Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 2668761, 10 pages
https://doi.org/10.1155/2021/2668761

Research Article

LstFcFedLear: A LSTM-FC with Vertical Federated Learning Network for Fault Prediction

Xiangquan Zhang,[1] **Zhili Ma**,[1] **Anmin Wang**,[2] **Haifeng Mi**,[2] **and Junjun Hang**[3]

[1]*State Grid Gansu Electric Power Company, Lanzhou 730030, China*
[2]*State Grid Baiyin Power Supply Company, Baiyin 730900, China*
[3]*Huainan Normal University, Huainan 232001, China*

Correspondence should be addressed to Xiangquan Zhang; nhuang1111@163.com

Received 18 May 2021; Revised 5 July 2021; Accepted 5 August 2021; Published 28 August 2021

Academic Editor: Jinbo Xiong

The firefighting IoT platform links multiple firefighting subsystems. The data of each subsystem belongs to the sensitive data of the profession. Failure prediction is a crucial topic for firefighting IoT platforms, because failures may cause equipment injuries. Currently, in the maintenance of fire IoT terminal equipment, fault prediction based on equipment time series has not been included. The use of intelligent technology to continuously predict the failure of firefighting IoT equipment can not only eliminate the intervention of regular maintenance but also provide early warning of upcoming failures. In order to solve this problem, we propose a vertical federated learning framework based on LSTM fault classification network (LstFcFedLear). The advantage of this framework is that it can encrypt and integrate the data on the entire firefighting IoT platform to form a new dataset. After the synthesized data is trained through each model, the optimal model parameters can be finally updated. At the same time, it can ensure that the data of each business system is not leaked. The framework can predict when IoT equipment will fail in the future and then provide what measures should be used. The experimental results show that the LstFcFedLear model provides an effective method for fault prediction, and its results are comparable to the baseline.

1. Introduction

The firefighting IoT platform is one of the key safeguards for enterprise fire safety. However, the current fire Internet of things platform has low accuracy in identifying various types of alarm information. How to effectively identify the false alarm information of the fire Internet of things platform is very important. The current firefighting IoT platform is linked to multiple firefighting subsystems, such as smoke and sprinkler sensors in the office area and power environment monitoring in the substation. Since the data belongs to different business departments, the data of each department is expected to run on their own independent systems, which requires that each data cannot interact with other data. However, in order to improve the accuracy of the false alarm prediction of the fire Internet of things platform, it is necessary to use all the business data to train the network model. Based on this, we introduced a federated learning framework and proposed the LstFcFedLear network. The accuracy of fault prediction is one of the keys to ensure the normal operation of the fire IoT. Predictive science is a discipline that analyzes a large amount of data and discovers some potential relevance among them. This provides an important basis for industry equipment failure prediction [1, 2]. In order to further improve the accuracy of various failure predictions, many new technologies (for example, artificial intelligence, big data, and blockchain) have been gradually applied to factories [3, 4].

In essence, failure prediction is to correlate the occurrence of failure events with the failures that may occur in the future. Failure prediction has made important developments in 1979. The corresponding mathematics of fault prediction is to use input to predict output. Due to the widespread existence of nonlinearity and uncertainty, it is more difficult to establish an efficient system model in mathematics, which is also one of the objective reasons for missed detection and false alarms. Later, Box et al. proposed the application of time series to forecasts, which greatly improved the accuracy. The characteristic of the neural network is that it can perform nonlinear mapping, so it is widely used in the field of prediction. Unfortunately, neural networks need to be data-driven, and the biggest thing is that they need to manually set the network model parameters.

From the current enterprise monitoring system, it is relatively easy to obtain a large amount of historical equipment data and operating data. Therefore, it is feasible to use historical data to predict failures. With the rapid development of new technologies, it is also feasible to use artificial intelligence, big data, and other technologies to assist in forecasting. At present, in the use of artificial intelligence for fault prediction research, a supervised or unsupervised method is one of the two most common methods. For the design of the network structure, the designer can only rely on experience to subjectively design the depth of the network, the number of neurons, and other parameters. Designers with different experience design different networks. This leads to a problem, and the same problem may have different solutions. Among the many algorithm models, the support vector machine algorithm is more widely used.

If the condition of all equipment in the factory can be monitored and the failure can be alerted in advance, the reliability and stability of the entire factory can be greatly increased. In recent years, in the field of PHM, a lot of research on these fault topics has been carried out, which greatly reduces the cost of fault maintenance of factory equipment and also improves the efficiency of the factory [5]. The key function of PHM is to diagnose equipment failures and discover the causes of equipment failures [6]. Equipment failure prediction is very challenging, and the main reason is the need to consider both the maintenance plan and the type of failure. Over the years, the forecasting model has been continuously developed and improved. But so far, the complex algorithm model [7, 8] still has many limitations. In order to overcome these shortcomings, some studies have adopted machine learning algorithms, such as neural networks [4] and support vector machines (SVM) [5] to predict failure types. These studies have promoted the development of probabilistic models to a certain extent [9, 10], but probabilistic models lack clear physical meaning in fault prediction.

Compared with traditional machine learning algorithms that cannot process time series data, the advantage of the LstFcFedLear method is that it can predict the sequence of future data through learning from historical experience. The main contributions of this article to our work are summarized as follows:

(1) We propose a vertical federated learning framework based on LSTM fault classification network (LstFcFedLear). The advantage of this framework is that it can encrypt and integrate the data on the entire firefighting IoT platform to form a new dataset

(2) The LstFcFedLear model can ensure that the data of each business system is not leaked. This framework can predict the probability of future failure of the fire IoT and can provide corresponding measures to solve the failure

The structure of this article is as follows: Section 2 introduces related research. Section 3 introduces the new framework method. Section 4 shows the experimental verification results. Section 5 is the conclusion and future work.

2. Related Works

VSC and MMC lack the ability to regulate DC short-circuit current during DC faults. For multiterminal DC system fault detection, the calculation of short-circuit current during the discharge phase of the DC fault capacitor is crucial. Li et al. proposed a transient equivalent model suitable for fault analysis of multiterminal DC systems. This model only retained the high-frequency components in the original fault network, which greatly simplified the circuit analysis at the initial stage of the fault [11]. Since the current waveform when the arc fault occurs is very similar to the current waveform of some loads, it is difficult to detect arc faults through simple current characteristics. Aiming at this problem, Lin et al. proposed an arc fault detection method combining a self-organizing feature mapping network and a sliding window method [12]. On the basis of autonomously mining the inherent characteristics of current data, the current signal is continuously detected by using the correlation and continuity between adjacent periodic current samples. The proposed method can effectively realize arc fault detection, and the accuracy of arc fault detection can reach 99%.

