IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 October 2023, accepted 1 December 2023, date of publication 7 December 2023,
date of current version 14 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3340527

RESEARCH ARTICLE

Research on the Classification Method of CTC Alignment Joint-Test Problems Based on RoBERTa-wwm and Deep Learning Integration

NING QIN
The Center of National Railway Intelligent Transportation System Engineering and Technology, China Academy of Railway Sciences Corporation Ltd., Beijing 100081, China

e-mail: qning1980@sina.com

This work was supported in part by the Center of National Railway Intelligent Transportation System Engineering and Technology, China Academy of Railway Sciences Corporation Ltd., under Grant RITS2022KF06; and in part by the Foundation of China Academy of Railway Sciences Corporation Ltd. under Grant 2022YJ065.

ABSTRACT A text classification method based on RoBERTa-wwm and deep learning integration is proposed for fast classification of the massive unstructured problematic text data generated during the alignment joint-test of CTC (Centralized Traffic Control). Firstly, 10 common types of problems were summarized based on the statistical results of the problems from the alignment joint-test of CTC generated between 2011 and 2021; Secondly, in the text classification model, the pre-trained model RoBERTa-wwm is used to capture the semantic features of words in the problem text; Building an integrated model for deep learning based on BiLSTM-BiGRU and CNN to fully learn the deep hidden information in texts; Combining the principles of BiLSTM and BiGRU based on combined weight calculation to maximize performance; Normalization by Softmax function yields classification results for the CTC JCT problem. Finally, experimental validation is performed using data from CTC alignment joint-test problems generated during the last decade, The experimental results show that compared with several existing typical pre-trained models and classifier combinations, the proposed method in this paper achieves better results in terms of accuracy, precision, recall and F1 values, reaching 0.9317, 0.9322, 0.9317 and 0.9318, respectively, and the final Loss is only 0.24.

INDEX TERMS RoBERTa-wwm, deep learning, BiLSTM-BiGRU, alignment joint-test, text classification.

I. INTRODUCTION

CTC (Centralized Traffic Control) system, as an important part of high-speed railroad signal system, is an important guarantee for high-speed railroad to achieve high speed, high density, high efficiency and safe and punctual operation, and plays a very important role in high-speed railroad train transportation command and signal control system, which is the core carrier of signal equipment service transportation production [1]. Previously, a large amount of multifarious and heterogeneous problem data has been accumulated during a large amount of alignment joint-test works of the centralized dispatching system. Among them, the CTC alignment joint-test data stored in text form carries a lot of important safety information, mainly based on the description of the corresponding phenomena when CTC system problems occur. Therefore, text classification algorithms based on text mining technology and artificial intelligence are needed. In the context of railroad big data and intelligent railroad construction, the use of text mining technology and deep learning algorithms to classify CTC system problems intelligently, help field maintenance personnel to quickly locate the fault location according to the CTC problem description and give the corresponding maintenance and repair decisions in a timely manner, which has far-reaching significance for further improving the safety and security level of high-speed railway operation [2].

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

The intelligent classification method for the CTC alignment joint-test problem based on text data consists of two main tasks, the first of which is to obtain its feature vector representation by feature extraction of the problem text data: in order to enable the computer to better understand human language, The One-Hot encoding proposed in the literature [3] numbers each subword sequentially (denoted by 0 and 1), however, this numbering method tends to generate vectors with too much dimensionality and sparse representation. With the rapid development in the field of NLP (Natural Language Processing), the distributed word vector representation of text is gradually replacing the One-Hot coding approach, which maps words into a real vector with small dimensions and dense values, with good computability, while avoiding the difficulty of manual feature selection and the problem of sparse vector representation. The common distributed word vector representation methods mainly include TF-IDF (Term Frequency-Inverse Document Frequency) [4], Word2vec algorithm [5], ELMO (Embeddings from Language Models) algorithm [6], and so on, but due to the deep neural networks have a large number of parameters and are trained on a limited data set, their networks tend to be overfitted. Therefore, Vaswanid et al. [7] proposed Transformer, a deep learning model, to address the slow training, inefficiency and overfitting problems of most deep learning networks [8]; At the same time, the Transformer is combined with the Self-Attention mechanism, thus achieving fast parallelism. Since then the pre-trained language model PLM (Pretrained Language Model) has entered a phase of rapid development and various large-scale PLMs including BERT (Bidirectional Encoder Representation from Transformers) [9] and GPT (Generative Pre-training Transformer) [10] have been highly successful in various NLP tasks and have spawned several different improved versions and multilingual PLMs.

The presence of a large number of complex semantic features and contextual structural relationships within the text largely increases the difficulty of classifying them accurately. To improve the classification, traditional machine learning-based methods [11], [12], [13], [14] have been implemented mainly by improving feature engineering [15]. However, the feature engineering construction of machine learning is done manually, and the ability of machine learning to handle large-scale data is slightly insufficient, and it cannot solve the problem of highly sparse feature vectors. Compared with machine learning, deep learning has obvious advantages in short text classification: first, it does not need to manually construct the initial features that can be incorporated into the model training, and can realize the end-to-end classification method; second, deep learning can obtain a low-dimensional dense feature representation of short text, and can fully extract the front and back text information as well as deeper semantic relationships; third, deep learning models can not only cope with large amounts of data, but also discover new features from them as the amount of data expands. Based on these three advantages, deep learning has gradually replaced machine learning as the mainstream method in the field of short text classification research, and the most representative deep learning networks are CNN (Convolutional Neural Networks) and RNN (Recurrent Neural Network). The former can extract local features within the text by convolution operation, and later take pooling operation to filter out the most critical information; the latter is good at extracting information in long sequence data. Although CNN and RNN have achieved good results in the field of English text classification, in Chinese text classification, due to the complex and diverse structure of Chinese characters and the inclusion of many homonyms and homophones, therefore traditional single-text classification models have difficulty in dealing with this problem, and considering that RNNs allocate equal attention to all contexts, a single deep learning model cannot adequately learn text features, and so on [16]. In summary, this paper proposes a text classification method based on RoBERTa (Robust optimize BERT approach)-wwm (whole word masking) and deep learning integration, and the main contributions are presented in the following 3 points:

1) Select the signal CTC system problems found during the alignment joint-test of high-speed railroads in the past ten years from 2011 to 2021, and organize, summarize and analyze the data according to the function and fault characteristics of each device in the CTC system to obtain 10 key CTC system problem types.
2) The pretrained model RoBERTa-wwm learns the semantic features of words in short text by using pre-training method as whole word masking: RoBERTa uses more and larger text data for training than previous pretrained models, thus improving the model performance; whole word masking adopts a stricter masking strategy, replacing all words in the input text with [MASK], so that the information in the text data can be better utilized.
3) A deep learning integrated network based on BiLSTM (Bi-directional Long Short-Term Memory)-BiGRU (Bidirectional Gated Recurrent Unit) and CNN is constructed as a downstream model to fully learn the local feature information of the text with CNN; adopting BiLSTM-BiGRU based on combined weights to learn the pre and post text information, while combining the two with the features output from RoBERTa-wwm into the Softmax function, retaining the original information of RoBERTa-wwm, which in turn enables the model in this paper to learn the problem text features of CTC system in a comprehensive way.

II. RELATED WORK

A. PRETRAINED LANGUAGE MODEL

The PLM [17] can be pre-trained on a large-scale textual pre-base by self-supervised learning and can be fine-tuned on a specified domain dataset to complete the training. Devlin [9] first proposed the pre-training model BERT in 2018, and the two training methods used in this model are MLM (Mask Language Model) and NSP (Next Sentence Prediction, NSP) respectively, and has achieved excellent performance in NLP tasks in most domains, and has since spawned a variety of improved versions of the BERT model, for example, ALBERT (A Lite BERT) [18] can share parameters across layers with only one-fifth of the number of model internal parameters of BERT; RoBERTa [19] uses a broader dataset for training while replacing static masks with dynamic masks; XLnet [20] borrows the idea of Transformer-XL [21]. BERT is prone to lose some information when dealing with extra-long sequences, but Transformer-XL adopts a segmentation loop mechanism, which can effectively extract the information in extra-long sequences. Radford proposed the GPT [22] model in 2018, whose pre-training consists of two phases: one is unsupervised pre-training to obtain high-volume PLMs by learning on a large text corpus; the other is supervised fine-tuning to adapt the model to discriminative tasks with labeled data. Radford subsequently developed GPT-2 in 2019 [23], [24], which still uses a GPT-like unidirectional transformer model, but with much improved training data both in terms of quantity quality and extensiveness, an increased number of network parameters, and no need to fine-tune it using domain-specific data in most cases.

In recent years, pre-trained language models have developed equally rapidly for Chinese pre-training. In 2019 Baidu released the Chinese pre-training model ERNIE (Enhanced Language Representation with Informative Entities) [25], which unifies the modeling of grammatical structure, lexical structure and semantic information in Chinese text training data to improve the representation of Chinese grammar; in 2021, Harbin Institute of Technology and KDDI jointly developed a Chinese pre-training model BERT-wwm-base [26] based on the Google BERT-base [27] model, which uses the whole word masking to complete the pre-training work on large-scale Chinese corpus.

