Received October 13, 2020, accepted October 27, 2020, date of publication October 30, 2020, date of current version November 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3034939

SVM-BiLSTM: A Fault Detection Method for the Gas Station IoT System Based on Deep Learning

YAO JIAHAO[1], XIAONING JIANG[1], SHOUGUANG WANG[1], (Senior Member, IEEE), KELEI JIANG[2], (Student Member, IEEE), AND XIAOHAN YU[1]

[1]School of Information and Electronic Engineering, Zhejiang Gongshang University, Zhenjiang 310018, China
[2]College of Arts and Sciences, University of Washington, Seattle, WA 98195, USA

Corresponding author: Xiaoning Jiang (jiangxiaoning@zjgsu.edu.cn)

This work was supported in part by the National Nature Science Foundation of China (NSFC) under Grant 61701441, in part by the Key Projects of the Zhejiang Science and Technology Plan under Grant 2018C01084, in part by the Zhejiang Natural Science Foundation under Grant LQ20F020009, and in part by the Zhejiang Provincial Key Laboratory of New Network Standards and Technologies, Zhejiang Gongshang University, under Grant 2013E10012.

ABSTRACT In this article, a bi-directional long-short term memory (BiLSTM) network algorithm combined with a support vector machine (SVM), SVM-BiLSTM, is proposed to detect faults in the Gas Station Internet of Things (GS-IoT) system. The operational process data in the GS-IoT System, which is collected from the edge of the IoT gateways, is compared with the human emotional reaction behavioral mechanism data. A word segmentation method is invented to map the collected data to a low dimensional space, which makes the data processing relatively easier while retaining the intrinsic information of the data. In order to deal with a certain correlation among the data of the GS-IoT system, the BiLSTM algorithm is used to analyze the abnormal data and find types of faults. Since the structure of the BiLSTM is complex and its calculation is slow, we design a novel method which leverages SVM to increase the detection efficiency. We also compare the performance of the proposed algorithm with Convolutional Neural Networks (CNN), Gated Recurrent Unit (GRU), Knowledge-Based System (KBS), pure SVM and BiLSTM. The results show that the proposed algorithm is able to detect faults with more efficiency and accuracy in the GS-IoT system.

INDEX TERMS Deep learning, SVM-BiLSTM, fault detection, sentiment analysis, IoT system.

I. INTRODUCTION

With the advances of the economy and in the context of Industry 4.0, technologies such as the Internet of Things (IoT), edge computing, and cloud computing have developed rapidly. Before the era of big data, fault diagnosis mainly depends on the rich knowledge of the industry field, the accuracy of the diagnosis model and the integrity of the data samples. These methods have advantages such as simplicity, interpretation, and ease of development. However, in the face of large-scale and complex systems, they are vulnerable to interference from the environment and may lead to poor diagnosis results. As an important part of the gas station, the GS-IoT system is typically consisted of an edge service system, an edge IoT gateway, some payment POSs and gas engines. Abilities of the GS-IoT system is essential to the lifeline of gas stations. Faults in the GS-IoT system are usually presented because of the complex reasons and dangerous working environment. Every component of the GS-IoT system is related to the normal operation in the gas station. It is vitally important to find a fast and effective fault detection method for gas station maintenance personnel. Such a method helps maintenance personnel to reach the exact location of the fault and minimizes downtime and economic losses of gas stations. However, it is not practical to install detection devices directly in each component of the system because gas stations are high-risk locations. So it is vital to design a remote intelligent detection method for the GS-IoT system.

Fault detection and diagnosis problems have been widely studied in engineering, and there are usually three solutions to this problem as follows: 1. knowledge-based fault detection (rule-based detection); 2. Fault detection based on data mining; 3. Fault detection based on machine learning and deep learning.

The knowledge-based fault detection discerns a fault based on its characteristics by using the expert knowledge base,

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

which is used to make similarity judgment [1] or develop a fault tree to detect faults [2]–[4].

With the advent of the big data technology, some researches devote to the fault detection based on data mining. The combination of data-driven and expert systems [3] as well as the combination of data-driven and models [5] are developed. There are also some new concepts, such as equipment ECG [6], which effectively improves the fault detection effect.

With the rapid development of artificial intelligence technology, more researchers are paying attention to intelligent fault detection methods. Simple machine learning algorithms such as Bayesian networks are utilized for fault detection [7]. With the excellent performance of the Support Vector Machine (SVM) model, fault detection based on SVM is widely concentrated on [8]. SVM is a machine learning algorithm based on statistical theory, whose advantages include a small sample space, a non-linear, high-dimensional model and easy generalization. References [9]–[11] achieve good results by using the SVM to detect faults of a turbo pump. The SVM algorithm is difficult to implement for large-scale training samples and multi-classification problems. Therefore, improvements can be achieved by combining with other algorithms [12]–[14] and deep structure [15]. In order to tackle large samples in the SVM, Kang *et al.* [16] divide the samples into several sub-regions and assigned different weights according to their euclidean distances to the hyperplane, which distinguishes training times for different weights. As cloud computing grows in maturity, so does online detection technology. Zhang *et al.* [17] develop an online detection model of the SVM grid based on a non-stationary parameter searching method.

The development of computer hardware leads more people to focus on the deep learning technology for accurate fault detection. Cui *et al.* [18] present a kernel principal component analysis and a wavelet neural network to process the original parameter data of an aeroengine. They extract the data from its principal component and adopt the wavelet neural network to carry out a diagnosis and analysis on the feature data sample set. The image processing problem solved by CNN can also be applied for fault detection and get the ideal result [19]. T. Benkedjouh *et al.* [20] describe a Short-Time Fourier Transform (STFT) to process the original data sent by the aeroengine, and make use of CNNs to learn it and determine its fault information. The emergence of LSTM solved the correlation problem that RNN cannot solve [21]. LSTM is good at calculating time series, and it can also optimize its input to achieve better classification results. Principi *et al.* [22] evaluated three different autoencoder architectures, which are the Multi-Layer Perceptron (MLP) autoencoder, the convolutional neural network autoencoder, and the recurrent autoencoder composed of the LSTM units. In addition, LSTM was also used for fault detection [25]–[27]. The emergence of BiLSTM [28] is a better solution to deal with the relationship between contexts. Enshaei and Naderkhani [29] adopt BiLSTM to diagnose the original signal of an asynchronous motor fault, compared it with the GRU, and obtain good results.

Based on the analysis above, we propose a fault diagnosis method based on SVM and BiLSTM to obtain better results. Generally, the data-based method detect faults accurately, and the analysis of the data characteristics distinguish fault types effectively. SVM and BiLSTM cannot address both accuracy and efficiency in fault detection of GS-IOT system. The proposed method in this article effectively solves this problem. At the same time, the traditional fault detection method often needs to collect various physical quantities of the equipment. However, it is not practical to install sensors in the GS-IoT system because gas stations are placed in high-risk locations, and any other external electronic equipment presents a risk of explosion. We take advantage of the IoT technology to effectively solve this problem. The goal of this article is to determine the fault location through analysing the emotion of the interaction log.

The goal of fault detection in this article mainly includes the hardware level of the equipment, as well as the operation failure that some services cannot provide normally. When the faults occur, the communication data of GS-IOT system will change. We use the technology of IOT to collect data. And SVM-BiLSTM is adopted to detect the fault. The first stage, we classify all samples into normal and abnormal ones through SVM. In this case, we can find all the faults that cause data changes, even if such faults are unknown. In the second stage, BiLSTM is used to detect faults accurately. In this way, the accuracy of fault detection is improved by deep learning.

The remainder of this article is organized as follows: in section II, the construction method of SVM, LSTM and BiLSTM are introduced. A novel SVM-BiLSTM fault detection method is elaborated in section III. In section IV, the effectiveness of the proposed methods is shown by the GS-IoT system diagnosis experiments, respectively. Section V concludes this article and gives the research directions in the future work.