The existing bearing fault alarm system is mainly based on the rule diagnosis of a single shaft temperature variable, and the alarm is not timely. In response to the above problems, Liu et al. combined the correlation of the multiaxis axle temperature of the same car and proposed a data-driven method for detecting and positioning train bearing faults [13]. The proposed DiCCA modeling method is verified by using the axle temperature data of a train in actual operation, and the results show the effectiveness of the proposed method. Based on a data-driven approach, Xiong et al. proposed an edge-assisted privacy protection original data sharing framework, which ensures that the data connected to autonomous vehicles will not be destroyed [14].

Using the sensor data of the traction system, Chen et al. proposed an optimal data-driven fault detection method to solve the fault problem of the dynamic traction system [15]. And based on the improved SVM, the optimal data-driven fault diagnosis problem is studied. Finally, through the actual high-speed train experimental platform, the rationality and effectiveness of the proposed method are verified. Yang et al. proposed a data-driven soft closed-loop fault-tolerant control strategy for the voltage sensor failure of the DC side capacitor in the H-bridge structure of STATCOM [16]. This method selected capacitor voltage and system output current as original signals and established the MLS-SVM prediction model based on historical operating data [17]. The predictive output of the MLS-SVM model and the residual signal output by the actual sensor are used to establish a sensor fault detection and judgment mechanism.

The test results showed that the method has good accuracy and real-time performance [18].

Condition monitoring and fault diagnosis are necessary means to ensure the safe and stable operation of mechanical equipment. Wang et al. proposed a deep learning framework based on ABiLSTM for intelligent fault diagnosis of mechanical equipment [19]. The framework first preprocesses the raw data collected by the sensor and divides it into a training sample set and a test sample set. Secondly, Oramas and Tuytelaars extracted features of the original time-domain signal by training multiple bidirectional LSTM networks of different scales and obtained multiscale features of equipment failures [20]. The experimental results show that the ABiLSTM model can achieve multiscale feature extraction of the original signal. By comparing with methods such as CNN, DAE, and SVM, the fault recognition performance of the ABiLSTM model is better than that of various common models [21, 22]. The results of generalization performance experiments on the ABiLSTM model show that the fault recognition accuracy of samples under off-changing conditions can still reach more than 95%; the LSTM network architecture is shown in Figure 1.

3. Materials and Methods

3.1. LSTM Network. A Recurrent Neural Network (RNN) is a type of neural network specially used to process time series data samples. Each layer of RNN not only outputs to the next layer but also outputs a hidden state. RNN's convolutional neural network can be easily extended to images with a large width and height, and some convolutional neural networks can also handle images of different sizes. RNN can be extended to longer sequence data, and most RNNs can handle data with different sequence lengths. It can be seen as a fully connected neural network with self-loop feedback. In forward propagation, the hyperbolic tangent activation function is generally used from the input layer to the hidden layer [23]. The hidden layer to the output layer uses softmax to map the output to a probability distribution of $(0, 1)$. We will see that the output value of the hidden layer at the current moment is affected not only by the input at the current moment but also by the input at all times in the past. In this way, the output value of the hidden layer can be regarded as the memory of the network, which makes it very suitable for processing data samples that have dependencies before and after [24, 25]. An important feature of RNN is that the parameters of the model are shared at different times. This allows us to share the statistical strength of different locations over time. When some parts of the sequence data appear in multiple locations, this parameter sharing mechanism becomes particularly important [26, 27]. LSTM is a type of RNN. The timing backpropagation algorithm transmits the error information step by step in the reverse order of time. When the length of each time series training data is large or the time is small, the gradient of the loss function with respect to the hidden layer variable at a certain time is more likely to disappear or explode.

The input vector of a standard RNN network is $x = (x_1, \cdots, x_T)$. The RNN network uses equations (1) and (2) to solve the hidden vector $h = (h_1, \cdots, h_T)$ and the output vector $y = (y_1, \cdots, y_T)$.

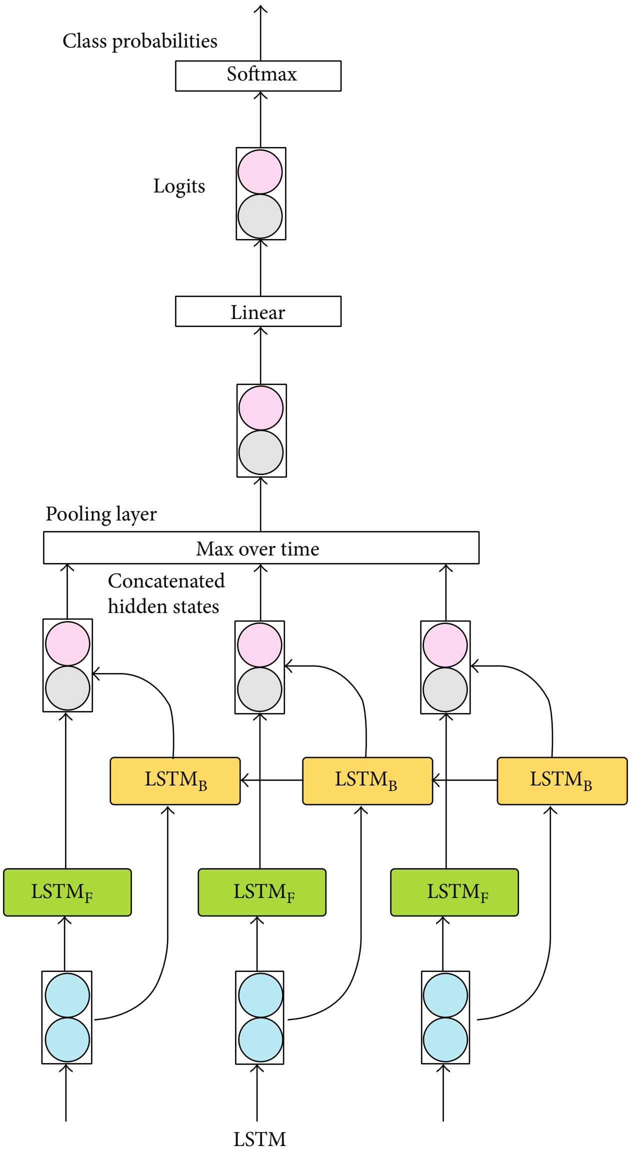


Figure 1: The LSTM network architecture.