B. DEEP LEARNING BASED CLASSIFIER MODEL

The deep learning-based classifier model serves as a downstream model for the text classification task, allowing further refinement of text features and laying the foundation for subsequent output of classification probability values. The commonly used text classification models based on machine learning mainly include the plain Bayesian classifier NBC (Naive Bayes Classifier) [28], decision tree DT (Decision Tree) [29], support vector SVM (Support Vector Machine) [30], and integrated classification models Bagging [31], Boosting [32] and so on. Machine learning cannot meet the demand of increasing data processing and cannot learn deeper features from data by itself, so deep learning-based text classification methods are gradually becoming a hot research topic in NLP today. The common text classification models based on deep learning mainly include CNN [33], [34], [35], RNN [8], [36], [37] and their improved models. Literature [38] proposed CNN as a text classification model to extract high level text features by obtaining locally sensitive information from text represented by word vectors through convolution and pooling operations; literature [39], [40] used BiLSTM neural networks as downstream models in text classification tasks, further incorporating contextual information on top of the generated feature vectors, and achieved good classification results; In the literature [41], BiGRU (Bi-directional Gate Recurrent Unit) is proposed as a classifier model, and this neural network synthesizes the forgetting gate and the input gate into a single update gate based on BiLSTM, which can still achieve the training effect of BiLSTM while reducing the structural complexity; Tencent AI-lab [42] proposed DPCNN (Deep Pyramid Convolutional Neural Networks) in ACL2017 to deepen the network by multi-layer convolution and pooling operations to extract long-range text dependencies; In the literature [43], [44], a CNN-BiLSTM parallel feature extraction model was proposed to extract local feature information and contextual feature information of Chinese text, respectively; Xu et al. [45] improved on the literature [46] by introducing BiGRU to extract contextual information of words, and then using an attention mechanism to dynamically extract a set of concepts related to the context and then aggregated to obtain the concept representation; Hao et al. [47] used BiLSTM to obtain word and character representations separately and integrate the two through an attention mechanism, which was input to CNN to extract features for classification.

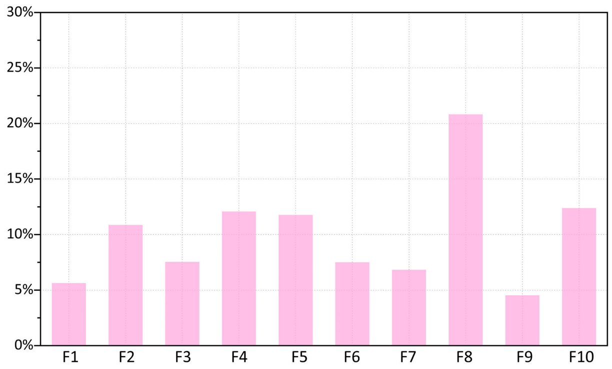


FIGURE 1. Percentage of various types of problems in CTC system.

III. PRELIMINARY WORK

A. INTRODUCTION TO THE DATASET FOR CTC PROBLEMS

In this paper, according to the function and fault characteristics of CTC system, the signal CTC system problems found during the alignment joint-test of high-speed railroads from 2011 to 2021 are counted, classified and summarized, and a total data set of 43,142 data is obtained, and 10 types of CTC alignment joint-test problems are summarized, which are ATO, CTCS3 level display, train number tracking,

TABLE 1. Examples of CTC system data for various types of problems.

Serial number	Description of the fault occurrence phenomenon	Fault classification
F1	The downbound train was near relay 40 and the train number was lost, causing the Southeast Line House to fail to automatically trigger the approach.	ATO
F2	During the inter-bureau operation test of signal test trains (CTCS3 class mode control trains), after passing through the RBC jurisdictional demarcation points of the two bureaus, the CTCS3 class car number italic and mobile authorized light band logo display on the station map interface of the CTC systems of the two adjacent bureaus disappeared.	CTCS level 3 display
F3	The downbound train was near relay 40 and the train number was lost, causing the Southeast Line House to fail to automatically trigger the approach.	Train number tracking
F4	The D55345 test train from Zhengzhou Bureau's controlled area to Lumiao Station did not receive the 3G pick-up approach notice from Lumiao Station.	Approach Preview
F5	April 19-April 20, 2016, D55608/5: 2:48 p.m. The approach failed to be triggered at Liuhe Station, and the dispatcher triggered the approach after reissuing the plan.	Inlet Trigger
F6	Ningan passenger station and Hefu high-speed railway station are adjacent to each other. Ningan passenger station can receive plans from Hefu high-speed railway station, while Hefu high-speed railway station cannot receive plans from Ningan passenger station.	Adjacent station information interaction

approach preview, approach trigger, neighboring station information interaction, wireless regulation, speed limit, operation chart and station display, which are indicated by F1-F10 in the following. The proportion of the data volume of each category in the total data set is shown in Figure 1. It can be seen that the speed limit category has the largest proportion of problems, the neighboring station information interaction category has the smallest proportion, and four types of problems, CTCS3 level display category, approach preview category, approach trigger category, and station display category, also have a high proportion.

The text data representation of the problems generated during the CTC alignment joint-test is shown in the table 1, consisting of the description of the phenomenon corresponding to the problem and the type to which the problem belongs. In the subsequent experiments, the total 43142 fault data sets were divided into training set, development set and test set, and 35142 of them were selected as the training set data for the fitting of the subject network; 3000 data for the development set are used to tune parameters, select features, and make other decisions about the learning algorithm; 5000 data for the test set are used to test and evaluate the model, with 500 questions for each category.

B. RoBERTa-wwm PRE-TRAINING PRINCIPLE

RoBERTa-wwm model is based on the evolution of BERT model, in order to extract deeper bi-directional relationships in Chinese utterances, the core architecture of BERT model is composed of 12 layers of bi-directional Transformer Encoder, The MHA (Multi-headed Self-attention) mechanism is used to convert the distance between two words at any position in a sentence to 1, and the masked language model NSP and the next sentence prediction NSP are used as pre-training targets. MLM randomly selects 15% of the fields in the input statement, and replaces the selected Token with 80% probability as "[MASK]", 10% probability as random words, and the remaining 10% probability to keep the original words unchanged in each training, so as to improve the feature representation ability and generalization ability of the model. The basic idea of NSP is to model the relationship between two sentences by putting them together and using the learned information to determine that the two sentences are adjacent and in normal order in the original text.

RoBERTa-wwm in the original BERT model mainly made the following 3 improvements: 1) modified the static mask to dynamic mask, when the BERT model training data, each sample only once random mask, in each training the position of the mask is the same, while the dynamic mask before each training to dynamically generate a mask, so that the model can learn more utterance patterns; 2) remove the NSP Loss, which improves the model efficiency to some extent; 3) use a larger dataset for pre-training, and use Byte Pair Encoding (BPE) to process the text data. And the RoBERTa-wwm model combines the advantages of RoBERTa model and wwm by changing the sample generation strategy in the pre-training stage to a full word mask wwm strategy, but removing the dynamic mask.

IV. THE CLASSIFICATION MODEL FOR CTC ALIGNMENT JOINT-TEST PROBLEMS BASED ON RoBERTa-wwm AND DEEP LEARNING INTEGRATION

The structure of the proposed classification model based on RoBERTa-wwm and deep learning integration for the CTC JCT problem is shown in Figure 2. The model mainly consists of RoBERTa-wwm, CNN, BiLSTM-BiGRU and Output layers. Transformation of unstructured fault text into structured word vector representation by Embedding in RoBERTa-wwm as an upstream task of the model in this paper; In the downstream task, CNNs with three convolutional kernels of different sizes are used to extract local feature information in different dimensions, and contextual information is extracted by a BiLSTM-BiGRU network based on combined weights, after which this local feature information, contextual information and the original information output by RoBERTa-wwm are combined and input to the Output layer; The maximum probability value is output by Softmax in the Output layer as the classification result.