II. CORRELATION ALGORITHMS

In this part, we will introduce the correlation algorithms.

A. SVM

Support vector machine is a two-classification model where the basic model is the linear classifier with the largest interval spacing in the feature space, but in the case of nonlinear separability, the simple division of samples by a hyperplane does not exist. The solution is to map the original two-dimensional space to a suitable high-dimensional space so that the sample is linear within the feature space. In a group of **N** training data, $D = \{(x_i, y_i)\}_{\mathrm{i}=1}^{N}, x_i \in R^M$ is the i^{th} first training data, and $y_i \in \{-1, 1\}$ is the sample label. Where $f(x)$ represents a segmentation line in 2-D space or a hyper plane in a high-dimensional space, ω is the weight of input, and b is the intercept of the hyper plane. Divide the hyperplane into feature spaces:

$$f(x) = \omega^T \varphi(\mathrm{x}) + b \tag{1}$$

The formula (1) is convex quadratic programming, and according to the Lagrangian Duality Theory, it transforms the problem of solving the Lagrangian optimal solution by using the Lagrangian multiplier as α. The Lagrangian function can be written as:

$$f(x) = \frac{1}{2}\left\|\omega\right\|^2 + \sum_{i=1}^{N} \alpha_\mathrm{i}(1 - \mathrm{y}_i(\omega^T x_i + b)) \tag{2}$$

For the linearly inseparable data, k(·, ·) is kernel function to achieve linear separation of the samples in a high-dimensional feature space. A kernel function is the core of SVM when dealing with a nonlinear situation. It operates inner products on a highdimension space and linear inseparable data in a feature space are mapped to a high-dimension nonlinear space, namely, the inner product operation of high-dimension's whole vectors is complete with the kernel function [16]. The objective function is:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j k\left(x_i, x_j\right)$$
$$s.t. \sum_{i=1}^{N} \alpha_i y_i = 0, \quad 0 \le \alpha_i \le C \tag{3}$$

Parameter C denotes the penalty dealing with constraint violation for the model. According to the minimum optimization principle, the classification function is obtained by solving the optimization objective $\boldsymbol{\alpha}$:

$$f = \sum_{i=1}^{N} \alpha_i y_i k\left(x, x_i\right) + b \tag{4}$$

B. LONG-SHORT TERM MEMORY NETWORK (LSTM)

LSTM is first introduced by Cho *et al.* [24] as an improved recurrent neural network (RNN) that is able to solve the long-distance dependence problem. The hidden layer of the original RNN has only one state h, which is very sensitive to short-term inputs and shows a gradient disappearance problem during reverse propagation, while the LSTM uses a door structure to effectively resolve gradient disappearance. Figure 1 shows the basic structure of the LSTM.

The LSTM network uses a gate structure to add, delete and process information, which is the output vector of the gate multiplied by the vector to be controlled. Because the output of a gate is a real vector between 0 and 1, when the gate output is 0, any vector multiply by it will result in a vector of 0, which is equivalent to an inability to pass; when the output is 1, any vector multiply by it will not change, which is equivalent to an ability to pass. Because the value range of the sigmoid function is (0,1), the state of the door is half open and half closed. Suppose W is the weight vector of the door, and B is the offset term, then the door can be expressed as Formula (5).

$$\mathrm{g}(x) = \sigma\left(Wx + b\right) \tag{5}$$

LSTM uses two gates to control the content of unit state C. One gate is the forget gate, and the other is the input gate.

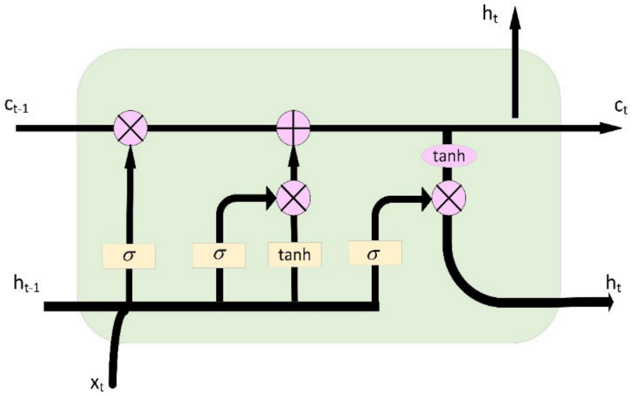


FIGURE 1. LSTM cell.

The output gate is used by LSTM to control the current output value of the unit state to LSTM.

1) FORGOTTEN DOOR

How much of the unit state C_{t-1} of the previous time is reserved to the current time C_t is determined. W_f is the weight matrix of the forgetting gate, b$_f$ is the bias term of the forgetting gate, σ is the sigmoid function, h_{t-1} is the implicit state in t−1, and x_t is the input; the expression of the forgetting gate is

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{6}$$

2) INPUT DOOR

Decide what new information is stored in the cell transposition, which consists of two parts:

1. Sigmoid (input door layer) determines what value sets need to be newer, W_i is the weight matrix of the input gate, and b$_i$ is the offset term of the input gate; the input expression is:

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{7}$$

2. The tanh layer creates a new candidate vector $\tilde{C}_t$, where W_c, b_c is the weight and offset of the candidate vector; the current input cell status is:

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{8}$$

Finally, the unit state C_t at the current moment can be obtained:

$$C_t = f_t * C_{t-1} + i_t * C_t \tag{9}$$

3) OUTPUT DOOR

First, run a sigmoid layer to determine which part of the cell state will be the output O_t. The cell state is the C_t output using tanh (obtaining a value between −1 and 1) and multiplying it by O_t.

$$O_t = \sigma\left(W_O \cdot [\mathrm{h}_{t-1}, x_t] + n_O\right)$$
$$\mathrm{h}_t = O_\mathrm{t} * \tanh\left(C_\mathrm{t}\right) \tag{10}$$

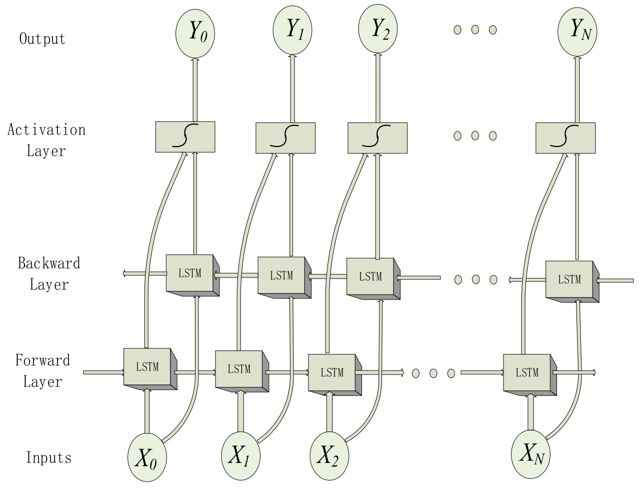


FIGURE 2. The architecture of the BiLSTM network.

C. BiLSTM

LSTM can only predict the output of the next time according to the time sequence information of the previous time. However, in some problems, the output of the current time is related not only to the previous state but also to the future state. The structure of the BiLSTM is shown in Figure 2 [17].

$X_0, X_1 \cdots X_N$ are the input. The forward layer is calculated forward from the moments 1 to N in order to obtain and save the output of each moment forward of the implied layer. The backward layer is calculated in reverse from moment N to moment 1, and the output of each moment is hidden backwards. The final output $Y_0, Y_1 \cdots Y_N$ is obtained by combining the corresponding time output of the forward layer and the backward layer at each time.

III. SVM-BiLSTM

In this part, we will propose our fault detection method.