$$h_t = \sigma(W_{ih} x_t + W_{hh} h_{t-1} + b_h), \quad (1)$$

$$y_t = W_{ho} h_t + b_o. \quad (2)$$

Among them, W_{ih} refers to the input weight matrix. W_{hh} refers to the weight of the hidden layer. W_{ho} refers to the calculated output matrix of the hidden layer. b_h and b_o refer to all bias vectors, and σ is usually set as a sigmoid function $\sigma(x) = 1/(1 + \exp(-x))$.

The biggest problem encountered by RNN is the gradient problem of gradient disappearance and gradient explosion. LSTM is one of the RNN architectures, essentially using memory cells and gate cells to solve the problem of gradient disappearance and gradient explosion. The memory cell function of LSTM focuses on the input gate unit, which can

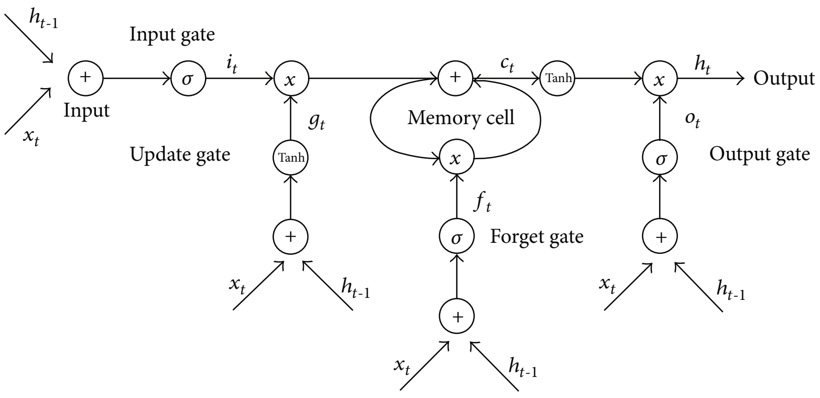


Figure 2: The process of the LstFcFedLear network.

make the state of each memory cell free from external interference. Each multiplication forget gate allows the memory to filter out irrelevant storage contents. The activation function of each multiplication forget gate is essentially an application program, which is mainly used for the calculation of the state of the internal memory unit. This article uses the LSTM structure proposed by Gers et al. The results of the memory unit and gate unit are shown in

$$i_t = \sigma(W_{xi} x_t + W_{hi} h_{t-1} + W_{ci} c_{t-1} + b_i), \tag{3}$$

$$f_t = \sigma\left(W_{xf} x_t + W_{hf} h_{t-1} + W_{cf} c_{t-1} + b_f\right), \tag{4}$$

$$c_t = f_t c_{t-1} + i_t \tan h(W_{xc} x_t + W_{hc} h_{t-1} + b_c), \tag{5}$$

$$o_t = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + W_{co} c_t + b_o), \tag{6}$$

$$h_t = o_t \tanh (c_t). \tag{7}$$

Among them, the input vectors are f_t, o_t, and c_t, which correspond to the vectors of the input gate, forget gate, and output gate at time t. It is worth noting that we regard the vector with the same size as the hidden vector h_t. The weight matrix W refers to the connection coefficient matrix between two different bodies.

The LSTM network is mainly composed of 32 bidirectional units, followed by the 50% discard layer and the sigmoid activation function. This uses L2 regularization to prevent network overfitting. In the model training, the cross-entropy loss function and Adam optimizer are used to train and solidify each LSTM network. When a fault is evaluated, LSTM divides the output fault into 5 levels, from small to large (0-4).

The performance of the pure LSTM model is better than that of the hybrid model [28]. Figure 2 shows the structure of the proposed model. By extracting the characteristics of the sequence data and using them as the input of the convolutional neural network model, the spatial characteristics of the data can be obtained. In order to prevent overfitting, Figure 3 adds a layer to prevent overfitting.

3.2. Data Preprocessing. In this article, the premise is that we predict daily failures. To this end, we focus on the cluster location and date of occurrence. This turns the research question into determining when a failure occurs. Since we are solving two different prediction problems here, namely, binary classification and regression, we consider the production process of each type of prediction input dataset.

To address the problem of fault type classification for fire facility, we used the Fault Type of Fire Facility (FTFF) dataset from the Firefighting Internet of Things platform database of China State Grid Gansu Electric Power Company. This dataset contains two subdatasets, namely, FTFF1 and FTFF2.

We study the real dataset of 15084 alarm records of power firefighting equipment recorded between 2019 and 2020 from fault in power fire facility maintenance. All the FTFF datasets contain Alarm Time (AT), Fault Type (FT), Failure Equipment (FE), Fault Location (FL), Municipal Units (MU), and Level 2 Units (L2U). Considering the large number of subjects in each of these two datasets, we used the FTFF1 dataset for training and the FTFF2 dataset for testing.

The problem of fault type classification was approached as a five-class classification problem, with the following classes: 0 for the Overdue Fault (OF), 1 for the Offline Fault (OLF), 2 for the Power Failure (PF), 3 for the Equipment Damage (ED), and 4 for the False Alarm (FA). For each input data from one discrete time point, if the classification model identified this input as being of class 0–4, then the fault type of this discrete time point would be set to 0–4 (i.e., the fault type was predicted every time point).

The original fault dataset is first transformed into a fault vector, $v_R = (v_{R1}, \cdots, v_{RT})$. In the regression algorithm model, for the fault at a certain time t, the text is expressed as $v_{Rt} = (v_{t1}, \cdots V_{ts})$. Among them, v_{ts} refers to the fault in the s-th space at a certain time t. In the simplified neural network, the sigmoid and hyperbolic tangent function are usually integrated in the gate and used as the activation function. The purpose is to convert the input value to between 0 and 1, where 1 means it is worth paying attention to and 0 means not needing attention. On the other hand, the role of the tanh function is to adjust the network performance by compressing the value to between -1 and 1. LSTM-FC is very sensitive to causality.