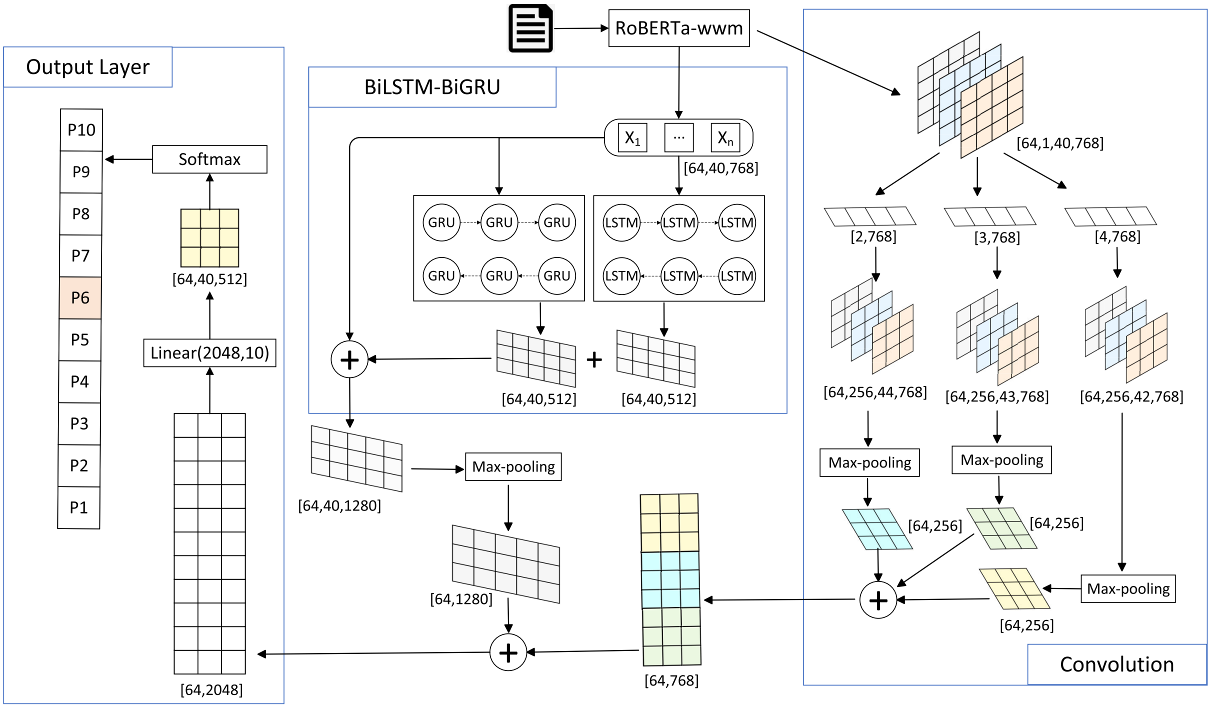


FIGURE 2. Classification model structure of CTC JCT problem based on RoBERTa-wwm and deep neural network.

A. CALCULATION PRINCIPLE OF RoBERTa-wwm

The schematic diagram of the RoBERTa-wwm calculation process is shown in Figure 3. Firstly, BertTokenizer converts the input Sentence into the Assemble format defined by BertModel, i.e., adding "[CLS]", "[SEP]" and "[PAD]" to the sentence, after that, Position_Embeddings, Token_Type_Embeddings and Word_Embeddings are output by BertTokenizer, whose internal word embedding feature vector dimension is 768, and the value of the direct sum of the three is used as the input of Transformer Encoder. Multi-Head Self-Attention extracts feature information of different dimensions with 12 Attention Heads; the role of Feed Forward is to add a nonlinear mapping between each encoder and decoder layer by adopting the activation function Gelu, and the feature vectors are outputted by 12 Transformer Encoders. Hidden State with structure [Padding Size, 768] is obtained, after which two outputs, PoolOut and MASK Loss, are obtained from Hidden State, in which Mask Loss predicts the words that are MASKed, and since the full word mask wwm is used, the model selects all words of the same type that are MASKed to participate in the computation of the loss function.

Since the proposed model sets Batch_Size as 64, Padding_Size as 40, and RoBERTa-wwm outputs word vectors with dimension 768, RoBERTa-wwm transforms the unstructured fault text into feature vectors with the structure [64, 40, 768] and feeds them to the downstream model.

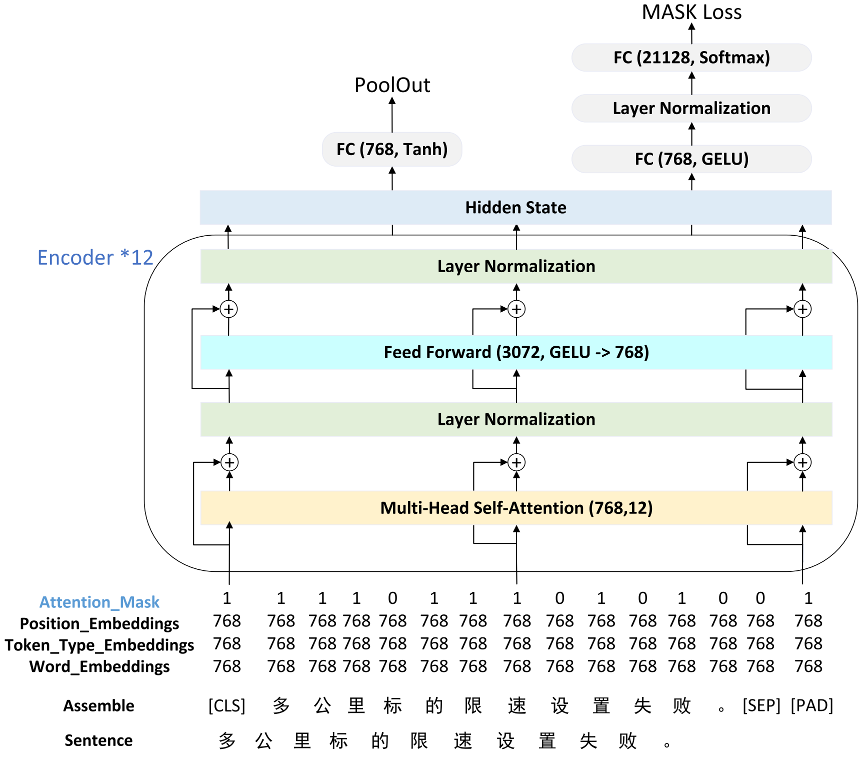
B. CNN

CNN includes two operations of Convolution and Max-Pooling, where Convolution is a feature calculation of the input feature vector matrix using convolution kernels of different sizes. Combined with the Convolution part in Fig. 2, it can be seen that after inputting the output of the upstream model to the CNN the convolution kernels of sizes [2, 768], [3, 768], and [4, 768] are used respectively to extract the feature vectors with structures [64, 256, 44, 768], [64, 256, 43, 768], and [64, 256, 42, 768]. feature vectors, after which Max-Pooling Max-Pooling is taken to obtain three Tensor of size [64, 256] for splicing in order to extract the local feature information of different dimensions in the CTC problem data.

C. BiLSTM-BiGRU

Since the Self-Attention structure is used in Transformer Encoder, the sequential nature of the output features cannot be guaranteed. In order to get the sequential features with correct order, this paper uses BiGRU-BiLSTM weight combination model to model the contextual information of the text of CTC cascade cascade problem, and GRU and LSTM structures are used in the hidden layer unit.

The LSTM neural unit is composed of forgetting gate, input gate and output gate, and its structure is shown in Fig. 4(a). Firstly, h_{t-1} and x_t are connected and then calculated by Sigmoid function with different weight matrix $\left(\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_o\right)$ and bias $\left(b_f, b_i, b_o\right)$ to output f_t, i_t and o_t respectively. In the forgetting gate f_t is multiplied with C_{t-1} to control the amount of information that needs to be forgotten from the previous hidden layer h_{t-1}; in the input gate the content is planned to $(-1, 1)$ by the Tanh function to obtain the information updated cell $\bar{C}_t$, after which it is multiplied with i_t to control which information needs to be retained, where $\mathbf{W}_C$ is the weight matrix. The updated cell state C_t is



Attention_Mask	1	1	1	1	0	1	1	1	0	1	0	1	0	0	1
Position_Embeddings	768	768	768	768	768	768	768	768	768	768	768	768	768	768	768
Token_Type_Embeddings	768	768	768	768	768	768	768	768	768	768	768	768	768	768	768
Word_Embeddings	768	768	768	768	768	768	768	768	768	768	768	768	768	768	768
Assemble	[CLS]	多	公	里	标	的	限	速	设	置	失	败	。	[SEP]	[PAD]
Sentence	多	公	里	标	的	限	速	设	置	失	败	。			

FIGURE 3. Schematic diagram of RoBERTa-wwm calculation.

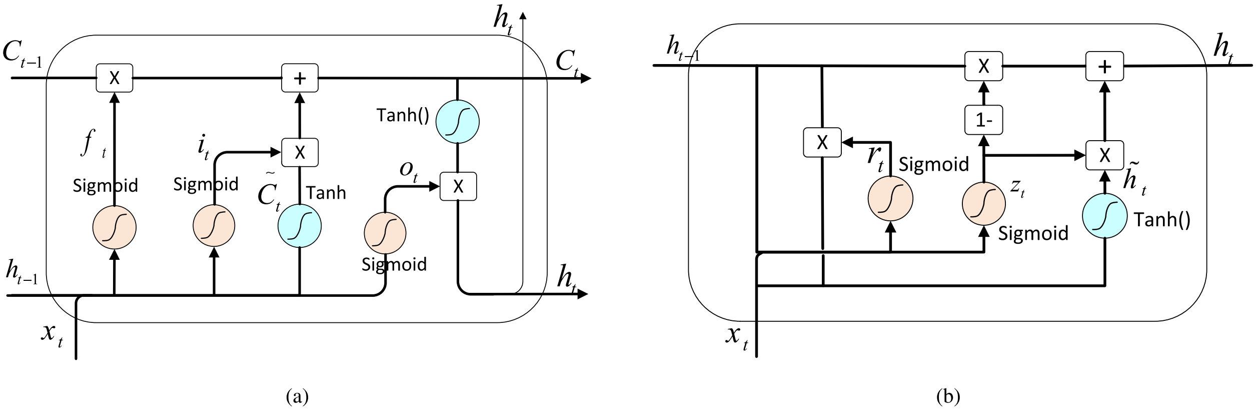


FIGURE 4. LSTM and GRU internal structure.

used as the input of the output gate, which is scaled by Tanh and multiplied by o_t, thus outputting h_t as the next LSTM hidden layer state.

