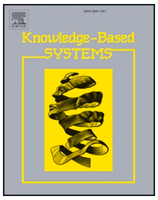
Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Railway accident causation prediction with improved transformer model based on lexical information and contextual relationships

Bin Jiang[a,b,c], Keming Wang[a,b,c,*]

[a] *School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China*
[b] *National-Local Joint Engineering Laboratory of System Credibility Automatic Verification, Chengdu, China*
[c] *Engineering Research Center of Sustainable Urban Intelligent Transportation, Ministry of Education, Chengdu, China*

ARTICLE INFO

Keywords:
Railway safety
Causal analysis
Natural language processing
Deep learning
Data analysis

ABSTRACT

The railway system is a prime example of a safety-critical system. Predicting the causes of railway accidents holds immense significance in enhancing railway transportation safety. Previous approaches to railway causation analysis have encountered huge challenges regarding data processing and analytical capabilities. To address this concern, this paper proposes an innovative deep model framework based on the Transformer architecture that utilizes historical data on railway equipment accidents to predict the causes behind such incidents. Firstly, this paper proposes the utilization of Convolutional Block Attention in the domain of text processing, serving as a lexical encoder to augment word semantics acquisition in accident texts. Subsequently, in order to address the deficiency of traditional Transformers that lack positional representation information, we propose incorporating a BiGRU (Bidirectional Gated Recurrent Unit) as a contextual positional information encoder to capture contextual positional information in railway accident data effectively. Finally, considering that accident data reports are discrete tabular data, this study suggests employing cue word techniques for preprocessing accident data to alleviate the model's learning burden. We applied the proposed model to the FRA (Federal Railroad Administration) dataset. The results demonstrate that our model surpasses the current state-of-the-art language models, exhibiting superior performance compared to the optimal model with a notable improvement of 3.56%, 0.42%, and 0.76% in Precision, Recall, and F1-score, respectively. Furthermore, our model accurately predicts accident categories prone to misjudgment even when trained on limited data, outperforming existing language models. The study findings will contribute to the prevention and management of railway accidents.

1. Introduction

In recent years, with the rapid development of the global railway industry, railways have emerged as the primary mode of daily commuting and the foremost choice for freight transportation. The railway system plays a pivotal role in driving economic growth across most nations. However, occasional railway accidents pose adverse impacts on transportation [1]. For instance, in June 2022, an operational accident took place on China's Guiguang Railway, resulting in the tragic demise of one train driver and injuries to one train attendant and seven passengers [2]. Analyzing the causes behind railway accidents is a fundamental measure for enhancing the safety of railway transportation through ways such as optimizing operational processes and improving system design reliability [3]. However, traditional accident analysis methods have encountered limitations in terms of data processing and analytical capabilities. Therefore, the new effective analysis of the causes behind railway accidents has emerged as a research area of concern [4,5]. Analyzing historical railway operational accidents provides valuable insights for preventing future incidents.

Railway accidents involve multiple factors, including but not limited to technical, human-related, and environmental aspects. On the technical front, equipment malfunctions, track defects, and communication system failures may be critical factors leading to accidents. Human-related factors encompass driver errors, improper operations, and managerial mistakes. Environmental factors like extreme weather conditions and natural disasters may also adversely affect railway transportation. All these accident-related pieces of information are recorded and stored in textual form. In the era of big data, effectively mining and utilizing information from historical accident data is crucial for understanding and preventing accidents [6]. The manual exploration of the

[*] Corresponding author.
E-mail addresses: jb@my.swjtu.edu.cn (B. Jiang), kmwang@swjtu.edu.cn (K. Wang).

https://doi.org/10.1016/j.knosys.2024.111897
Received 13 December 2023; Received in revised form 25 March 2024; Accepted 30 April 2024
Available online 10 May 2024

relationships between accident information and their causes from vast amounts of historical accident data can be highly time-consuming and labor-intensive. Therefore, it is imperative to identify a technology capable of rapidly uncovering relationships among numerous samples, thereby reducing the cost associated with analysis.