SVM has some difficulties in dealing with multi- classification problems. What's more, BiLSTM needs forward and reverse calculations, which cause its computation time to increase dramatically, and it is not necessary for shallow features. While the shallow features are relatively obvious and have no strong correlation, the error data is only a small part of the log file and it takes more time to use BiLSTM directly. There is a strong link between the GS-IOT system data before and after. So BiLSTM is more accurate in GS-IOT system fault location than other algorithms. Therefore, we first use SVM to extract the shallow characteristics of the log file [30]. In the second stage, we input the error data into BiLSTM to analyze the specific fault to determine the precise location of the faults. We show how SVM-BiLSTM is trained in Algorithm 1:

We set the input as X, set $y_{SVM} = 0$ to represent normal and $y_{SVM} = 1$ to represent abnormal. The data is labeled y, the range is 0-5, with 0 being normal, and 1 to 5 represent an error type. Next, multiply the y_{SVM} output from SVM and the set label y to obtain the label 0-4 of BiLSTM:

$$y_{BiLSTM} = y_{SVM} * y - 1 \tag{11}$$

Algorithm 1 SVM-BiLSTM Model Training

```
Input: X/*X is the given dataset */
Output: V1, V2/*output model */
Begin
V1←SVMtrain(X, $y_{SVM}$)
IF ($y_{SVM} = 1$)
    $y_{BiLSTM}$ ← $y_{SVM}$*y-1
    T←X- $Xy_{SVM=0}$/* $Xy_{SVM=0}$ is the normal data in given dataset*/
    V2 ← BiLSTMtrain (T, $y_{BiLSTM}$)
    Return V2
ELSE
    X← add new samples
    Return V1
ENDIF
End
```

After obtaining y_{BiLSTM}, X with value of -1 is discarded to obtain T combined with y_{BiLSTM} to form a new input for BiLSTM training. After the model is trained, we use the trained SVM to extract shallow features. The specific implementation is shown in Algorithm 2:

Algorithm 2 Extract the Shallow Characteristics

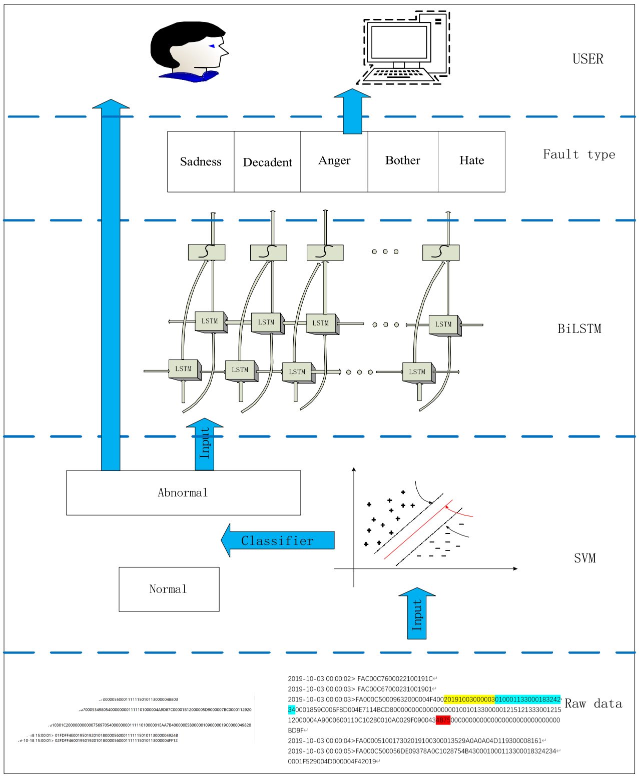
```
Input: V1, X/*X is the real-time data*/
Output: C1, D/*C1 is the result of extracting shallow features, D is the data corresponding to C1 */
Begin
    IF(X != NULL)
        R ← SVMpredict(V1,X)
        C1← argmax(R)
        D ← D.add(X)
    ENDIF
  Return C1,D
End
```

The fine fault prediction is realized in Algorithm 3:

Algorithm 3 SVM-BiLSTM Fine Fault Detection

```
Input: D,C1,V2/* D is the data from algorithm 2,V2 come from algorithm 1 */
Output: C2/*fault detection results*/
Begin
IF(C1=abnormal)
    R ← BiLSTMpredict(V2,D)
    C2 ← argmax(R)
    Return C2
ENDIF
End
```

The first stage has preliminarily classified all samples into normal and abnormal ones. In the second stage, BiLSTM is used to detect faults accurately. After that, we can create an online monitoring real-time system [26],[30] as shown in Figure 3 [37]:



```
2019-10-03 00:00:02> FAC00C7600022100191C
2019-10-03 00:00:03> FAC00C67000231001901
2019-10-03 00:00:03>FA000C50009632000004F40020191003000003010001133000183242
340001859C006F8D004E7114BCD800000000000000000100101330000012151213330012 15
12000004A9000600110C10280010A0029F0900434B7500000000000000000000000000000
BD9F
2019-10-03 00:00:04>FA00005100173020191003000135 29A0A0A04D119300008161
2019-10-03 00:00:05>FA000C500056DE09378A0C1028754B430001000113300018324234
0001F529004D000004F42019
```

FIGURE 3. Usage of SVM-BiLSTM.

We inform users of the accurate fault information detected by BilSTM and the abnormal data classified by SVM. Of course, we can use the model to detect and classify the historic data to search the problem have occured.

IV. EXPERIMENTAL VALIDATION

In this part, experiments are conducted to verify our novel fault detection method.

A. EXPERIMENTAL ENVIRONMENT

The past decade has witnessed the rapid development and wide applications of cloud computing technology. Cloud-based and online fault detection [17] has gradually become a trend. When the GS-IOT system is connected to the cloud, we can analyze the data to detect the fault in real-time. The entire system devices for detecting the fault is shown in Figure 6. On the bottom is the GS-IoT system to be tested. The devices are shown in Station Edge, with the electronic point of sale (EPOS) in the lower left corner, the oil machine (OM) in the right of the EPOS, the oil tank (OT) in the right of the OM, the Edge Internet of Things Gateway (EIOTG) in the middle and the edge server (ES) in the right corner. We collect data from HangZhou, YinChan and ShangHai stations. The data is uploaded to the cloud and is analyzed by SVM-BiLSTM to find abnormal devices on the Fault Detection Edge.

In order to better show how the fault detection system works, Figure 5 shows the software and hardware of each component of the whole system. At the same time,

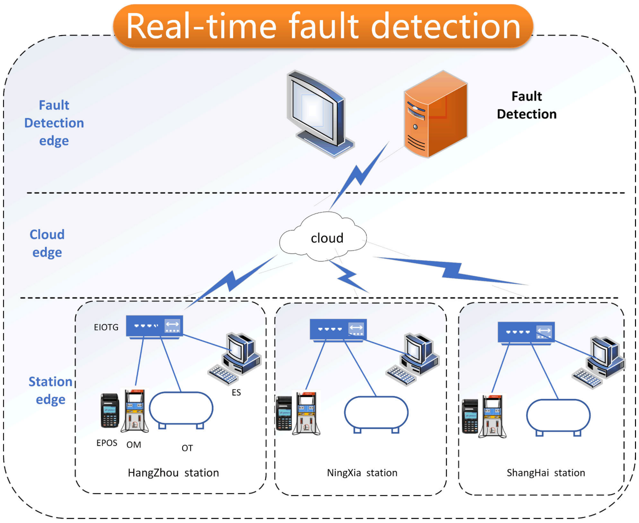


FIGURE 4. The experimental devices.

Figure 5 shows the connections among the components. In the physic and transport layer, the OM, EPOS, and EIOTG use serial port for full duplex communication, the ES, OT and EIOTG communicate through TCP. The OM model is 3100 tax control, using analog chip as hardware. The EPOS uses hardware STM32f407 with computing power to process the refueling business and transmit some control commands to OM. OT uses the sensor of liquid level meter to transmit the volume data of oil in OT. The EIOTG, as the middleman of the system, obtains the data from the EPOS through serial ports, and sends it to the ES. Meanwhile, the EIOTG is responsible for uploading the data that is collected from the ES, OT and EPOS by using Message Queuing Telemetry Transport (MQTT) client. In the cloud layer, Ali ECS which uses ubuntu16.04 as operating system on the left is data storage center. We deploy MQTT server to collect the data uploaded by EIOTG, then send the history data to computational Ali ECS in the form of log and send the real-time data through the MQTT. The SVM-BiLSTM is deployed on the right and adopts the TENSORFLOW as backend. The result is displayed on the fault detection server through the MQTT.