3.3. Prediction Models. In the design of this article, we design two types of fault diagnosis models: binary classification and prediction. In this research, we develop two types of failure

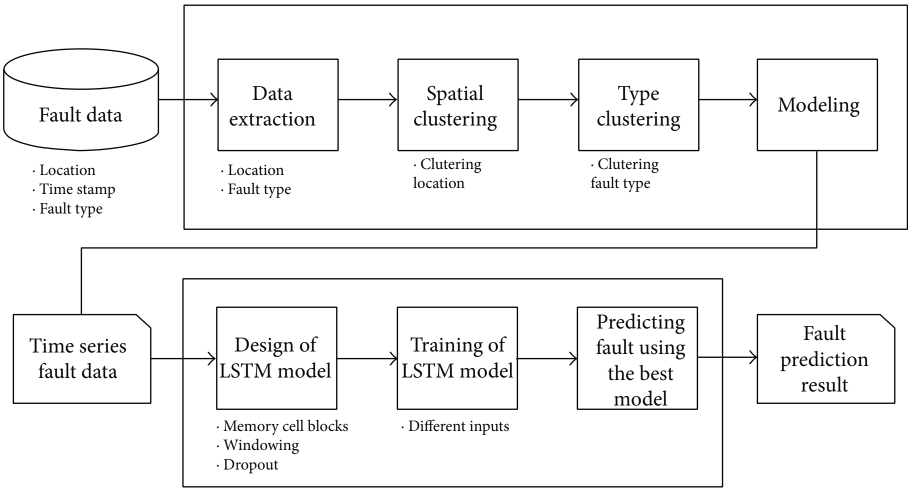


Figure 3: The proposed method for data preprocessing.

Table 1: The results of the forecasting model.

Model		Results	
Specificity = $\frac{\text{TN}}{(\text{TN} + \text{FP})}$	MAE	RMSE	SDE
KNN	0.323	0.823	0.763
SVM	0.301	0.743	0.692
CNN	0.287	0.665	0.662
LstFcFedLear	0.226	0.619	0.634

prediction models, namely, binary classification and prediction. In addition, we also focus on evaluating the relationship between spatial clusters to judge the impact on the prediction results. In this article, we have designed four types of failure prediction models, as shown in Table 1. In the second section, we introduced that the RNN model can receive delay information and has the ability to judge whether this information has an impact on the storage unit. The proposed LSTM-FC model is shown in Figures 1 and 4. It can be seen that these two models use different types of input data, and the activation functions in the output layer are also different. It is worth noting that it expresses faults through weighting factors, and these weights are determined by the following equation:

$$e_t = h_t w_a, \tag{8}$$

$$a_t = \frac{\exp(e_t)}{\sum_{i=1}^{T} \exp(e_t)}, \tag{9}$$

$$v = \sum_{i=1}^{T} a_i h_i. \tag{10}$$

In these equations, h_t refers to the fault that occurs at time t. w_a is the weight matrix set by the attention layer. a_t refers to the probability of possible failure at time t. v refers to the weighted summation of the probabilities at all times t.

By converting the input into a fault sequence, $X = \{x_1, x_2, \cdots, x_N\}$. x_t refers to the fault that occurs at time t calculated by the LSTM model. At each time step, we first use the forward LSTM to predict the probability of the next failure. The overall goal is to minimize the following objective functions:

$$L_f = -\frac{1}{N} \sum_{t=1}^{N} \log \Pr (X_{t+1} | X_1, \cdots, X_t). \tag{11}$$

Among them, L_f refers to all the parameters of the model in forward prediction. The $\Pr(\cdot)$ function in the LSTM model is calculated as x_{t+1}, which mainly depends on the previous probability.

After getting a set of fault history data, the probability of the next fault can also be predicted through the reverse sequence. Therefore, we have also established a backward LSTM, the purpose of which is to predict the previous failure probability based on the later occurrence probability.

$$L_A = -\frac{1}{N} \sum_{t=N-1}^{0} \log \Pr (a_t | a_N, \cdots, a_{t+1}). \tag{12}$$

Finally, we analyze and classify the fault types and incorporate the key information into the whole process of LSTM training.

In summary, by using the optimized model, that is, equation (13), the parameters of the algorithm model can be obtained. After that, the corresponding topological quantities can be calculated.

3.4. LstFcFedLear Model. LSTM-FC can calculate the topological value of the genetic network. Using this advantage of LSTM-FC, the LSTM-FC network algorithm can be iterated repeatedly to obtain the most reasonable parameter matrix. In Algorithm 1, we show the algorithms of the LSTM-FC method one by one.

In order to be able to encrypt and integrate the data on the entire fire IoT platform to form a new dataset and to ensure that the data of each business system is not leaked, we have designed the following framework, as shown in Figure 5. Then, the new dataset is fed into the LstFcFedLear