As an example supporting this study, the accident data used is sourced from the U.S. FRA Rail Equipment Accident (REA) accident/incident database [7]. This database encompasses all railway accidents in the United States since 1975. Each accident record includes 145 accident elements, such as the accident location, time, latitude and longitude, weather, visibility, brief accident description, etc. This information is comprised of both narrative text and numerical data. Additionally, the United States Federal Railroad Administration has meticulously categorized the causes of accidents, resulting in a database that includes 390 accident causes. While these historical data provide sufficiently detailed information, analyzing such accident data remains time-consuming. Moreover, there is a desire for these accident data to contribute to future predictions of accident causes. Therefore, the challenge is extracting meaningful information corresponding to the accident causes from many accident elements and derive relevant patterns. Another challenge is how to effectively utilize these accident elements to deduce the causes of accidents accurately. To address the challenges mentioned above, existing research has often employed deep learning techniques, such as recurrent neural networks (RNN) [8] and graph neural networks [9]. However, these studies are prone to issues of weak generalizability, and the models are limited to only a few categories, which are insufficient for a comprehensive analysis of railway accidents.

To address the challenges mentioned above, the text-processing capabilities of natural language processing (NLP) technology and the robust learning capabilities of deep learning techniques are effectively employed in this study. We propose a novel deep learning model framework aimed at enhancing the accuracy of current predictions of railway accident causes, addressing the issues of lengthy investigation cycles and extended analysis time associated with railway accidents. More specifically, we introduce a novel deep learning model framework based on Transformer, utilizing Convolutional Block Attention as a lexical encoder to enhance the learning of word semantics in accident texts. To address the deficiency of traditional Transformers in lacking positional representation information, we utilized BiGRU (Bidirectional Gated Recurrent Unit) as a contextual positional information encoder. This further captures contextual positional information, enabling the model to discern concealed relationships among different accident elements in discretized railway accident data and augmenting the model's comprehension capabilities. This framework enables more diverse categories of railway accident cause predictions. In contrast to current research that can only categorize accident causes into a limited number of categories, our model can predict causes across 62 categories. Through comparisons of precision, recall, and F1 scores, our model has achieved the best results on the U.S. FRA Rail Equipment Accident (REA) accident/incident database.

The rest of this paper is organized as follows. Section 2 presents a related literature review. Section 3 formalizes the task of predicting railway accident causes. Section 4 presents the overall framework and specific methods of the proposed model. Section 5 covers experimental results and analysis. Section 6 provides a summary and outlines future work.

2. Literature review

2.1. Applications of natural language processing technology in the transportation field

In recent years, many researchers in the field of safety have utilized various natural language processing techniques to investigate the correlation between traffic safety accidents and their textual content. Gao et al. [10] developed a verb-based text-mining method and investigated around 1000 traffic accident records from Missouri. The findings demonstrate that the extracted information is valuable not only for crash classifications but also for enhancing comprehension of crash causes. Das et al. [11] analyzed the unstructured language content in the data of the Motorcycle Crash Causation Study (MCCS) by applying various natural language processing techniques, including text mining and topic modeling. Fatal and non-fatal accidents were clustered separately, leading to a better understanding of the causal mechanisms between motorcycle accidents. Das et al. [12] employed text mining and interpretable machine learning (IML) techniques to analyze all TUOP accidents (with accident narratives) that occurred in Louisiana from 2010 to 2016. The XGBoost model performed better in the adopted modeling strategy. Kwayu et al. [13] used ten years (2009–2018) of Michigan traffic fatal crash narratives as a case study. Structural topic modeling (STM) and network topology analysis were used to generate and examine the prevalence and interaction of themes from the crash narratives, mainly categorized into pre-crash events, crash locations, and involved parties in the traffic crashes. The results indicate that different accidents are associated with specific themes, and machine learning algorithm tests show high classification accuracy. Wali et al. [14] analyzed rail-trespassing incidents using a ten-year dataset (2006−2015) from the Federal Railroad Administration. Advanced text mining extracts unique factors from crash narratives, revealing a significant association between narratives and severe trespasser injuries. Factors like confirmed suicide attempts and headphone use contribute to fatal injuries. Advanced statistical models provide deeper insights. Practical implications and future research directions are discussed. Bareiss et al. [15] developed a system that utilizes the BERT natural language processing model to identify pedal misapplication (PM) accidents from accident descriptions and validate the system's accuracy. After training, the language model was applied to a test dataset comprising 8668 cases from the North Carolina and National Motor Vehicle Crash Causation Survey (NMVCCS). Zhang et al. [16] applied data mining and deep learning techniques to the accident investigation reports released by the National Transportation Safety Board (NTSB) in the United States to assist airlines in predicting adverse events. Lao et al. [17] proposed a semi-supervised weighted prototypical network (SSWPN) for fault classification of switch machines under unlabeled fault samples. The results indicate that SSWPN exhibits good robustness and generalization.