Many researchers are devoted to fault detection for edge devices [32]–[34]. The EPOS and ES are like the two interlocutors. Since the limitations of the device and the protocol, they are unable to talk directly. Figure 6 shows the data flow in GS-IOT system, EIOTG and Cloud are brokers in data transmission.

In order to maintain the normal operation of the GS-IoT system, when the OM status does not change, it sends general query command through the EPOS and EIOTG_Broker to the ES in hexadecimal format according to the protocol. The ES will return data according to OM requirements, enabling the OM to complete operations such as initialization, downloading data, proofreading time etc. When the OM status change, it sends specific commands to the ES to complete operations such as refueling, querying cumulative, downloading blacklist, etc. EPOS will send keyboard commands

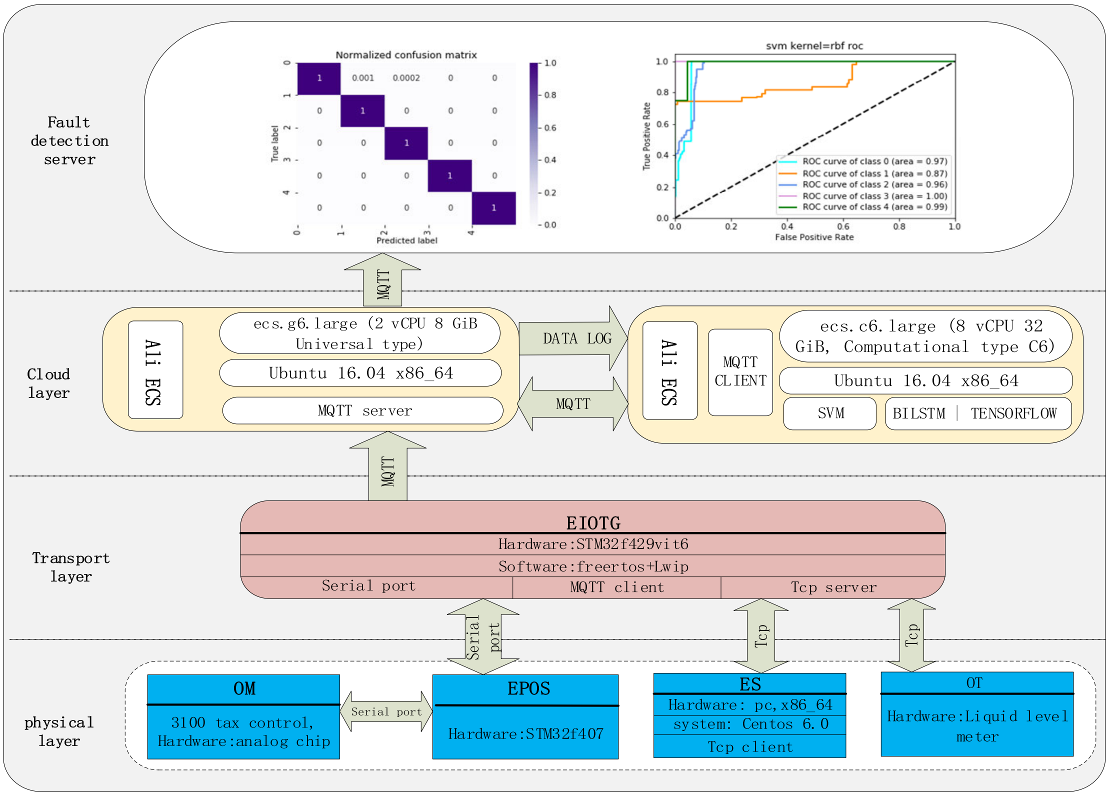


FIGURE 5. System framework.

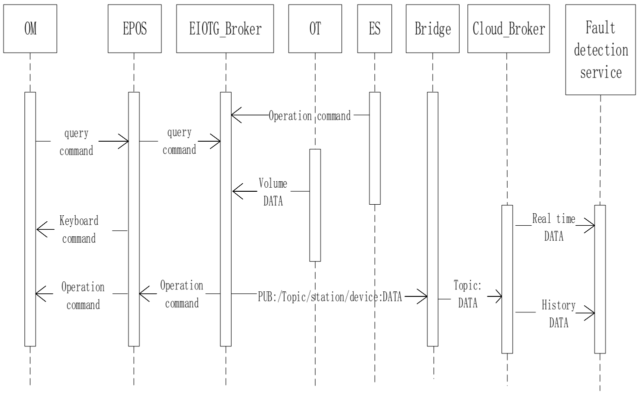


FIGURE 6. Data transmission.

to OM to complete operations such as setting of refueling amount and liter etc. And OT send volume data of oil to the EIOTG_Broker. EIOTG_ Broker will publish all the data with designated topic, and each gas station will bridge the data to Cloud_Broker due to the network reasons.

The EIOTG collects the data in seconds and fully reflect the real-time situation of the entire system. So, when we send data into the model through the MQTT in real time, we can get real-time fault detection. The detected fault will be published to the corresponding topic through MQTT. The fault information is stored by the Cloud_Broker. At the same time, users receive fault notification by subscribing to topics. The Fault notification includes the failure of SVM notification and the specific fault reason of BiLSTM notification. Parameters of EIOTG equipment are shown in Table 1:

TABLE 1. The parameters of the EIOTG.

Name	Parameter
Kernel	With FPU ARM®32-bit Cortex®-M4
Main frequency	80 MHZ
Storage	2 MB FLASH 256 KB SRAM
Serial port	4USART, 4UART, support 232 and 485
Network	PHY on chip, using special DMA, supporting IEEE1588V2, MII/RMII

Through the EIOTG, we collect 2,000,000 pieces of data for one week, including nearly 80,000 abnormal pieces of data. The data is divided into 1250 samples, each containing 1250 pieces of data, of which 250 samples are used as test sets and 125 samples are validated. GS-IOT system have many faults, the faults show in the following table.

0-5 is the fault caused by the physical level, and 6-8 is the fault caused by the business level. We collect and analyze the main kinds of error data when the OM serial port noise error, EPOS serial port more than 30% overrun error, power supply to EPOS storage module lower than 12V, the ES memory more than 90%, OM blacklist download failed. The normal data is described as happiness, and the fault data as anger, sadness, etc. The details are shown in the following Table 3.

We map different error types to emotions one by one, and code them with one-hot. Since our analysis of errors is based on historical data (logs), the analysis of the log text is similar

TABLE 2. GS-IOT system fault type.

Sequence Number	Fault Name
0	OM serial port noise error
1	EPOS serial port overrun error
2	Insufficient power supply to EPOS storage module
3	ES memory overflow
4	Diaphragm deformation of OM solenoid valve
5	insufficient power supply to the EPOS storage module
6	OM blacklist download failed
7	The cumulative number cannot be read
8	Callback data cannot be uploaded to ES

TABLE 3. Fault name and emotion.