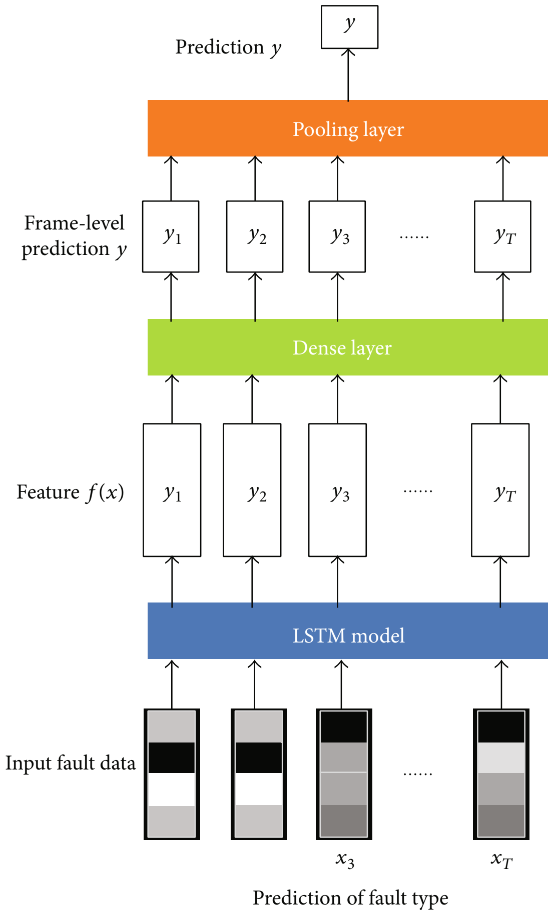


Figure 4: The prediction with LstFcFedLear from the dataset.

```
INPUT PARAMETERS: Reasonable labeling T+,T−
Label labeling is not reasonable ˜ T+, ˜ T−
Maximum number of pre-training
Maximum number of training
OUTPUT VALUE: Train well-performing models
1: d, c ←0
2: get x on T
3: while d < preEpoch do
4:      for each fault i in T do
5:          get ˜r+ i ,˜r−i in L
6:          get L by (9) on i;
7:          update θ;
8:          compute Lf
9:          update θ;
10:   end for 11 j ← j +1; 12 end while
13: while k < trainEpoch do
15:      for each fault xi in T do
16:          compute xi;
17:          compute yi;
18:          compute LA;
19:          update θ;
20:   end for
21:   k ← k +1;
22: end while
```

Algorithm 1: LSTM-FC network algorithm.

model for training as a training set. The LstFcFedLear submodel corresponds to each fire subdata. The LstFcFedLear submodel is responsible for training each subdata to obtain the corresponding training parameters. Finally, all LstFcFedLear submodels update their parameters to a unified model. Specific steps are as follows:

Step 1. The central server sends the public key to the LstFcFedLear1, LstFcFedLear2, LstFcFedLear3,…, LstFcFedLearn models and uses the Paillier partial homomorphic encryption algorithm to align the encrypted samples. The Paillier encryption algorithm is mainly divided into three steps. The first is to generate a key according to the Paillier encryption algorithm. Then, use the generated key to encrypt each part of the data. Finally, after the model training is completed, the model is decrypted [29].

Step 2. The encrypted samples are fed to the LstFcFedLear1, LstFcFedLear2, LstFcFedLear3,…, LstFcFedLearn models for iterative training, and the local parameter gradients of the models are calculated, respectively.

Step 3. LstFcFedLear1, LstFcFedLear2, LstFcFedLear3,…, LstFcFedLearn models push the gradient and loss calculated by each to the central server. The central server uses the private key to decrypt.

Step 4. The central server sends the decrypted gradient and loss back to the LstFcFedLear1, LstFcFedLear2, LstFcFedLear3,…, LstFcFedLearn models.

Step 5. LstFcFedLear1, LstFcFedLear2, LstFcFedLear3,…, LstFcFedLearn models update the model parameters.

Step 6. LstFcFedLear1, LstFcFedLear2, LstFcFedLear3,…, LstFcFedLearn models are iteratively trained to generate a joint model.

4. Experiment Results

4.1. Dataset. To address the problem of fault type classification for fire facility, we used the Fault Type of Fire Facility (FTFF) dataset from the Firefighting Internet of Things platform database of China State Grid Gansu Electric Power Company. This dataset contains two subdatasets, namely, FTFF1 and FTFF2.

We study the real dataset of 15084 alarm records of power firefighting equipment recorded between 2019 and 2020 from fault in power fire facility maintenance. All the FTFF datasets contain Alarm Time (AT), Fault Type (FT), Failure Equipment (FE), Fault Location (FL), Municipal Units (MU), and Level 2 Units (L2U). Considering the large number of subjects in each of these two datasets, we used the FTFF1 dataset for training and the FTFF2 dataset for testing.

The problem of fault type classification was approached as a five-class classification problem, with the following classes: 0 for the Overdue Fault (OF), 1 for the Offline Fault (OLF), 2 for the Power Failure (PF), 3 for the Equipment Damage (ED), and 4 for the False Alarm (FA). For each input data from one discrete time point, if the classification model identified this input as being of class 0–4, then the fault type of this discrete time point would be set to 0–4 (i.e., the fault type was predicted every time point) [30–32].

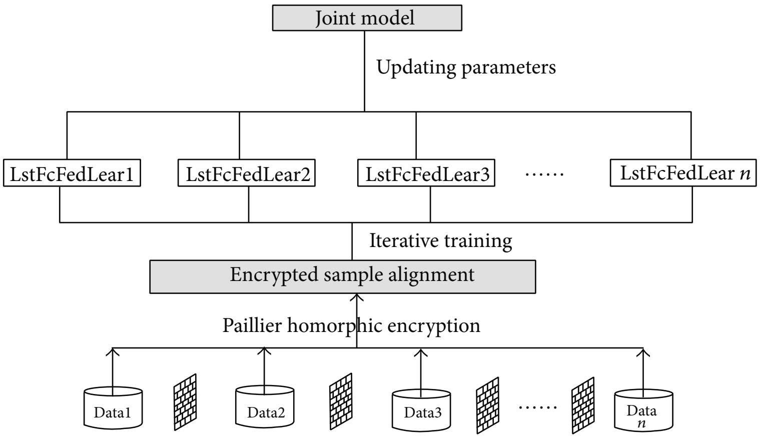


Figure 5: LstFcFedLear model framework.

