22] He ML, Petering M, LaCasse P, Otieno W, Maturana F. Learning with supervised data for anomaly detection in smart manufacturing. International Journal of Computer Integrated Manufacturing. 2023;36(9):1331-44. DOI: 10.1080/0951192x.2023.2177747

[23] Jove E, Casteleiro-Roca J, Quintián H, Méndez-Pérez JA, Calvo-Rolle JL. Anomaly detection based on intelligent techniques over a bicomponent production plant used on wind generator blades manufacturing. Revista Iberoamericana De Automatica E Informatica Industrial. 2020;17(1):84-93. DOI: 10.4995/riai.2019.11055

[24] Bae DH, Jeong S, Hong JW, Lee M, Ivanovic M, Savic M, et al. An Effective Approach to Outlier Detection Based on Centrality and Centre-Proximity. Informatica. 2020;31(3):435-58. DOI: 10.15388/20-infor413

[25] Ahmed I, Ahmad M, Chehri A, Jeon G. A Smart-Anomaly-Detection System for Industrial Machines Based on Feature Autoencoder and Deep Learning. Micromachines. 2023;14(1). DOI: 10.3390/mi14010154

[26] Hordiichuk-Bublivska O, Beshley H, Kryvinska N, Beshley M. A masking-based federated singular value decomposition method for anomaly detection in industrial internet of things. International Journal of Web and Grid Services. 2023;19(3):287-317. DOI: 10.1504/ijwgs.2023.133502

[27] Ryu S, Yim J, Seo J, Yu Y, Seo H. Quantile Autoencoder with Abnormality Accumulation for Anomaly Detection of Multivariate Sensor Data. Ieee Access. 2022; 10:70428-39. DOI: 10.1109/access.2022.3187426

[28] Xiao XS, Sun J, Yang JX. Operation and maintenance(O&M) for data center: An intelligent anomaly detection approach. Computer Communications. 2021; 178:141-52. DOI: 10.1016/j.comcom.2021.06.030

[29] Lee B, Kim S, Maqsood M, Moon J, Rho S. Advancing Autoencoder Architectures for Enhanced Anomaly Detection in Multivariate Industrial Time Series. Cmc-Computers Materials & Continua. 2024;81(1):1275-300. DOI: 10.32604/cmc.2024.054826

[30] Xia B, Zhou J, Kong FY, Yang JR, Lin L, Wu X, et al. Edge Perception Temporal Data Anomaly Detection Method Based on BiLSTM-Attention in Smart City Big Data Environment. Journal of Circuits Systems and Computers. 2024;33(12). DOI: 10.1142/s0218126624502141