Fault Sequence Number	Fault Name	One-hot	Emotion
0	OM serial port noise error	1 0 0 0 0	Sadness
1	EPOS serial port Overrun error	0 1 0 0 0	Decadent
2	Insufficient power supply to EPOS storage module	0 0 1 0 0	Anger
3	ES memory overflow	0 0 0 1 0	Bother
4	OM blacklist download failed	0 0 0 0 1	Hate

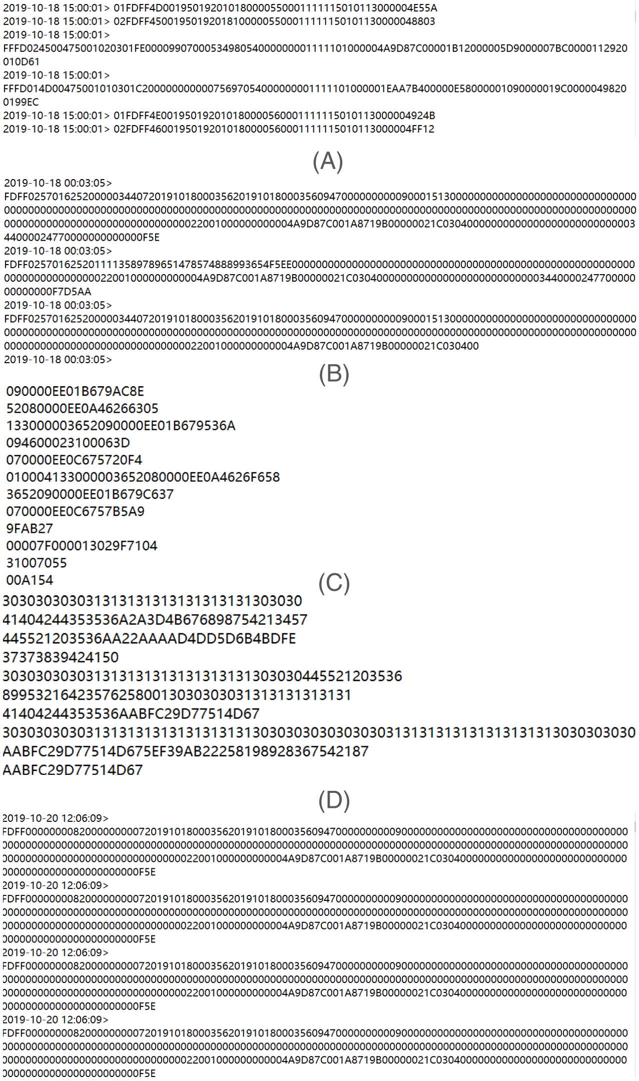
to an emotional analysis of language, and the fault type of the GS-IoT system is reflected from its expression. When a fault occurs, the data will be different from normal. In order to better show the changes of data, the real data of the fault is shown in the figure below:

(A) (B) (C) (D) and (E) shown in Figure 7 are the part of the true data of corresponding errors in Table 2. The message sequence will be disordered in the unknown position when a noise error occurs in the serial port of the OM. We compare this to the system expressing sadness. When the serial port part of the EPOS is overrun error, some messages will be linked together or some part of a message will be spliced to the normal message. We regarde this moment as the system expressing anger. When there is an insufficient power supply to EPOS storage module, the message will be truncated in different positions. We compare this to the emotion decadence. When the ES memory overflow, the message will be presented in a garbled state, which we relate to being bothered. When OM blacklist download failed, the content of the message is repeat thousands of times, it takes up the communication resources of normal message, which leads to the failure of normal service, that's when the system is expressing hatred.

B. EXPERIMENTAL ANALYSIS

1) SIMPLE ANALYSIS OF DATA

We select some normal data from the historical data that is saved in the cloud data center. A part of the normal data is shown in Figure 8:



```
2019-10-18 15:00:01> 01FDFF4D00195019201018000055000111111501011300000 4E55A
2019-10-18 15:00:01> 02FDFF4500195019201810000055000111111501011300000 48803
2019-10-18 15:00:01>
FFFD02450047500102030 1FE00009907000534980540000000001111101000004A9D87C00001B12000005D9000007BC0000112920
010D61
2019-10-18 15:00:01>
FFFD014D00475001010301C200000000000756970540000000001111101000001EAA7B400000E58000001090000019C0000049820
0199EC
2019-10-18 15:00:01> 01FDFF4E0019501920101800005600011111150101130000004924B
2019-10-18 15:00:01> 02FDFF460019501920101800005600011111150101130000004FF12
```

(A)

```
2019-10-18 00:03:05>
FDFF0257016252000003440720191018000356201910180003560947000000000090001513000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000022001000000000004A9D87C001A8719B00000021C03040000000000000000000000000003
44000024770000000000000F5E
2019-10-18 00:03:05>
FDFF025701625201111358978965147857488899365 4F5EE000000000000000000000000000000000000000000000000000000000
000000000000000220010000000000004A9D87C001A8719B00000021C030400000000000000000000000000344000024770000
00000000F7D5AA
2019-10-18 00:03:05>
FDFF0257016252000003440720191018000356201910180003560947000000000090001513000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000022001000000000004A9D87C001A8719B00000021C030400
2019-10-18 00:03:05>
```

(B)

```
090000EE01B679AC8E
52080000EE0A46266305
133000003652090000EE01B679536A
094600023100063D
070000EE0C675720F4
01000413300000365208 0000EE0A4626F658
3652090000EE01B679C637
070000EE0C6757B5A9
9FAB27
00007F000013029F7104
31007055
00A154
```

(C)

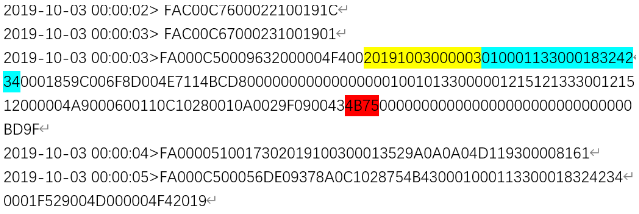
```
3030303030313131313131313131303030
41404244353536A2A3D4B676898754213457
445521203536AA22AAAAD4DD5D6B4BDFE
37373839424150
303030303031313131313131313131303030445521203536
8995321642357625800130303030313131313131
41404244353536AABFC29D77514D67
30303030303131313131313131313130303030303030303131313131313131313131303030303030
AABFC29D77514D675EF39AB2225819892836754218 7
AABFC29D77514D67
```

(D)

```
2019-10-20 12:06:09>
FDFF00000000820000000007201910180003562019101800035609470000000000900000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000220010000000000004A9D87C001A8719B00000021C030400000000000000000000000000000
000000000000000000000000F5E
2019-10-20 12:06:09>
FDFF00000000820000000007201910180003562019101800035609470000000000900000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000220010000000000004A9D87C001A8719B00000021C030400000000000000000000000000000
000000000000000000000000F5E
2019-10-20 12:06:09>
FDFF00000000820000000007201910180003562019101800035609470000000000900000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000220010000000000004A9D87C001A8719B00000021C030400000000000000000000000000000
000000000000000000000000F5E
2019-10-20 12:06:09>
FDFF00000000820000000007201910180003562019101800035609470000000000900000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000220010000000000004A9D87C001A8719B00000021C030400000000000000000000000000000
000000000000000000000000F5E
```

(E)

FIGURE 7. **Fault data. (A) Decadent. (B) Anger. (C) Sadness. (D) Bother. (E) Hate.**



```
2019-10-03 00:00:02> FAC00C7600022100191C↵
2019-10-03 00:00:03> FAC00C67000231001901↵
2019-10-03 00:00:03>FA000C50009632000004F40020191003000003010001133000183242↵
340001859C006F8D004E7114BCD8000000000000000000010010133000001215121333001215
12000004A9000600110C10280010A0029F0900434B7500000000000000000000000000000000
BD9F↵
2019-10-03 00:00:04>FA00005100173020191003000013529A0A0A04D119300008161↵
2019-10-03 00:00:05>FA000C500056DE09378A0C1028754B43000100011330001832 4234↵
0001F529004D000004F42019↵
```

FIGURE 8. **Normal data.**

GS-IoT system interaction process is in the hexadecimal form, where each message has to follow protocol constraints and each field contains a certain amount of information. For example, the yellow field 20191003000003 in the Figure 8 represents the time of the message, the blue field represents the fuel card number, the red field indicates the current cumulative refueling of the oil gun. Therefore, BiLSTM can better locate the fault accurately. However, the data is converted to text type when it is stored in the cloud. Unfortunately, data required by our algorithm SVM-BiLSTM is

TABLE 4. Parameter setting in SVM.