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

$$i_t = \sigma\left(\mathbf{W}_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{2}$$

$$o_t = \sigma\left(\mathbf{W}_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{3}$$

$$\tilde{C}_t = \tanh\left(\mathbf{W}_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{6}$$

GRU is a modification of the structure of the three gates of the LSTM by changing the oblivion gate, the input gate, and the output gate into two gates, the update gate z_t, and the reset gate r_t, and by combining the cell state with the output into a single stateh_t, which is shown in Figure 4(b).

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t] + b_z\right) \tag{7}$$

$$\tilde{h}_t = \tanh\left(W_h \cdot [r_t * h_{t-1}, x_t] + b_h\right) \tag{8}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{9}$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t] + b_r\right) \tag{10}$$

The BiLSTM-BiGRU integrated model based on combined weights, i.e., assigning appropriate weight values to the outputs of BiLSTM and BiGRU, and weighting the output values of both. The higher the overall performance of a single

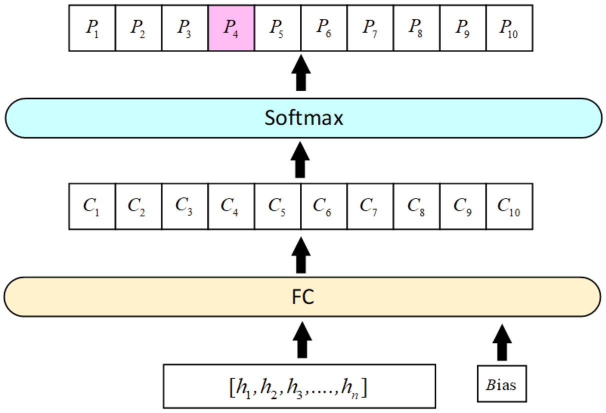

P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10}
Softmax
C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9 C_{10}
FC
$[h_1, h_2, h_3, ..., h_n]$
Bias

FIGURE 5. The structure of output layer.

neural network, the higher its overall weight is assigned, and the weight values are selected by the weight value tuning experiment in the subsequent 4.6. The forward and backward implicit state sequence information extracted by BiLSTM and BiGRU is also combined to solve the problem that the traditional one-way model cannot capture the contextual information, and the sequential nature of the feature vectors is ensured to a certain extent.

D. OUTPUT LAYER

The task of the output layer is mainly accomplished by the fully connected layer FC and Softmax function together, and its working process is shown in Figure 5. Where the fully connected layer FC is each unit of the former layer is connected to the latter layer, i.e., the output tensor dimension of the deep learning integrated model composed of Convolution and BiLSTM-BiGRU is reduced from 768 to 10 (the number of fault categories);

Softmax maps the input values from the FC layer between (0, 1), all input values are calculated using e^n. Then, these results are summed, and then the proportion of each value in the summed values is calculated. equation (15) is calculated as the output is the probability value P_j of category j. By calculating the probability value P_j of each category, a probability distribution $[P_1, P_2, \ldots, P_{10}]$ is obtained, where the subscript corresponding to the maximum probability value number is the classification result of Softmax.

$$P_j = \frac{\exp\left(C_j\right)}{\sum_{k=1}^{10} C_k} \tag{11}$$

V. EXPERIMENTAL VERIFICATION

A. EXPERIMENTAL ENVIRONMENT AND HYPER-PARAMETERS CONFIGURATION

Experimental hardware: CPU is i7-6700HQ, graphics card is GTX3060 with 6G video memory, OS is Win11 64-bit, Python version is 3.90, development tool is Spyer 5.0.5, Pytorch version is 1.11.1 GPU version. Seven PLMs, BERT, BERT-wwm, RoBERTa-wwm, ALBERT, XLnet, ERNIE and GPT-2, were selected for this experiment and their parameter settings are shown in Table 2. The deep neural networks involved in the downstream model during the experiment are BiLSTM, BiGRU, and CNN, and the remaining parameter settings are summarized in Table 3.

TABLE 2. PLM parameter settings.

Parameter	Value
Hidden_size	768
Max_position_embeddings	512
Num_attention_heads	12
Num_hidden_layers	12
Nidden_dropout_prob	0.1
Nidden_act	Gelu
Intermediate_size	3072
Vocab_size	21128
Initializer_range	0.02

TABLE 3. Other experimental parameter settings.

Parameter	Value
Batch_size	64
Padding_size	40
Learning rate	1e-4
Optimizer	Adam
GRU/LSTM's Hidden dimension	768
LSTM/GRU's Output dimension	256
Number of kernels	256
Size of kernels	(2, 3, 4)
Activation function	Relu
Dropout	0.4

B. EXPERIMENTAL EVALUATION INDEX

In this experiment, $Accuracy$, $\Pr ecision$, $\mathrm{Re}\, call$ and $F1$ are used as evaluation indicators. Where TP indicates the number of samples classified and correctly classified; FP indicates the number of samples classified and incorrectly classified; and FN indicates the number of samples not classified and therefore definitely incorrect.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \tag{12}$$

$$\Pr ecision = \frac{TP}{TP+FP} \times 100\% \tag{13}$$

$$\mathrm{Re}\, call = \frac{TP}{TP+FN} \times 100\% \tag{14}$$

$$F1 = \frac{2}{1/\Pr ecision + 1/\mathrm{Re}\, call} \times 100\% \tag{15}$$

The evaluation metrics $Avg - \mathrm{P}$, $Avg - \mathrm{R}$ and $Avg - F1$ are also defined as the weighted average of the 10 labels' $\Pr ecision$, $\mathrm{Re}\, call$ and $F1$, respectively, and the calculation procedure is shown in (20).

$$Avg - \mathrm{P} = \frac{\sum\limits_{i=1}^{10} N_i \cdot P_i}{\sum\limits_{i=1}^{10} N_i},$$

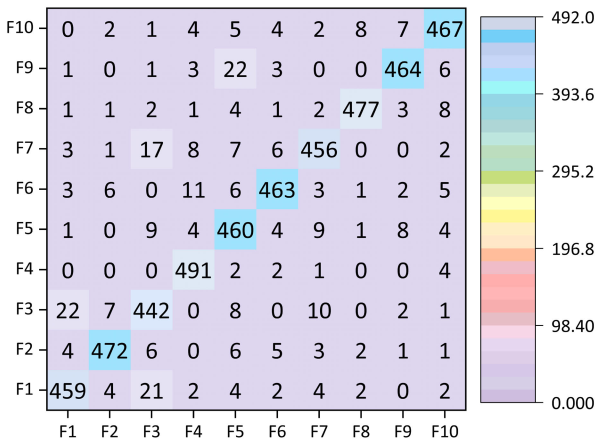


FIGURE 6. Confusion matrix generated by the proposed model.

$$Avg - \mathrm{R} = \frac{\sum_{i=1}^{10} N_i \cdot R_i}{\sum_{i=1}^{10} N_i},$$

$$Avg - F1 = \frac{\sum_{i=1}^{10} N_i \cdot F1_i}{\sum_{i=1}^{10} N_i} \qquad (16)$$

where i denotes the category serial number, N_i denotes the number of data entries of category i, and P_i, R_i, $F1_i$ denote the $\mathrm{Pr}\,ecision$, $\mathrm{Re}\,call$ and $F1$ corresponding to category i, respectively.

C. THE RECOGNITION EFFECT OF THE PROPOSED MODEL FOR DIFFERENT FAULT CATEGORIES

The experiments were trained using a manually constructed CTC alignment joint-test dataset, and the training was completed and evaluated on 5000 test sets with 10 predefined problem types (F1-F10). Based on the confusion matrix shown in Figure 6, $\mathrm{Pr}\,ecision$, $\mathrm{Re}\,call$ and $F1$ are selected as the evaluation indexes of this paper's model at the same time, and the classification results for various types of problem texts of this paper's CTC system are shown in Table 4.

As can be seen from Table 4, the three evaluation index values for the recognition of the 10 problem types all reach above 0.89, while the performance of $\mathrm{Pr}\,ecision$ alone shows that F8 has the highest $\mathrm{Pr}\,ecision$ of 0.9715, which is only 9.36% higher than F5. F4, which has the highest $\mathrm{Re}\,call$, can reach 0.9820, which is only 8.36% higher than F10, which has the lowest $\mathrm{Re}\,call$. In the performance of $F1$, the maximum value is only 6.14% higher than This shows that the classification model proposed in this paper is comprehensive and has excellent and balanced classification effect for each CTC problem type.