The aforementioned studies demonstrate the extensive application of natural language processing and deep learning technologies in traffic accident analysis, which have yielded impressive results and significantly alleviated investigators' workload.

2.2. Railway accident cause analysis

The essence of railway accident prediction entails addressing a multi-classification problem, wherein the objective is to develop predictive methodologies that effectively classify data instances (accident records) within the dataset into predefined accident-type labels. Soleimani et al. [18] through text mining on narrative text information from traffic accident data, investigated the underlying causes of accidents at highway-railway intersections. They applied machine learning algorithms, including random forest and logistic regression, to construct predictive models for classification tasks. Zhou et al. [19] addressed the issue of imbalanced accident data in highway-railway grade crossing (HRGC) incidents by employing the random forest algorithm to construct a predictive model. They compared it with a decision tree model, and the results indicated a significant improvement in prediction accuracy with the random forest algorithm without introducing additional false negative or false positive predictions. Bridgell et al. [20] introduced an Extreme Gradient Boosting method (XGBoosting) to predict railway accident types. The results indicated that among 16 different machine learning algorithms, the Extreme Gradient Boosting

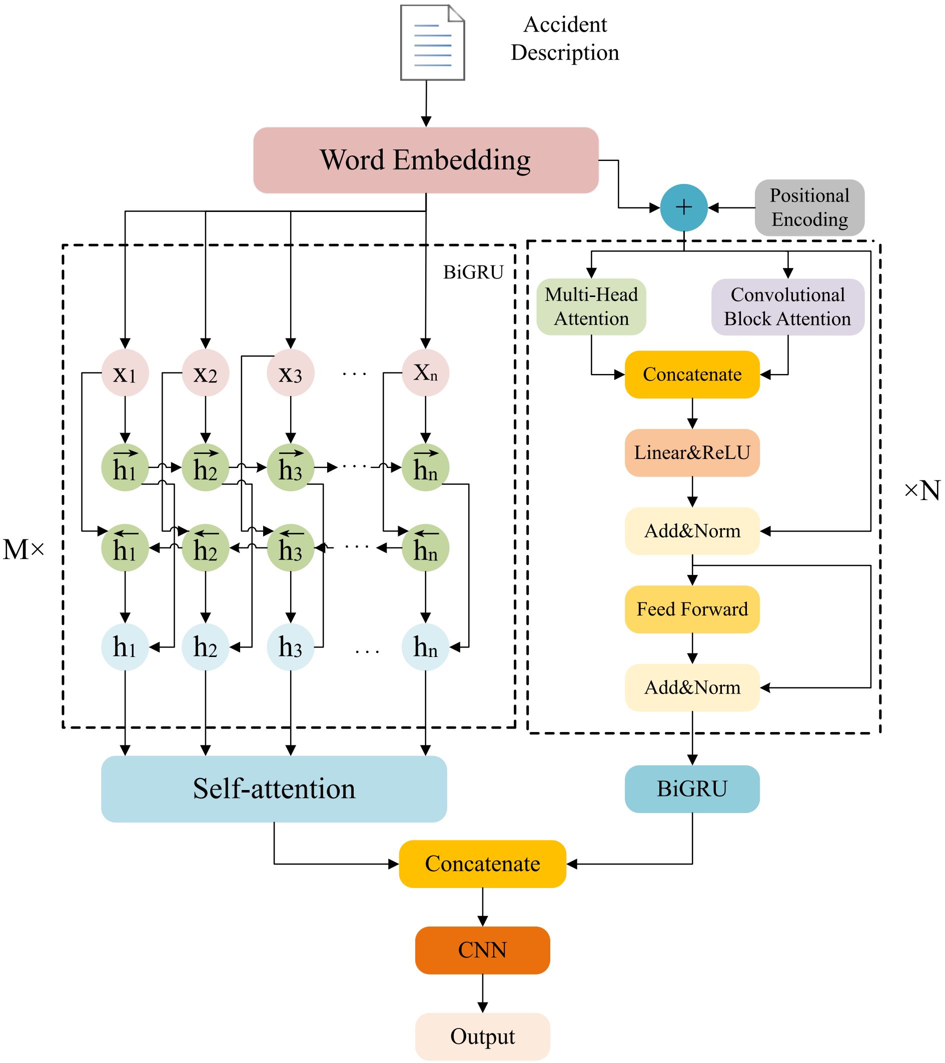


Fig. 1. Transformer-based railway accident cause analysis model.

algorithm demonstrated the best performance in predicting accident types. Gao et al. [21] utilized 19 years of highway-railway grade crossing (HRGC) accident data from North Dakota. They compared a convolutional neural network model with commonly used machine learning models, and the results indicated that, compared to other traditional machine learning methods, deep learning approaches better adapted to data variations, demonstrating superior predictive performance. Meng et al. [22] proposed an ensemble learning strategy for accident prediction and applied it to the Federal Railroad Administration (FRA) dataset. The results