Parameter	Meaning	Value/Formula
C	Penalty Parameters	10
Kernel	Rbf	rbf
Sigma	Parameter of rbf	1/n_features
l_m	Maximum number of iterations	No limit
Err	Threshold of error	0.001

numeric type. Hence, a language library has been established that uses the word breaker to divide words, and a one-to-one index table is set up to convert each piece of data into a number by finding the corresponding index.

2) EXPERIMENTAL

To verify the diagnosis strategy proposed in this article, KBS, SVM, BiLSTM, GRU, and CNN are used to analyze and process the data, and the diagnosis results are compared. In KBS, the fault decision tree is constructed, and the fault types are identified by the expert knowledge. For example, in insufficient power supply to EPOS storage module error, the data length is obviously shorter and most of them are normal message truncation. SVM uses a one-vs-one multi-classification method in this experiment. We use analytical to select parameters, the parameters of SVM are as follows:

The parameters of CNN are as follows: the first convolution layer have a total of 32∗3∗3 convolution kernels, the first pooling layer had 3∗3 pooling kernels, the second convolution layer have a total of 64∗3∗3 convolution kernels, and the second pooling layer have a 3∗3 pooling kernel. Finally, the data is input into the full connection layer after flattening. The batch size is 20, the number of iterations was 200, and the classifier is softmax, Using Adam as an optimizer. The parameters of BiLSTM are as follows: nesting 5-layer BiLSTM network, one block of LSTM with N cells, and finally input to the full connection layer using softmax as a classifier.

In SVM-BiLSTM, the text data is transformed into data that the model can train according to the self-built dictionary. Since length of each data is different, we use pads that are 0 to make up the short data to the same length. Finally get the matrix $X_{n\times m}$. There are n pieces of collected samples, each of which has m attributes. The whole process is shown in equation (12).

$$\begin{matrix} X_{11} & X_{12} & \cdots \\ X_{21} & X_{22} & \cdots \\ \vdots & \vdots & \vdots \\ X_{\mathrm{n}1} & X_{\mathrm{n}2} & \cdots \end{matrix} \Rightarrow X_{n\times m} = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1m} \\ X_{21} & X_{22} & \cdots & X_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ X_{\mathrm{n}1} & X_{\mathrm{n}2} & \cdots & X_{\mathrm{nm}} \end{bmatrix} \tag{12}$$

TABLE 5. Diagnosis results for different methods.

Methods	Accuracy (%)	Time(s)
KBS	81.20	——
SVM	92.41	1447.82
GRU	95.52	1710.91
CNN	97.89	3610.17
BiLSTM	99.93	11132.43
SVM- BiLSTM	99.12	1833.28

Since the data have different dimensions, we normalize all the original input. The normalized attribute matrix of $X_{n\times m}$ is $Y_{n\times m}$ whose entry at (i, j), $i \in \{1, 2, \dots, n\}$ and j ∈ {1, 2, . . . , m} is:

$$\mathrm{y}_{ij} = \frac{x_{ij}}{\sum\limits_{i=1}^{n} x_{ij}} \tag{13}$$

After (13), the sum of all components in a column is one. After the training samples are obtained, the SVM model is established for preliminary feature extraction. The abnormal data after classification is inputted to BiLSTM.

We use Accuracy, Precision, Recall and F-score to make a comprehensive comparison between the proposed one and tradition model. And the calculation formula is as follows:

$$\begin{aligned} \text{Accuracy} &= \frac{N_N + F_F}{N_N + N_F + F_N + F_F}\% \\ \Pr\text{ecision} &= \frac{F_F}{N_F + F_F}\% \\ Recall &= \frac{F_F}{F_N + F_F}\% \\ F - \text{score} &= \frac{2}{\frac{1}{\Pr ecision} + \frac{1}{Recall}}\% \end{aligned} \tag{14}$$

N_N means normal samples are predicted as normal ones. N_F means normal samples are predicted as abnormal ones. F_N are the abnormal samples are predicted as normal ones. F_F are the abnormal samples are predicted as abnormal ones. Accuracy represents the number of correct classifications in all experiments. Precision is how many real results are positive in the prediction results. And Recall indicates how many of the real results are positive classes. F-score comprehensively considers precision and recall indicators.

The different results of accuracy are shown in the following table.

As can be seen from the table above, SVM-BiLSTM is outstanding in terms of overall performance, with the accuracy of KBS, SVM, GRU, CNN, and BiLSTM being 81.20%, 92.41%, 95.52% 97.89%, and 99.93%. BiLSTM has the highest accuracy of all methods, but it is highly flawed in terms of efficiency. KBS have the lowest accuracy. The use of GRU is efficient, but it does not meet the accuracy requirements. SVM-BiLSTM has a great advantage in terms of overall performance and ensure accuracy and efficiency. In addition, SVM-BiLSTM is much more accurate and computationally efficient than CNN. Table 6 shows the average Precision, Recall and F-sore of different methods.

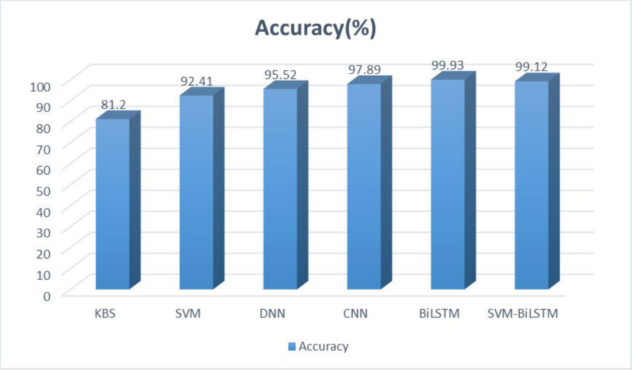


FIGURE 9. Accuracy figure.

TABLE 6. Average precision, recall and F-score of different methods.

Methods	Precision	Recall	F-score
KBS	0.85	0.81	0.83
SVM	0.92	0.88	0.87
GRU	0.93	0.91	0.91
CNN	0.98	0.96	0.97
BiLSTM	1	0.99	1
SVM-BiLSTM	0.99	0.98	0.99

TABLE 7. Each fault type diagnosis results for different methods.

Methods	0	1	2	3	4
KBS	0.76	0.68	0.86	0.92	0.83
SVM	0.85	0.81	0.94	0.99	0.86
GRU	0.91	0.87	0.96	0.95	0.91
CNN	0.96	0.95	0.99	0.98	0.99
BiLSTM	0.99	1	1	1	1
SVM-BiLSTM	0.99	1	1	1	1

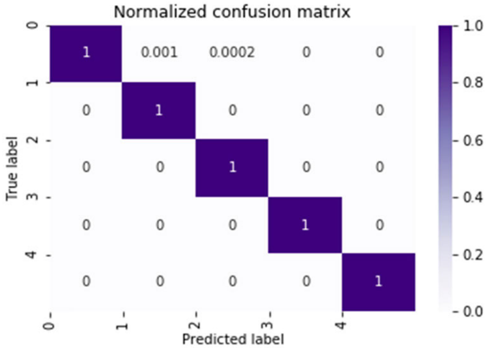
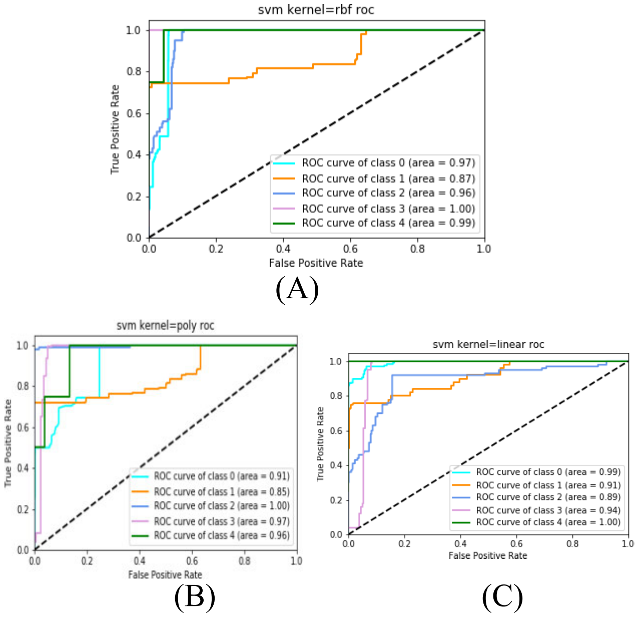


FIGURE 10. SVM-BiLSTM Confusion matrix.