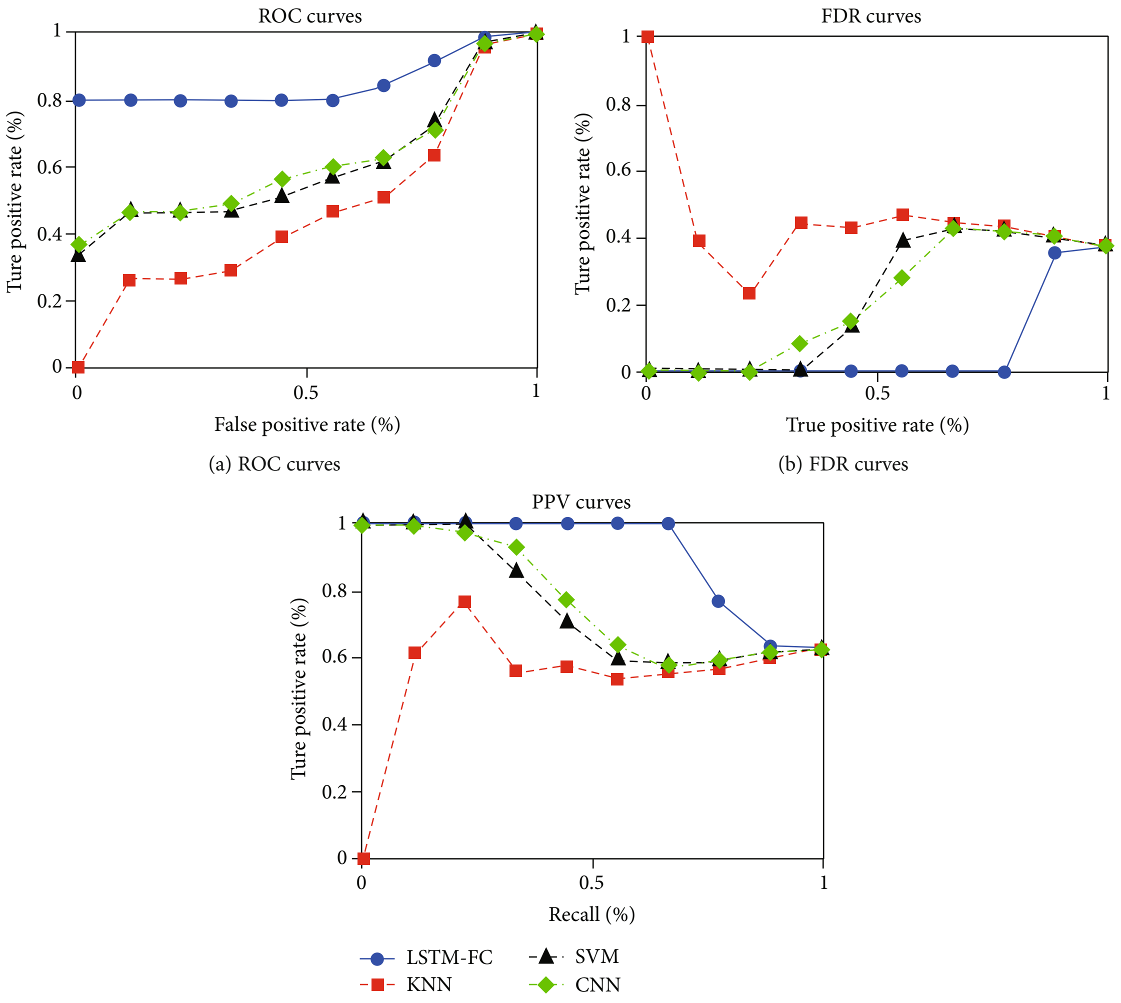
4.2. Prediction by LstFcFedLear. It is easy to see that displaying a certain vector in a sequence or a certain sequence in a sequence is basically the same as the training process of the LstFcFedLear algorithm. The following figure illustrates the training progress curve of the LstFcFedLear and SVM algorithm in detail. It can be clearly seen from the figure that the loss and root mean square error performance of the two algorithm models in the training process are very similar. The loss curve can explain that the learning speed of SVM is relatively slow at the beginning of training. But it is worth noting that as time goes by, the learning curve of SVM is close to a certain value, and it has gradually stabilized. It can be inferred from these data that LstFcFedLear did not learn enough knowledge at the beginning, but over time, this problem was solved. In general, the performance of LTSM is slightly better than that of LstFcFedLear.

In order to further demonstrate the accuracy and generalization ability of LstFcFedLear, we compare its accuracy with the other three methods in [7–9]. In order to show the awareness of the results of the experiment, we use the method in [10]. The simulated test data was obtained using SynTReN, an environment frequently used in the industry. Figure 1 illustrates the operational characteristics (ROC) of the new data generated by LstFcFedLear and CNN. Obviously, on the synthesized test data, the accuracy of LstFcFedLear is better than that of the CNN method. As shown in Figure 6(b), compared with the FDR performance of SVM, KNN, and CNN methods, the error rate of LstFcFedLear is the lowest. This result clearly shows that for the genetic disease dataset, LstFcFedLear is better than SVM, KNN, and CNN.

As shown in Figure 6(c), the comparison between the positive prediction curve of LstFcFedLear and the PPV of SVM, KNN, and CNN shows that the LstFcFedLear method is optimal. This also further shows that LstFcFedLear is also superior to SVM, KNN, and CNN in terms of synthesizing genetic test data.

In this experiment, the LstFcFedLear model has 16 to 100 storage units. The training time interval of the model is $[16, 100]$, and the unit is s. Throughout the experiment, the mean square error is the only indicator that measures the performance of binary classification and regression models in the learning phase. In order to reduce the loss and increase the learning rate, the Adam optimizer is used in the LstFcFedLear model with the parameters beta1 = 0.9 and beta2 = 0.999. In order to prevent overfitting in the learning phase, a total of 3 times of cross-validation were used in this experiment. The pros and cons of the model's hyperparameters are the key to whether a model can achieve the best performance. In this experiment, we repeatedly test the hyperparameters of the LstFcFedLear model, such as the model's backtracking situation, the number of storage units, and the loss rate. The backtracking rate indicates how large the time interval to consider [33]. Table 2 shows the different dropout probability values in the experiment. What we want to explain here is that the loss function in the article represents the mean square error between the training value and the predicted value. In addition, it can be clearly seen from Table 2 that the accuracy of LstFcFedLear is as high as 94.6, which is 8.03% higher than the average of KNN, SVM, and CNN. The sensitivity of LstFcFedLear is as high as 93.4, which is 7.77% higher than the average of KNN, SVM, and CNN. The specificity of LstFcFedLear is as high as 95.1, which is 8.37% higher than the average of KNN, SVM, and CNN. These three indicators also show that LstFcFedLear is the best.