TABLE 4. The recognition effect of the proposed model for each CTC problem category.

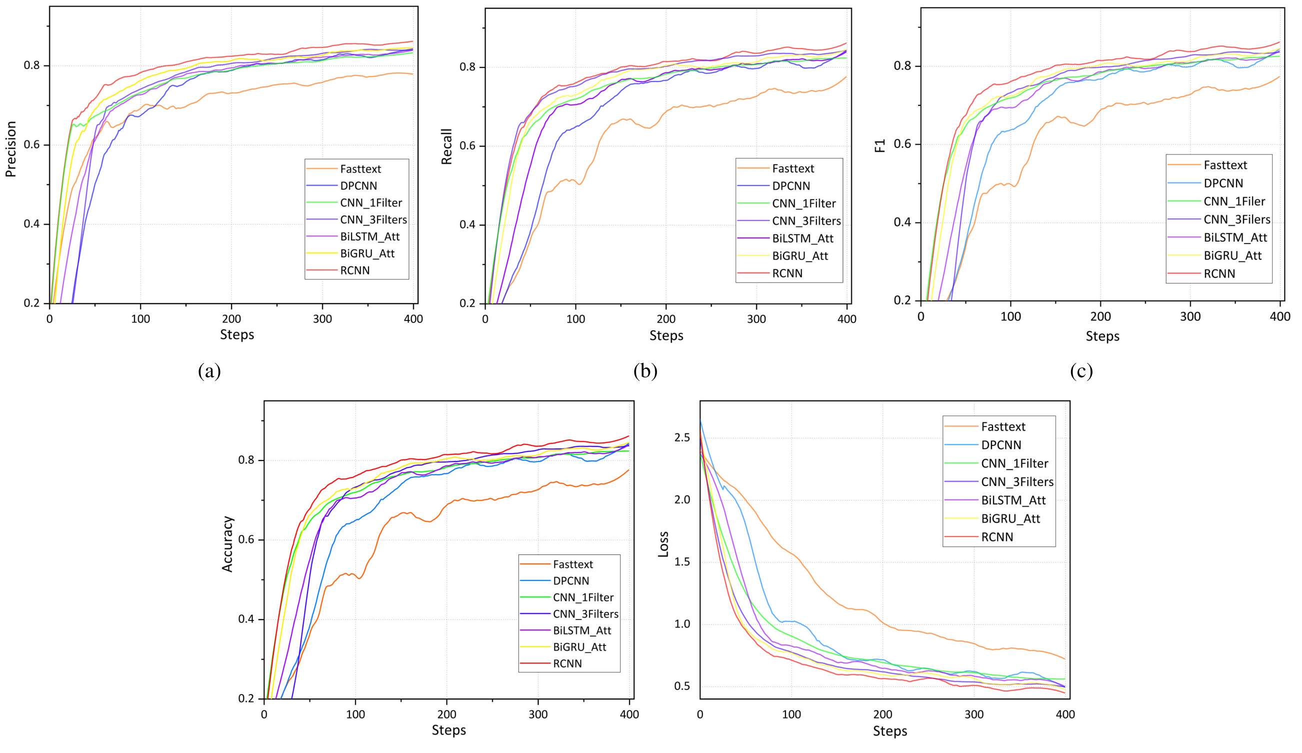
Type of problem	$\mathrm{Pr}\,ecision$	$\mathrm{Re}\,call$	$F1$
F1	0.9291	0.9180	0.9235
F2	0.9574	0.9440	0.9507
F3	0.8858	0.8984	0.8920
F4	0.9370	0.9820	0.9590
F5	0.8779	0.9200	0.8984
F6	0.9449	0.9260	0.9354
F7	0.9306	0.9120	0.9212
F8	0.9715	0.9540	0.9627
F9	0.9528	0.9280	0.9402
F10	0.9340	0.9340	0.9340

D. PERFORMANCE COMPARISON OF DIFFERENT CLASSIFIER MODELS

The following are the classifier models involved in this experiment:

1) Fasttext: Fasttext is a word vector and text classification tool open-sourced by Facebook, whose training is divided into two stages: effective text classification and learning word vector representations.
2) DPCNN: Since TextCNN cannot obtain long-range dependencies of text by convolution, DPCNN can extract dependencies within long texts by continuously deepening the network.
3) CNN_1Filter: The convolution process of this network uses 256 convolution kernels of size 3 to extract the local feature information of the text.
4) CNN_3Filters: The convolution process of this network uses 256 convolution kernels of size 2, 3 and 4 respectively to extract local feature information of the text to extract the local feature information of different dimensions in the sentences.
5) BiLSTM_Att: In the pre-experiments, using single BiLSTM or BiGRU, the performance on the dataset of this paper is not satisfactory because it is difficult to extract the local features of the text, so the addition of the Attention mechanism after BiLSTM can exactly compensate this deficiency.
6) BiGRU_Att: Same as 5), after the BiGRU model, access to the Attention mechanism.
7) RCNN: The features output from BiLSTM and CNN_3Filters are fused, i.e., BiLSTM extracts the contextual information and CNN_3Filters extracts the local feature information.

The experimental results are shown in Figure 7, Table 5 and Table 6, where Figure 7 shows the trend of the evaluation index values of the training process of each classifier model, Table 5 shows the classification effects of different classifier models for each problem category, and Table 6 shows the overall performance of different classifier models, and the information in the above figures and tables can be combined to conclude that: The variation of $Accuracy$, $\mathrm{Pr}\,ecision$, $\mathrm{Re}\,call$ and $F1$ of the other models except Fasttext ranged



(a)

(b)

(c)

(d)

(e)

FIGURE 7. Parameter trends of the training process of each classifier model.

TABLE 5. Classification effects of different classifier models for each problem category ($F1$).

Type	Classifier Models						
	Fasttext	DPCNN	CNN_1Filter	CNN_3Filters	BiGRU_Att	BiLSTM_Att	RCNN
F1	0.7602	0.8202	0.8225	0.8513	0.8263	0.8163	**0.8335**
F2	0.8105	0.8610	0.8778	0.8902	0.8723	0.8677	**0.8933**
F3	0.6473	0.7386	0.7385	0.7562	0.7432	0.7347	**0.7659**
F4	0.8561	0.9143	0.8989	0.9129	0.9169	0.9023	**0.9231**
F5	0.6472	0.7009	0.7536	0.7537	0.7302	0.7244	**0.7654**
F6	0.7888	0.8636	0.8537	0.8720	0.8597	0.8545	**0.8805**
F7	0.7309	0.8377	0.8058	0.8229	0.8202	0.7919	**0.8458**
F8	0.9007	0.9304	0.8675	0.8906	0.9295	0.9225	**0.9443**
F9	0.8073	0.8560	0.8548	0.8646	0.8659	0.8758	**0.8834**
F10	0.7778	0.8540	0.8220	0.8625	0.8692	0.8738	**0.8755**

TABLE 6. Overall performance of different classifier models.

Evaluation Metrics	Classifier Models						
	Fasttext	DPCNN	CNN_1Filter	CNN_3Filters	BiGRU_Att	BiLSTM_Att	RCNN
Accuracy	0.7724	0.8362	0.8294	0.8476	0.8436	0.8374	**0.8606**
Loss	0.72	0.5	0.54	0.48	0.48	0.51	**0.44**
$Avg - F1$	0.7727	0.8377	0.8295	0.8625	0.8433	0.8364	**0.8611**

from 0.7 to 0.8 throughout the training process, while the values of the four evaluation metrics of Fasttext were lower than those of the other models throughout the training process, and the Loss values were consistently greater than those of the other models; By comparing DPCNN, CNN_ 3Filters and CNN_ 1Filter, it can be seen that 3 convolutional kernel CNNs are superior in all four metrics throughout the training process, and the overall classification effect for 10 category problems is greater than that of DPCNN and CNN_ 1Filter, and finally *Accuracy* is improved by 1.5% and 1.42% respectively compared with the two, and $Avg - F1$ is improved by 2.48% and 3.3%, because the DPCNN network has multiple groups of convolution and pooling inside and a complex structure, which cannot play its own advantages in short text classification, while the effect of single-size convolution kernel feature extraction

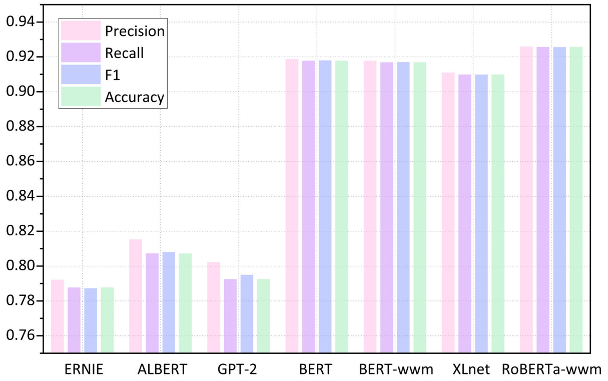


FIGURE 8. Classification results of different PLMs for the dataset of this paper.