The experimental results in precision, recall and F-score from these methods also lead to the same conclusion that the proposed method well outperforms its peers.

To better show the classification effect of the algorithm, Table 7 shows the fault detection accurate rate of each fault detection, and SVM-BiLSTM confusion matrix is drawn, the result is shown in Figure 10.

We conclude by analyzing Table 5 that the classification results of KBS and SVM on category 0 and 1 are not ideal. In addition, the BiLSTM has the best classification structure, the results of GRU and CNN are also acceptable. The horizontal axis of the SVM-BiLSTM Confusion matrix represents the predicted result, and the vertical axis represents the real label. The diagonal represents the corresponding degree between the predicted result and the real result. The larger the value is, the darker the color is, and the more accurate the prediction of the model is. The color corresponds to the value on the right side of the graph. Accuracy values range from 0-1. 0-4 on axis refers to 5 fault types.



(A)

(B)

(C)

FIGURE 11. ROC curves of different kernel functions. (A)rbf. (B)poly. (C)linear.

3) PARAMETER OPTIMIZATION

To train a better model, we need to optimize the model's key parameters C and kernels of SVM. We use different C values and kernels in SVM to verify the effect. In the first phase, we choose a sample with 20,000 rows of data. In the second phase, we use different kernel functions and keep the other parameters the same. We draw Receiver Operating Characteristic (ROC) curves of different kernel functions to observe their effects and the ROC curves are as follows.

The abscissa of ROC curve is false positive rate and the ordinate is true positive rate. We use 5 colors to represent 5 different kinds. The closer the ROC curve is to the upper left corner, the better the classification effect is. It can be seen from Figure 12 that rbf kernel functions perform best, followed by linear kernel functions and poly kernel functions.

The number of C are set as follows: $C \in \{0.1, 0.5, 1, 5, 10\}$. A full-factorial analysis needs 3 × 5 experiments. We adopt the orthogonal arrays so as to decrease the number of experimental runs [31]. We use reasonable combination of kernel function and C and observe its accuracy. The result show in Table 8.

Different values of parameter C combine with different kernels have different effect on accuracy of the model.

TABLE 8. Diagnosis results for different c and kernel.

C	kernel	Accuracy	Accuracy std
0.1	rbf	0.827	+/-0.015
5	rbf	0. 927	+/-0.010
10	rbf	0.934	+/-0.033
0.5	linear	0.872	+-0.031
1	linear	0.882	+-0.093
10	linear	0.896	+-0.051
0.1	Poly	0.853	+-0.063
5	Poly	0.896	+-0.083
10	poly	0.921	+-0.104

TABLE 9. Diagnosis results for different layers.

Number of layers	Accuracy (%)	Computer Time(s)
1	92.70	502
2	94.19	576
3	96.76	861
4	98.48	1028
5	99.52	1420
6	99.63	1740

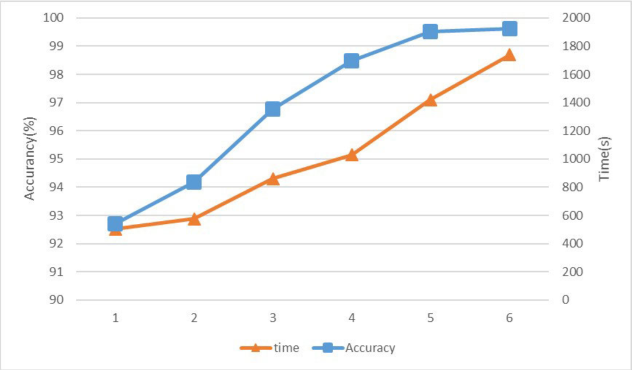


FIGURE 12. Diagnosis results for different layers.

Analysis of the data in the above table shows that the accuracy is the highest when the kernel is rbf and C=10. Hence, we adopt rbf as the kernel function and C=10 in SVM-BiLSTM.

To compare the impact of different layers of BiLSTM on the accuracy, we apply different layers of 1-6 to diagnose faults in BiLSTM, with a sample with 100,000 rows of data. The average accuracy of the diagnosis and the time spent in calculation are shown in Table 9, and the trend is shown in Figure 13.

An increase in the number of network layers increase the accuracy and the time required for calculation. The accuracy of the one layer is 92.70%, the accuracy of the three layers is 96.76%, and that of 5-6 layers is more than 99%. However, the diagnosis time increases fast from 502(s) on the first layer to 1740(s) on the sixth layer, and the diagnosis efficiency declines. The 5-layer BiLSTM is the most suitable.

V. CONCLUSION

This article proposes a combination of SVM and BiLSTM for the fault detection of the GS-IoT system by analogizing the OM and the ES communication to emotional expression. Through an emotional analysis of the data, we analyzed the data in real time and classified the types of faults. To cope with the complex and deep structure of the BiLSTM algorithm, we started with entering the data into the SVM to distinguish normal and fault and then employed the data into the BiLSTM to accurately locate the type of failure. The accuracy of the fault detection method proposed in this article reaches 99.12%, which meets the requirements of the gas station for fault detection accuracy.

This work can detect the occurrence of faults in the IoT system in gas station; however it is difficult to detect unknown faults [35], [36], such as oil leakage from OT, EPOS keyboards input error due to high temperature, insufficient power supply of OM solid state relay, etc. When the data itself is not wrong, but lost or redundant, it is difficult to use the current method to detect. This is to be addressed as our future work.

REFERENCES

[1] H. Chen, G. Jiang, K. Yoshihira, and A. Saxena, "Invariants based failure diagnosis in distributed computing systems," in *Proc. 29th IEEE Symp. Reliable Distrib. Syst.*, Oct. 2010, pp. 160–166.

[2] C. G. Siontorou, F. A. Batzias, and V. Tsakiri, "A knowledge-based approach to online fault diagnosis of FET biosensors," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 9, pp. 2345–2364, Sep. 2010.

[3] C. WenBin, L. XiaoLing, H. ChangJiu, and L. YiJun, "Knowledge base design for fault diagnosis expert system based on production rule," in *Proc. Asia–Pacific Conf. Inf. Process.*, Jul. 2009, pp. 117–119.

[4] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part II: Fault diagnosis with knowledge-based and hybrid/active approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3768–3774, Jun. 2015.

[5] D. Jung and C. Sundstrom, "A combined data-driven and model-based residual selection algorithm for fault detection and isolation," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 2, pp. 616–630, Mar. 2019.

[6] Y. Xu, Y. Sun, J. Wan, X. Liu, and Z. Song, "Industrial big data for fault diagnosis: Taxonomy, review, and applications," *IEEE Access*, vol. 5, pp. 17368–17380, 2017.

[7] J. P. Matsuura and T. Yoneyama, "Learning Bayesian networks for fault detection," in *Proc. 14th IEEE Signal Process. Soc. Workshop Mach. Learn. Signal Process.*, Oct. 2004, pp. 133–142.