4.3. Performance Comparison. Here, we compare the performance of SVM, KNN, and CNN in detail with the performance of the LstFcFedLear model proposed in this paper. The experiment uses the scikit-learn package in Python for testing. In order to be able to make a thorough comparison with the performance of the LstFcFedLear model, a 3-fold cross-validation was specifically used in the experiment. For the best training parameters, grid search technology is also used in the experiment. It can be seen from the experimental results that for SVM, the value of the penalty parameter is set to $c = 0.1$. In Tables 1–3, we, respectively, compared the fault classification performance of KNN, SVM, CNN, and LstFcFedLear in detail. The comparison results show that the LstFcFedLear model has the best performance. In terms of accuracy and recall, LstFcFedLear is the best among these three. The second place is the CNN model. Table 2 mainly



(a) ROC curves

(b) FDR curves

(c) PPV curves

Figure 6: Accuracy and precision for prediction/classification among LstFcFedLear and different competitors based on fault datasets: (a) ROC diagram, (b) FDR chart, and (c) PPV graph.

Table 2: The comparison results between the traditional fault prediction method and the method proposed.

Method	Accuracy (%)	Sensitivity (%)	Specificity (%)
KNN	84.2	81.3	83.3
SVM	86.3	87.3	87. 7
CNN	89.2	88.3	89.2
LstFcFedLear	94.6	93.4	95.1

Table 3: The running time of the different models in this paper between LstFcFedLear from SVM, KNN, and CNN.

Method	KNN	SVM	CNN	LstFcFedLear
Training time	4.209	8.703	787.342	983.306
Running time	2.217	4.332	9.243	7.276

compares the comparison results between the traditional fault prediction method and the method proposed in the article.

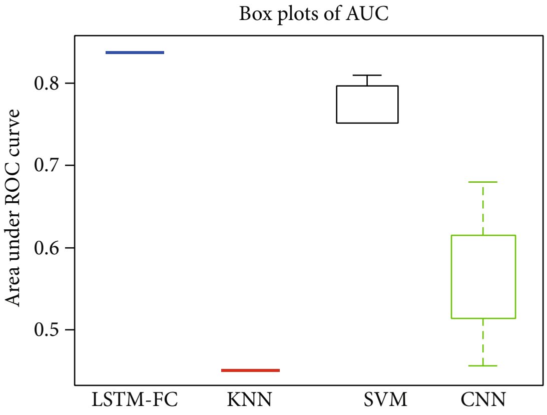


Figure 7: Graph of AUC box plot for LstFcFedLear.

The measurement indicators used in the experiment are sensitivity, accuracy, and Youden index scale, that is, true positive (TP), true negative (TN), false positive (FP), and false negative (FN) [28–31]. TP represents the number of samples classified as correct. TN represents the number of samples judged to be false. FP represents the number of samples classified as incorrect. FN represents the number of samples classified as correct. The specific judgment formula is as follows:

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}, \tag{13}$$

$$\text{Sensitivity} = \frac{\text{TP}}{(\text{TP} + \text{FN})}, \tag{14}$$

$$\text{Specificity} = \frac{\text{TN}}{(\text{TN} + \text{FP})}. \tag{15}$$

As shown in Figure 6, the area enclosed by the curve has shown that LstFcFedLear is larger than the other three types. That can show that in terms of accuracy, LstFcFedLear is definitely better than the other algorithms. It can be seen from Figure 6 that the value of LstFcFedLear actually reaches the maximum average number AUC, about 0.84. But the AUC values of SVM, KNN, and CNN are 0.47, 0.78, and 0.59, respectively. The rankings are LstFcFedLear, KNN, CNN, and SVM. In addition, we calculated and visualized the area enclosed under the curve in Figure 6 in order to highlight the accuracy of all query methods. The AUC value obtained by the LstFcFedLear model is about 0.84, which can also indicate that the model is the best. More importantly, the average AUC owned by GlobalMIT is about 0.47, which is obviously much lower than that of LstFcFedLear.

As shown in Figure 7, the LstFcFedLear model has the best performance in classifying all faults into positive probability because the area under the ROC curve corresponding to LstFcFedLear is the largest. According to the area ranking under the ROC curve, it can be seen that the ranking of SVM is only lower than that of LstFcFedLear but is better than that of CNN and KNN in turn. It is worth noting that the ROC area of the LstFcFedLear model is 10 times that of KNN, 6 times that of SVM, and 3 times that of CNN. The huge area difference once again illustrates the excellent accuracy of the LstFcFedLear model.

Table 1 characterizes the accuracy of these algorithm models from another level. The MAE value of LstFcFedLear is 0.226, which is 0.075 lower than the average of KNN, SVM, and CNN. The test results show that the MAE value of KNN is the largest, indicating that the effect of the model is the worst. The ranking of other models from good to bad is CNN, SVM, and KNN. In terms of RMSE, the RMAE value of LstFcFedLear is 0.619, which is 0.123 lower than the average of KNN, SVM, and CNN. The test results show that the RMAE value of KNN is the largest, indicating that the effect of the model is the worst, which is consistent with the performance on the MAE value. The ranking of other models from good to bad is CNN, SVM, and KNN. In terms of SDE, the SDE value of LstFcFedLear is 0.634, which is 0.072 lower than the average of KNN, SVM, and CNN. The test results show that the SDE value of KNN is the largest, 0.763, indicating that the effect of the model is the worst, which is consistent with the performance on the MAE and RMSE values. The ranking of other models from good to bad is CNN, SVM, and KNN.

In Table 3, we compare the running time of the LstFcFedLear, KNN, SVM, and CNN models. It can be seen that although LstFcFedLear is indeed superior in various performances, it pays relatively high in training time and running time costs. As can be seen in Table 3, in terms of training time cost, the time-consuming order is KNN, SVM, CNN, and LstFcFedLear from least to more. KNN is indeed very time-consuming, but its performance is too poor to be suitable for promotion and application in the industry. LstFcFedLear may take a little longer time, but it is relatively stable in terms of performance.

5. Conclusion

In short, we propose a vertical federated learning framework based on LSTM fault classification network to predict the failure of the fire IoT platform. The advantage of this framework is that it can encrypt and integrate the data on the entire firefighting IoT platform to form a new dataset. After the synthesized data is trained through each model, the optimal model parameters can be finally updated. At the same time, it can ensure that the data of each business system is not leaked. The experimental results showed that the LstFcFedLear model provides an effective method for fault prediction, and its results are comparable to the baseline. And the results among LstFcFedLear and SVM, KNN, and CNN methods showed that LstFcFedLear performs better than all methods in RMSE prediction, with the improvement being 9.8% and 24.3%, respectively. In the future, we plan to apply the LstFcFedLear model to power production application scenarios and then further optimize the robustness and other performance of the model.