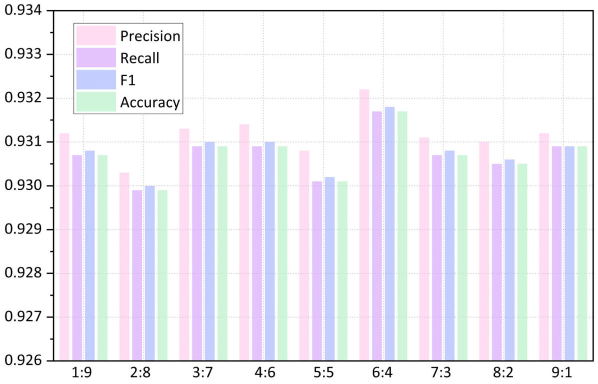


FIGURE 9. Classification results of BiLSTM-BiGRU with different weight ratios on the dataset of this paper.

is not as good as multi-size convolution kernel; The four metrics of CNN_ 3Filters and BiLSTM_ Att and BiGRU_ Att do not differ much in the trend of change during the training process, but from the F1-F7 problem classification results, CNN_ 3Filters is better, and the improvement for each problem type recognition metric is between 1 and 2%; RCNN both *Accuracy*, Pr*ecision*, Re*call* and $F1$ are greater than other models throughout the training process, and the performance indexes for classification of 10 CTC cascade problem categories are significantly improved, with the improvement range between 2.83% and 11.86%, among which the improvement range of *Accuracy* over other models is between 1.3% and 8.82%, This shows that the RCNN network can combine the features of BiLSTM to extract contextual features and CNN_ 3Filters to extract local features at the same time, and the classification effect for CTC joint tuning and joint trial problems is improved compared with the traditional single-depth neural network.

E. PERFORMANCE COMPARISON OF DIFFERENT PLMs

In order to verify the effectiveness of the pre-training model RoBERTa-wwm for the classification of CTC alignment joint-test problems, popular pre-training models in recent years are selected for comparison experiments with the RoBERTa-wwm involved in the model of this paper. All PLMs are trained and predicted on this paper's CTC joint-test problem dataset under the same experimental environment configuration, and the problem classification performance of different PLMs for this dataset is compared. The overall evaluation metrics in the experimental results are shown in Figure 8 and Table 7, in which the four evaluation metrics of BERT, BERT-wwm and RoBERTa-wwm are above 0.91 for *Accuracy*, Pr*ecision*, Re*call* and $F1$, all of which are higher than other PLMs, which shows that the PLM of BERT series has better classification performance for the dataset of this paper; The difference between the evaluation metrics of BERT-wwm and BERT is only about 1%, while RoBERTa-wwm improves about 0.5% compared with BERT-wwm and BERT in all four evaluation metrics, which shows that Roberta-wwm with more data and wwm training method is more effective in classifying CTC problems in this dataset.

In order to further refine the experiment, the classification results of 7 pre-trained models for 10 CTC cascade problems are obtained as shown in Table 8, and it can be seen that BERT, BERT-wwm, RoBERTa-wwm and XLnet perform better in terms of $F1$ for all problems classification compared with other PLMs, and the $F1$ for each problem classification is above 0.88. Therefore, the subsequent comparison tests of BERT, BERT-wwm, XLnet and RoBERTa-wwm will BERT-wwm, XLnet and RoBERTa-wwm continue to be tested for comparison.

As shown in Table 8, RoBERTa-wwm is better for the classification of F1, F2, F3, F6, F7 and F9 problems compared with other pre-trained models, especially this PLM is about 2.58%-2.96% improvement in $F1$ for the classification of F2 problems and 2.41%-4.36% improvement in $F1$ for the classification of F3 problems compared with other three PLMs, while RoBERTa-wwm has the highest *Accuracy* and $Avg - F1$ metrics of 0.9257 and 0.9256, and the final Loss is lower than other PLMs, which is only 0.24. This further indicates that RoBERTa-wwm has the best effect on the problem classification of this paper dataset, so the main model of this paper uses RoBERTa-wwm as the upstream pre-training model.

F. WEIGHT PROPORTIONAL TUNING OF BiLSTM-BiGRU

The BiLSTM-BiGRU integrated model based on combined weights as the core part of the model in this paper, assigning reasonable initial weight values to BiLSTM and BiGRU directly affects the classification performance of the proposed model, therefore, this section explores how to allocate the weight values of BiLSTM and BiGRU to maximize the performance of the integrated BiLSTM-BiGRU model, so that the model can achieve the best classification results. In this experiment, the model in this paper uses 10 sets of BiLSTM-BiGRU with different weight ratios (1:9, 2:8, 3:7, 4:6, 5:5, 6:4, 7:3, 8:2, 9:1) as the core part of the downstream model, respectively, while training and testing

TABLE 7. Overall performance of different PLMs.

Evaluation Metrics	PLMs						
	ERNIE	ALBERT	GPT-2	BERT	BERT-wwm	XLnet	RoBERTa-wwm
Accuracy	0.7877	0.8073	0.7925	0.9179	0.9169	0.9099	**0.9257**
Loss	0.65	0.6	0.64	0.26	0.28	0.31	**0.24**
$Avg-F1$	0.7873	0.8081	0.7950	0.9180	0.9170	0.9099	**0.9256**

TABLE 8. Effectiveness of different PLMs in classifying each problem category.

Type	PLM						
	ERNIE	ALBERT	GPT-2	BERT	BERT-wwm	XLnet	RoBERTa-wwm
F1	0.7468	0.7660	0.7487	0.8957	0.8985	0.8972	**0.9174**
F2	0.8234	0.8529	0.8323	0.9273	0.9283	0.9245	**0.9541**
F3	0.6953	0.6808	0.6850	0.8563	0.8523	0.8368	**0.8804**
F4	0.8717	0.8948	0.8893	**0.9598**	0.9575	0.9584	0.9580
F5	0.6593	0.6813	0.6506	**0.8898**	0.8854	0.8788	0.8863
F6	0.8020	0.8218	0.8039	0.9211	0.9256	0.9116	**0.9324**
F7	0.7541	0.7854	0.7530	0.8982	0.9051	0.8957	**0.9089**
F8	0.9119	0.9253	0.9256	**0.9618**	0.9533	0.9502	0.9598
F9	0.8216	0.8587	0.8577	0.9372	0.9344	0.9333	**0.9374**
F10	0.7853	0.8120	0.8023	**0.9323**	0.9288	0.9121	0.9215

TABLE 9. Effect of BiLSTM-BiGRU classification with different weight ratios.

BiGRU and BiLSTM Weighting ratio	Evaluation Metrics				
	Accuracy	Loss	$Avg-\mathrm{P}$	$Avg-\mathrm{R}$	$Avg-F1$
1:9	0.9307	0.24	0.9312	0.9307	0.9308
2:8	0.9299	0.24	0.9303	0.9299	0.9300
3:7	0.9309	0.24	0.9313	0.9309	0.9310
4:6	0.9309	0.24	0.9314	0.9309	0.9310
5:5	0.9301	0.24	0.9308	0.9301	0.9302
6:4	**0.9317**	0.24	**0.9322**	**0.9317**	**0.9318**
7:3	0.9307	0.24	0.9311	0.9307	0.9308
8:2	0.9305	0.24	0.9310	0.9305	0.9306
9:1	0.9309	0.24	0.9312	0.9309	0.9309

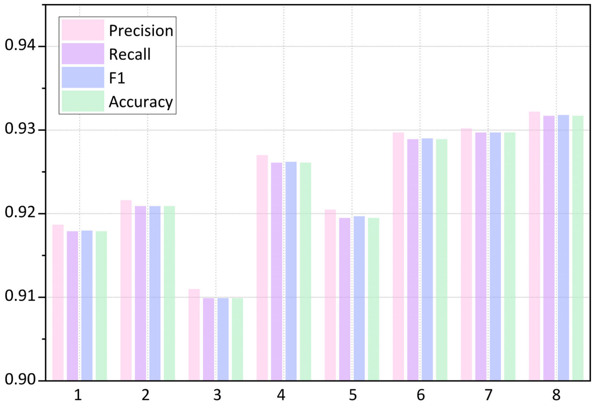


FIGURE 10. Classification results of various text classification models for the dataset.

with the parameters configured in Section V-A.A and using the same CTC alignment joint-test dataset, the classification results are shown in Figure 9 and Table 9.