[8] W. Zhang, H. Zhang, J. Liu, K. Li, D. Yang, and H. Tian, "Weather prediction with multiclass support vector machines in the fault detection of photovoltaic system," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 3, pp. 520–525, 2017.

[9] T. Hong, H. Li, and F. Zhong, "Adaptive two-class C-support vector machine algorithm for turbopump fault detection," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Dec. 2011, pp. 779–783.

[10] V. C. Ferreira, R. C. Carrano, J. O. Silva, C. V. N. Albuquerque, D. C. Muchaluat-Saade, and D. Passos, "Fault detection and diagnosis for solar-powered wireless mesh networks using machine learning," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 456–462.

[11] S. Zidi, T. Moulahi, and B. Alaya, "Fault detection in wireless sensor networks through SVM classifier," *IEEE Sensors J.*, vol. 18, no. 1, pp. 340–347, Jan. 2018.

[12] Z.-Y. Qi, Q.-H. Li, G.-X. Yi, Y.-G. Xie, and H.-T. Dang, "Online fault detection of HRG based on an improved support vector machine," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Jul. 2013, pp. 316–319.

[13] H. Zhang, Y. Li, Z. Lv, A. K. Sangaiah, and T. Huang, "A real-time and ubiquitous network attack detection based on deep belief network and support vector machine," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 3, pp. 790–799, May 2020.

[14] S. Teng, N. Wu, H. Zhu, L. Teng, and W. Zhang, "SVM-DT-based adaptive and collaborative intrusion detection," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 108–118, Jan. 2018.

[15] K. Li, R. Zhang, F. Li, L. Su, H. Wang, and P. Chen, "A new rotation machinery fault diagnosis method based on deep structure and sparse least squares support vector machine," *IEEE Access*, vol. 7, pp. 26571–26580, 2019.

[16] Q. Kang, L. Shi, M. Zhou, X. Wang, Q. Wu, and Z. Wei, "A distance-based weighted undersampling scheme for support vector machines and its application to imbalanced classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4152–4165, Sep. 2018.

[17] P. Zhang, S. Shu, and M. Zhou, "An online fault detection model and strategies based on SVM-grid in clouds," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 2, pp. 445–456, Mar. 2018.

[18] J. Cui, G. Li, M. Yu, L. Jiang, and Z. Lin, "Aero-engine fault diagnosis based on kernel principal component analysis and wavelet neural network," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2019, pp. 451–456.

[19] O. Gecgel, S. Ekwaro-Osire, J. P. Dias, A. Serwadda, F. M. Alemayehu, and A. Nispel, "Gearbox fault diagnostics using deep learning with simulated data," in *Proc. IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, Jun. 2019, pp. 1–8.

[20] T. Benkedjouh, N. Zerhouni, and S. Rechak, "Deep learning for fault diagnosis based on short-time Fourier transform," in *Proc. Int. Conf. Smart Commun. Netw. Technol. (SaCoNeT)*, Oct. 2018, pp. 288–293.

[21] T. D. Pham, K. Wardell, A. Eklund, and G. Salerud, "Classification of short time series in early parkinsons disease with deep learning of fuzzy recurrence plots," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 6, pp. 1306–1317, Nov. 2019.

[22] E. Principi, D. Rossetti, S. Squartini, and F. Piazza, "Unsupervised electric motor fault detection by using deep autoencoders," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 2, pp. 441–451, Mar. 2019.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1–15.

[25] L. Li, G. Liu, L. Zhang, and Q. Li, "Deep learning-based sensor fault detection using S-long short term memory networks," *Struct. Monitor. Maintenance*, vol. 5, no. 1, pp. 51–65, Mar. 2018.

[26] D. Park, S. Kim, Y. An, and J.-Y. Jung, "LiReD: A light-weight real-time fault detection system for edge computing using LSTM recurrent neural networks," *Sensors*, vol. 18, no. 7, p. 2110, Jun. 2018.

[27] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[28] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[29] N. Enshaei and F. Naderkhani, "Application of deep learning for fault diagnostic in induction Machine's bearings," in *Proc. IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, Jun. 2019, pp. 1–7.

[30] A. Ko and A. Chowdhury, "Avoidance of model re-induction in SVM-based feature selection for text categorization," in *Proc. Int. Joint Conf. Artif. Intell*, 2007, pp. 889–894.

[31] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 601–614, Feb. 2019.

[32] B. Blanco-Filgueira, D. Garcia-Lesta, M. Fernandez-Sanjurjo, V. M. Brea, and P. Lopez, "Deep learning-based multiple object visual tracking on embedded system for IoT and mobile edge computing applications," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5423–5431, Jun. 2019.

[33] F. Banaie, J. Misic, V. B. Misic, M. H. Y. Moghaddam, and S. A. H. Seno, "Performance analysis of multithreaded IoT gateway," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3143–3155, Apr. 2019.

[34] F. Samie, L. Bauer, and J. Henkel, "From cloud down to things: An overview of machine learning in Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4921–4934, Jun. 2019.

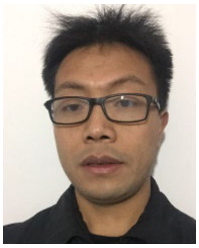
[35] X. Wang, X. Liu, and Y. Li, "An incremental model transfer method for complex process fault diagnosis," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 5, pp. 1268–1280, Sep. 2019.

[36] T. Bian and Z.-P. Jiang, "Reinforcement learning for linear continuous-time systems: An incremental learning approach," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 2, pp. 433–440, Mar. 2019.

[37] Z. Zhou, H. Liao, B. Gu, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, "Robust mobile crowd sensing: When deep learning meets edge computing," *IEEE Netw.*, vol. 32, no. 4, pp. 54–60, Jul. 2018.

YAO JIAHAO received the B.S. degree in information and electronic engineering from Zhejiang Gongshang University, Hangzhou, China, in 2018. He is currently a Graduate Student with the School of Information and Electronic Engineering, Zhejiang Gongshang University. His research interests include the Internet of Things and deep learning.

XIAONING JIANG received the M.E. degree in electronic engineering from Hangzhou Dianzi University, Hangzhou, China, in 1993, and the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, in 2000. He is currently an Associate Professor and a Senior Engineer with Zhejiang Gongshang University and the Vice Dean of the IoT Research Institute, Zhejiang Gongshang University. His research interests include applied computer systems, network and information security, the industrial IoT, visual analytics, and financial technology. He has published over 30 research articles and ten invention patents.



SHOUGUANG WANG (Senior Member, IEEE) received the B.S. degree in computer science from the Changsha University of Science and Technology, Changsha, China, in 2000, and the Ph.D. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2005.

He joined Zhejiang Gongshang University, in 2005, where he is currently a Professor with the School of Information and Electronic Engineering, the Director of the Discrete-Event Systems Group, and the Dean of the System Modeling and Control Research Institute. He was a Visiting Professor with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA, from 2011 to 2012. He was a Visiting Professor with the Electrical and Electronic Engineering Department, University of Cagliari, Cagliari, Italy, from 2014 to 2015. He was the Dean of the Department of Measuring and Control Technology and Instrument, from 2011 to 2014. He is an Associate Editor of IEEE ACCESS and the IEEE/CAA JOURNAL OF AUTOMATICA SINICA.

KELEI JIANG (Student Member, IEEE) is currently pursuing the bachelor's degree with the University of Washington, Seattle, USA. She has named to the Dean's List in the College of Arts and Sciences, from 2018 to 2020. Her research interests include the application of organizational informatics, human–computer interaction, computer art, and data visualization engineering.

XIAOHAN YU received the B.E. degree in electronic science and technology from the Harbin Institute of Technology, Weihai, in 2009, and the Ph.D. degree from the Department of Electrical Engineering, Korea University, Seoul, in 2015. He is currently an Associate Professor with the School of Information and Electronic Engineering, Zhejiang Gongshang University. His current research interests include compressive sensing, wireless sensor networks, and convex optimization.