Data Availability

We used the Fault Type of Fire Facility (FTFF) dataset from the Firefighting Internet of Things platform database of China State Grid Gansu Electric Power Company. This dataset contains two subdatasets, namely, FTFF1 and FTFF2.

Conflicts of Interest

All authors declare no conflict of interest over this article.

Acknowledgments

This work was supported by the State Grid Gansu Electric Power Company Science and Technology Xing'an Project (No. 52272221001Z).

References

[1] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks & Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[2] Y. Li, R. Gault, and T. M. McGinnity, "Probabilistic, recurrent, fuzzy neural network for processing noisy time-series data,"

IEEE transactions on neural networks and learning systems, vol. 2, no. 1, pp. 1–10, 2021.

[3] X. Liu, Y. Zhou, and Z. Wang, “Deep neural network-based recognition of entities in Chinese online medical inquiry texts,” *Future Generation Computer Systems*, vol. 114, pp. 581–604, 2020.

[4] S. N. Bhattu, S. K. Nunna, D. V. L. N. Somayajulu, and B. Pradhan, “Improving code-mixed POS tagging using code-mixed embeddings,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 19, no. 4, pp. 1–31, 2020.

[5] H. Yuan, “Combined networks with multi-level attention for distantly-supervised relation extraction,” *Journal of Physics: Conference Series*, vol. 1550, article 032065, 2020.

[6] S. Chen and M. Wu, “Attention collaborative autoencoder for explicit recommender systems,” *Electronics*, vol. 9, no. 10, p. 1716, 2020.

[7] R. Kiros, R. Salakhutdinov, and R. S. Zemel, *Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models*, TACL, 2015.

[8] J. Xiong, J. Ren, L. Chen et al., “Enhancing privacy and availability for data clustering in intelligent electrical service of IoT,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2019.

[9] X. Liang, L. Lin, W. Yang, P. Luo, J. Huang, and S. Yan, “Clothes co-parsing via joint image segmentation and labeling with application to clothing retrieval,” *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1175–1186, 2016.

[10] Y. Zuobin, S. I. Yuanping, M. A. Jianfeng, J. Wenjie, and X. U. Shengmin, “P2HBT: partially policy hidden E-healthcare system with black-box trace ability,” *Chinese Journal of Electronics*, vol. 30, no. 2, pp. 219–231, 2021.

[11] J. Li, Y. Li, N. Yuan, K. Jia, and G. Song, “DC fault analysis and detection for offshore wind farms integration via MTDC,” *Electric Power Automation Equipment*, vol. 12, pp. 119–128, 2020.

[12] J. Lin, Y. Wang, K. Li, and M. Tian, “Arc fault detection method based on self-organizing feature mapping network,” *Electric Power Automation Equipment*, vol. 8, pp. 210–219, 2020.

[13] Q. Liu, X. Fang, Y. Dong, and S. Qin, “Dynamic modeling and reconstruction based fault detection and location of train bearings,” *Chinese Association of Automation*, vol. 12, pp. 2233–2241, 2019.

[14] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, “Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles,” *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.

[15] H. Chen, B. Jiang, H. Yi, and N. Lu, “Data-driven fault diagnosis for dynamic traction systems in high-speed trains,” *SCIENCE CHINA Information Sciences*, vol. 50, no. 4, pp. 496–510, 2020.

[16] X. Yang, W. Duan, W. Chen, and J. Wang, “Study on fault tolerant control strategy of sensor fault in STATCOM,” *Journal of power capacitor and the reactive power compensation*, vol. 5, pp. 36–41, 2018.

[17] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, “Recurrent neural network based language model,” in *Eleventh annual conference of the international speech communication association*, pp. 1–15, Beijing, 2010.

[18] M. Sadrzadeh, D. Kartsaklis, and E. Balkır, “Sentence entailment in compositional distributional semantics,” *Annals of Mathematics and Artificial Intelligence*, vol. 82, no. 4, article 9570, pp. 189–218, 2018.

[19] T. Wang, T. Wang, P. Wang, H. Qiao, and M. Xu, “An intelligent fault diagnosis method based on attention-based bidirectional LSTM network,” *Journal of Tianjin University Science and Technology*, vol. 6, pp. 601–608, 2021.

[20] J. Oramas and T. Tuytelaars, “Modeling visual compatibility through hierarchical mid-level elements,” 2016, https://arxiv.org/abs/1604.00036/.

[21] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, “Jointly modeling embedding and translation to bridge video and language,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, 2016.

[22] Y. Tian, Z. Wang, J. Xiong, and J. Ma, “A blockchain-based secure key management scheme with trustworthiness in DWSNs,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” vol. 1512, 2015https://arxiv.org/abs/1512.00567/.

[24] J. Xiong, X. Chen, Q. Yang, L. Chen, and Z. Yao, “A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2347–2360, 2020.

[25] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. Belongie, “Learning visual clothing style with heterogeneous dyadic cooccurrences,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4642–4650, Chengdu, 2015.

[26] J. Xiong, R. Ma, L. Chen et al., “A personalized privacy protection framework for mobile crowdsensing in IIoT,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.

[27] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg, “Retrieving similar styles to parse clothing,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 5, pp. 1028–1040, 2015.

[28] W. Yang, P. Luo, and L. Lin, “Clothing co-parsing by joint image segmentation and labeling,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3182–3189, Shanghai, 2014.

[29] W. Fang, M. Zamani, and Z. Chen, “Secure and privacy preserving consensus for second-order systems based on Paillier encryption,” *Systems & Control Letters*, vol. 148, pp. 104869–104882, 2021.

[30] T. Yao, T. Mei, and C.-W. Ngo, “Learning query and image similarities with ranking canonical correlation analysis,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 28–36, Xiamen, 2015.

[31] L. Yu, E. Park, A. C. Berg, and T. L. Berg, “Visual Madlibs: fill in the blank description generation and question answering,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2461–2469, Guangzhou, 2015.

[32] J. Xiong, M. Zhao, M. Bhuiyan, L. Chen, and Y. Tian, “An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2021.

[33] J. Zhao, X. Zhang, F. di et al., “Exploring the optimum proactive defense strategy for the power systems from an attack perspective,” *Security and Communication Networks*, vol. 2021, Article ID 6699108, 14 pages, 2021.