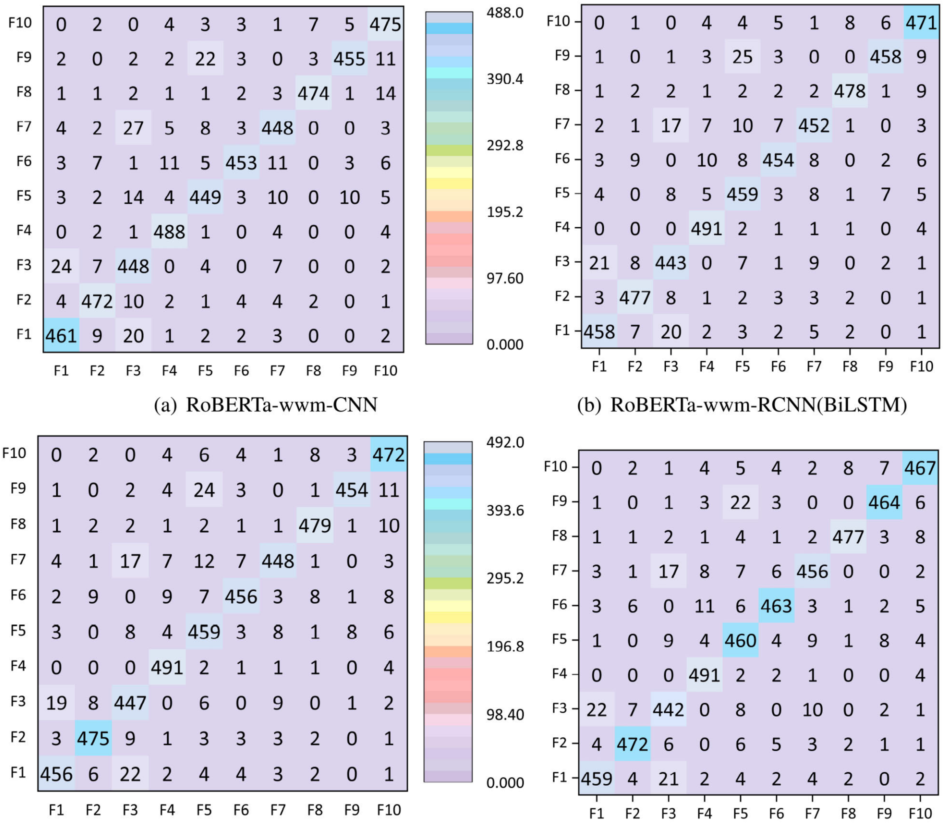
It can be seen that when the combined weight ratio of BiLSTM and BiGRU is 6:4, the best classification performance of this paper's model is achieved, and $Avg-P$ improves about 0.14%-0.2%, $Avg-\mathrm{R}$ improves about 0.08%-0.18%, $Avg-F1$ improves about 0.08%-0.12%, and *Accuracy* improves about 0.08%-0.12% compared with the BiLSTM-BiGRU model with other weight ratios 0.08%-0.18% or so. Meanwhile, it can be seen from Table 9 that the BiLSTM-BiGRU model with a weight ratio of 6:4 has the highest $Avg-\mathrm{P}$, $Avg-\mathrm{R}$ and $Avg-F1$ of 0.9322, 0.9317 and 0.9318, so the model in this paper in subsequent experiments uses the BiLSTM-BiGRU integrated model with a weight ratio of 6:4 as the core part of the downstream model.

G. PERFORMANCE COMPARISON OF DIFFERENT COMBINED TEXT CLASSIFICATION MODELS

Based on the above three sets of comparison tests, this subsection uses seven different combination models for comparison tests, and the combination models all use the pre-trained model as the upstream model and the deep neural network model as the downstream model, which are listed in Table 10 as ordinal numbers 1 9, respectively, where model 1 is directly connected to the Softmax function directly through BERT; model 2 adds the downstream model CNN on the basis of 1; model 3 uses BERT as the upstream model and further extracts the context information through the downstream model BiLSTM; Model 4 adopts RoBERTa-wwm as the upstream model and CNN as the downstream model; Model 5 replaces BERT with RoBERTa-wwm on the

TABLE 10. **Classification results of different combinations of text classification models for the dataset.**

number	Models	Evaluation Metrics				
		Accuracy	Loss	*Avg* − P	*Avg* − R	*Avg* − F_1
1	BERT	0.9179	0.26	0.9187	0.9179	0.918
2	BERT-CNN	0.9209	0.26	0.9216	0.9209	0.9209
3	BERT-BiLSTM	0.9099	0.31	0.911	0.9099	0.9099
4	RoBERTa-wwm-CNN	0.9261	**0.23**	0.927	0.9261	0.9262
5	RoBERTa-wwm-BiLSTM	0.9195	0.27	0.9205	0.9195	0.9197
6	RoBERTa-wwm-RCNN(BiLSTM)	0.9289	0.24	0.9297	0.9289	0.929
7	RoBERTa-wwm-RCNN(BiGRU)	0.9297	0.25	0.9302	0.9297	0.9297
8	The Proposed Model	**0.9317**	0.24	**0.9322**	**0.9317**	**0.9318**



	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
F10	0	2	0	4	3	3	1	7	5	475
F9	2	0	2	2	22	3	0	3	455	11
F8	1	1	2	1	1	2	3	474	1	14
F7	4	2	27	5	8	3	448	0	0	3
F6	3	7	1	11	5	453	11	0	3	6
F5	3	2	14	4	449	3	10	0	10	5
F4	0	2	1	488	1	0	4	0	0	4
F3	24	7	448	0	4	0	7	0	0	2
F2	4	472	10	2	1	4	4	2	0	1
F1	461	9	20	1	2	2	3	0	0	2

(a) RoBERTa-wwm-CNN

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
F10	0	1	0	4	4	5	1	8	6	471
F9	1	0	1	3	25	3	0	0	458	9
F8	1	2	2	1	2	2	2	478	1	9
F7	2	1	17	7	10	7	452	1	0	3
F6	3	9	0	10	8	454	8	0	2	6
F5	4	0	8	5	459	3	8	1	7	5
F4	0	0	0	491	2	1	1	1	0	4
F3	21	8	443	0	7	1	9	0	2	1
F2	3	477	8	1	2	3	3	2	0	1
F1	458	7	20	2	3	2	5	2	0	1

(b) RoBERTa-wwm-RCNN(BiLSTM)

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
F10	0	2	0	4	6	4	1	8	3	472
F9	1	0	2	4	24	3	0	1	454	11
F8	1	2	2	1	2	1	1	479	1	10
F7	4	1	17	7	12	7	448	1	0	3
F6	2	9	0	9	7	456	3	8	1	8
F5	3	0	8	4	459	3	8	1	8	6
F4	0	0	0	491	2	1	1	1	0	4
F3	19	8	447	0	6	0	9	0	1	2
F2	3	475	9	1	3	3	3	2	0	1
F1	456	6	22	2	4	4	3	2	0	1

(c) RoBERTa-wwm-RCNN(BiGRU)

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
F10	0	2	1	4	5	4	2	8	7	467
F9	1	0	1	3	22	3	0	0	464	6
F8	1	1	2	1	4	1	2	477	3	8
F7	3	1	17	8	7	6	456	0	0	2
F6	3	6	0	11	6	463	3	1	2	5
F5	1	0	9	4	460	4	9	1	8	4
F4	0	0	0	491	2	2	1	0	0	4
F3	22	7	442	0	8	0	10	0	2	1
F2	4	472	6	0	6	5	3	2	1	1
F1	459	4	21	2	4	2	4	2	0	2

(d) The Present Model

FIGURE 11. **Confusion matrix generated by different text classification models.**

basis of 3; and Model 6 adopts a single BiLSTM on the basis of keeping the same structure with the model in this paper instead of BiGRU-BiLSTM combination, and model 7 replaces BiLSTM with BiGRU on the basis of model 6. The experimental results of the eight text categorization models on this paper's dataset are shown in Figure 10 and Table 10, and are denoted by ordinal numbers 1 to 8 in the bar charts and tables, respectively.

By comparing models 1, 2 and 3, it can be seen that BERT improves all four evaluation metrics by about 0.3% after adding CNN as the downstream model, while using single BiLSTM as the downstream model of BERT causes a slight decrease in its performance metrics; Similarly by comparing 4 and 5, it can be seen that when using RoBERTa-wwm as the upstream model, the four evaluation metrics are improved by about 0.66% when using CNN as the downstream model compared to a single BiLSTM, because the texts of CTC problems in the dataset of this paper are all short texts of length up to 40, and CNN is better for local feature extraction, it is difficult for BiLSTM to play its own advantage of extracting long sequence information; By comparing experiments 4, 6 and 7, it can be seen that adding BiLSTM or BiGRU to RoBERTa-wwm-CNN improves the performance index by about 0.28%-0.36%, By comparing experiments 4, 6 and 7, it can be seen that adding BiLSTM or BiGRU to RoBERTa-wwm-CNN improves the performance index by about 0.28%-0.36%; Finally, comparing 6, 7 and 8, it can be seen that the integrated BiGRU-BiLSTM model with a weight ratio of 6:4 in the downstream model RCNN has some improvement compared to single BiGRU or BiLSTM, and the improvement of the four performance indicators is around 0.2%-0.28%, and $Accuracy$, $Avg - \mathrm{P}$, $Avg - \mathrm{R}$ and $Avg - F1$ were the highest values in the experiment with 0.9317, 0.9322, 0.9317 and 0.9318, respectively. The confusion matrix generated by combining models 4, 6, 7, and 8 is shown in Figure 11, from which it can be seen that model 8 in this paper has a more balanced classification effect for 10 different types of CTC system problems compared to the remaining three combined models.

VI. CONCLUSION

For the massive unstructured problem text data generated during the alignment joint-test of CTC systems, this paper proposes a classification model for CTC alignment joint-test problems based on the integration of RoBERTa-wwm and deep learning. The model uses the RoBERTa model with wwm as the upstream model, the CNN with 3 different sizes of convolutional kernels and the BiLSTM-BiLSTM with a weight ratio of 6:4 as the downstream model, and fuses the features extracted by Roberta-wwm with the CNN and BiLSTM-BiLSTM to extract the full range of the problem text to the deep features, followed by four experiments and analysis of the experimental results yielded: RoBERTa-wwm as an upstream model achieved better classification results compared to other PLMs on the dataset of this paper; A deep integrated neural network composed of CNN-BiLSTM-BiGRU as a downstream model further promotes the classification effect, and the text features extracted are more comprehensive compared with other downstream models. From the above two points, it is proved that the text classification model proposed in this paper is a method that can effectively improve the classification performance of CTC system problems, and the method can also provide a new way of thinking for railroad text classification. However, this study is targeted at Chinese text containing complex and diverse characters and words, and if we can accurately extract the character features and word features and analyze the correlation between them, it is expected to further improve the accuracy of the classification method of CTC alignment joint-test problems.

REFERENCES

[1] C. H. Gao, *Communication Based Train Control System*. Beijing, China: China Railway, 2018.

[2] A. L. A. T. D. Ambegoda, W. T. S. De Silva, K. T. Hemachandra, T. N. Samarasinghe, and A. T. L. K. Samarasinghe, "Centralized traffic controlling system for sri Lanka railways," in *Proc. 4th Int. Conf. Inf. Autom. Sustainability*, Dec. 2008, pp. 145–149.

[3] A. V. Uriarte-Arcia, I. López-Yáñez, and C. Yáñez-Márquez, "One-hot vector hybrid associative classifier for medical data classification," *PLoS ONE*, vol. 9, no. 4, Apr. 2014, Art. no. e95715.

[4] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," *ACM SIGIR Forum*, vol. 51, no. 2, pp. 268–276, Aug. 2017.

[5] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *J. Artif. Intell. Res.*, vol. 37, pp. 141–188, Feb. 2010.

[6] A. Neuraz, B. Rance, N. Garcelon, L. C. Llanos, A. Burgun, and S. Rosset, "The impact of specialized corpora for word embeddings in natural langage understanding," in *Proc. MIE*, 2020, pp. 432–436.

[7] V. Ashish, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[8] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (IndRNN): Building a longer and deeper RNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5457–5466.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[10] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, "Locating and editing factual associations in gpt," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 17359–17372, 2022.

[11] M. Chen, X. Jin, and D. Shen, "Short text classification improved by learning multi-granularity topics," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1–6.

[12] K. Lee, D. Palsetia, R. Narayanan, Md. M. A. Patwary, A. Agrawal, and A. Choudhary, "Twitter trending topic classification," in *Proc. IEEE 11th Int. Conf. Data Mining Workshops*, Dec. 2011, pp. 251–258.

[13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] M. Ikonomakis, S. Kotsiantis, and V. Tampakas, "Text classification using machine learning techniques," *WSEAS Trans. Comput.*, vol. 4, no. 8, pp. 966–974, 2005.

[15] Z. Faguo, Z. Fan, Y. Bingru, and Y. Xingang, "Research on short text classification algorithm based on statistics and rules," in *Proc. 3rd Int. Symp. Electron. Commerce Secur.*, Jul. 2010, pp. 3–7.

[16] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He, "A survey on text classification: From shallow to deep learning," 2020, *arXiv:2008.00364*.

[17] S. Edunov, A. Baevski, and M. Auli, "Pre-trained language model representations for language generation," 2019, *arXiv:1903.09722*.

[18] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," 2019, *arXiv:1909.11942*.

[19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[20] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[21] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," 2019, *arXiv:1901.02860*.

[22] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[23] J.-S. Lee and J. Hsiang, "Patent claim generation by fine-tuning OpenAI GPT-2," *World Pat. Inf.*, vol. 62, Sep. 2020, Art. no. 101983.

[24] R. Sawai, I. Paik, and A. Kuwana, "Sentence augmentation for language translation using GPT-2," *Electronics*, vol. 10, no. 24, p. 3082, Dec. 2021.

[25] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," 2019, *arXiv:1905.07129*.

[26] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, "Pre-training with whole word masking for Chinese BERT," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 3504–3514, 2021.

[27] I. O. Sigirci and G. Bilgin, "Spectral–spatial classification of hyperspectral images using BERT-based methods with HyperSLIC segment embeddings," *IEEE Access*, vol. 10, pp. 79152–79164, 2022.

[28] K. M. Leung, "Naive Bayesian classifier," Dept. Comput. Sci./Finance Risk Eng., Polytech. Univ., Tech. Rep., 2017, pp. 123–156.

[29] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling," *J. Chemometrics*, vol. 18, no. 6, pp. 275–285, Jun. 2004.

[30] M. Haddoud, A. Mokhtari, T. Lecroq, and S. Abdeddaïm, "Combining supervised term-weighting metrics for SVM text classification with extended term representation," *Knowl. Inf. Syst.*, vol. 49, no. 3, pp. 909–931, Dec. 2016.

[31] S. Dudoit and J. Fridlyand, "Bagging to improve the accuracy of a clustering procedure," *Bioinformatics*, vol. 19, no. 9, pp. 1090–1099, Jun. 2003.

[32] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[33] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on convolutional neural networks (CNN) in vegetation remote sensing," *ISPRS J. Photogramm. Remote Sens.*, vol. 173, pp. 24–49, Mar. 2021.

[34] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, pp. 1–74, Mar. 2021.

[35] P. Bharati and A. Pramanik, "EEP learning techniques—R-CNN to mask R-CNN: A survey," in *Computational Intelligence in Pattern Recognition*. 2020, pp. 657–668.

[36] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306.

[37] W. Fang, Y. Chen, and Q. Xue, "Survey on research of RNN-based spatio-temporal sequence prediction algorithms," *J. Big Data*, vol. 3, no. 3, pp. 97–110, 2021.

[38] L. Renjie, L. Haixiang, X. Li, L. Ran, Z. Zhengxiang, and B. Wansheng, "Fault diagnosis for on-board equipment of train control system based on CNN and PSO-SVM hybrid model," *J. Meas. Sci. Instrum.*, vol. 13, no. 4, 2022.

[39] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Exp. Syst. Appl.*, vol. 72, pp. 221–230, Apr. 2017.

[40] T. Peng, C. Zhang, J. Zhou, and M. S. Nazir, "An integrated framework of bi-directional long-short term memory (BiLSTM) based on sine cosine algorithm for hourly solar radiation forecasting," *Energy*, vol. 221, Apr. 2021, Art. no. 119887.

[41] X. Lin, Z. Quan, Z.-J. Wang, H. Huang, and X. Zeng, "A novel molecular representation with BiGRU neural networks for learning atom," *Briefings Bioinf.*, vol. 21, no. 6, pp. 2099–2111, Dec. 2020.

[42] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 562–570.

[43] J. Lin and E. Liu, "Research on named entity recognition method of metro on-board equipment based on multiheaded self-attention mechanism and CNN-BiLSTM-CRF," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–13, Jul. 2022.

[44] W. Lu, J. Li, J. Wang, and L. Qin, "A CNN-BiLSTM-AM method for stock price prediction," *Neural Comput. Appl.*, vol. 33, no. 10, pp. 4741–4753, May 2021.

[45] J. Xu, Y. Cai, X. Wu, X. Lei, Q. Huang, H.-F. Leung, and Q. Li, "Incorporating context-relevant concepts into convolutional neural networks for short text classification," *Neurocomputing*, vol. 386, pp. 42–53, Apr. 2020.

[46] J. Wang, Z. Wang, D. Zhang, and J. Yan, "Combining knowledge with deep convolutional neural networks for short text classification," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3172077–3172295.

[47] M. Hao, B. Xu, J.-Y. Liang, B.-W. Zhang, and X.-C. Yin, "Chinese short text classification with mutual-attention convolutional neural networks," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 19, no. 5, pp. 1–13, Sep. 2020.

NING QIN received the bachelor's degree in communication engineering from the School of Electronic and Information Engineering, Lanzhou Jiaotong University, in 2003. He is currently an Associate Research Fellow with the Signal and Communication Research Institute, China Academy of Railway Sciences Corporation. His research interests include the area of train dispatching and railway signal control.