showed that the proposed ensemble learning strategy exhibited smaller prediction errors and faster inference times in predicting railway accidents than the artificial neural network (ANN), XGBoost, GBDT, Stacking, and AdaBoost methods. Song et al. [23] improved the classical Transformer Bidirectional Encoder Representations from Transformers (BERT) using a deep neural network (DNN). The superiority of BERT-DNN was validated by employing several additional text classification methods on an actual railway accident database. The results demonstrated that the proposed method accurately predicts accident causes based on railway accident narratives and outperforms previous state-of-the-art text classification methods.

The studies above indicate that significant progress has been made in the field of railway accident prediction through the development of a range of predictive analysis methods. However, these studies often categorize accident causes into a limited number of categories, which limits their ability to comprehensively analyze railway accidents. Refining the classification of accidents may lead to a sharp decline in their predictive performance, making them less usable. Additionally, these studies frequently utilize relatively small training datasets, resulting in poor model generalization. This is one reason for proposing the framework presented in this paper.

3. Problem formulation

This paper focuses on the task of predicting the causes of railway accidents. The core research data is represented by the accident records report D, which can be described as:

$$D = [D_f, D_a] \tag{1}$$

Where: D_f represents the information features of accident elements in the accident report, such as the location, time, latitude and longitude, weather, visibility, and a brief description of the accident. D_a represents the main causes leading to the occurrence of the accident.

The complete D can only be obtained after the accident investigation and analysis. This means that the input for the railway accident cause prediction task includes only the accident element information D_f from the accident report, with the target output being D_a. Therefore, the railway accident cause prediction task can be formalized as the following description:

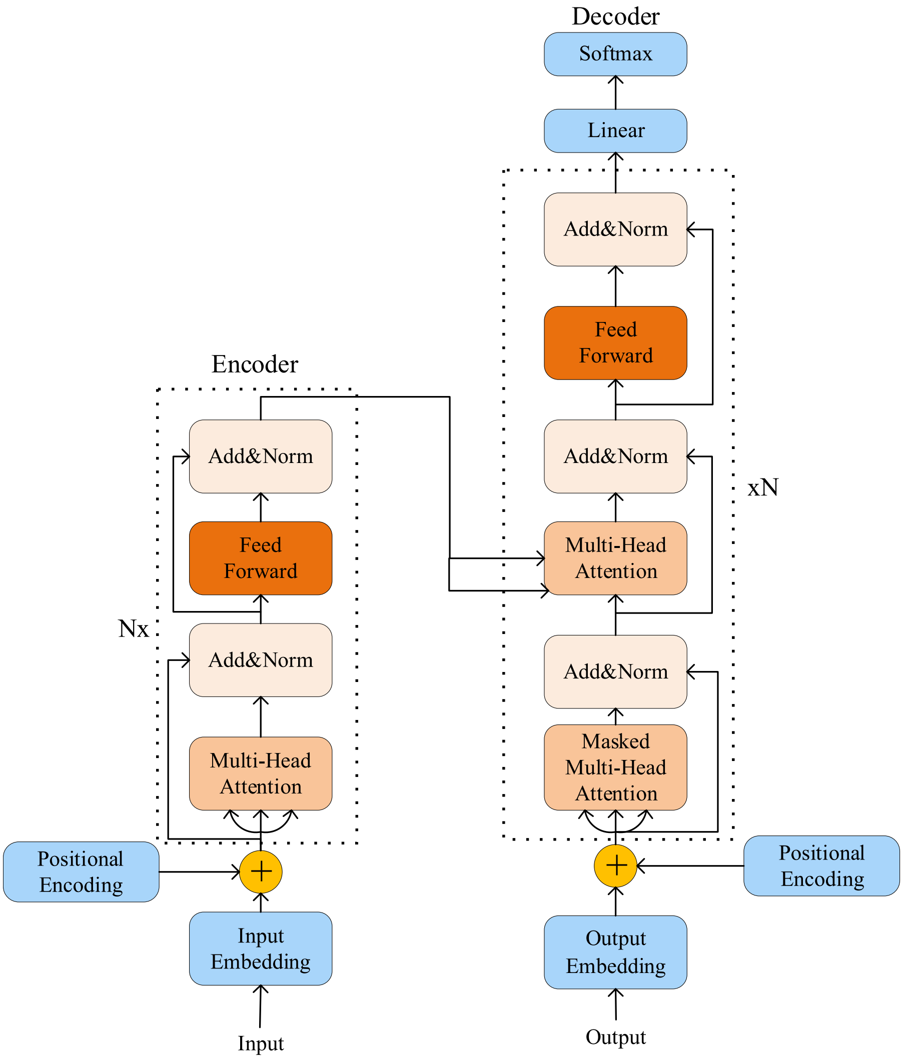


Fig. 2. Transformer architecture [26].

Given a set of $D_f^{\,i} = \left\{ {d_1}^i, {d_2}^i, {d_3}^i, {d_4}^i, \ldots, {d_n}^i \right\}$, $D_A = \{a_1, a_2, a_3, \ldots, a_m\}$, where $d_j^{\,i}$ represents the jth accident element information in the ith accident report, the data in our study is presented in tabular form, $d_j^{\,i}$ corresponds to each individual column of the table; while D_A represents the overall set of accident causes. Our goal is to learn a classifier ζ that can predict the main cause of the accident from the given $D_f^{\,i}$, $\zeta\left(D_f^{\,i}\right) = a_m^i$.

4. Transformer-based railway accident cause analysis model

To accomplish the railway accident cause prediction task, we propose a Transformer-based model for railway accident cause analysis. As shown in Fig. 1, the model consists of the following components:

(1) Word Embedding Layer: Since our accident reports are provided in text form, text preprocessing is performed initially to convert human language into a form that computers can understand and process. In the early days of natural language processing, methods such as One-Hot Encoding, Bag of Words (BOW), and N-gram models were commonly used. Later, with the advent of neural networks, Neural Network Language Models (NNLM) [24] and Word2Vec [25], proposed by Google, became prevalent. The word embedding model employed in this paper is the Word2Vec model, which was introduced to address the limitation of one-hot vector encoding in capturing the similarity between words. Our task may emphasize semantic understanding at the word level to a certain extent, rather than requiring more complex contextual dependencies. Word2Vec provides sufficient semantic information in such a scenario while avoiding the computational costs of handling large language models.

(2) Feature Encoding Layer: This layer primarily comprises a set of M layers of BiGRU and N layers of enhanced Transformer-Encoder blocks. In the model constructed in this paper, both M and N are set to 4 based on experimental adjustments. The main function of this layer is to recognize hidden relationships between features in accident reports and feed the encoded relational features into the output layer.

(3) Output Layer: The output layer transmits the relationship features acquired from the feature coding layer to the CNN for subsequent aggregation, thereby determining the most pertinent accident feature information that can optimally support the input accident case.

4.1. *Transformer-encoder architecture*

The Transformer model was initially proposed in 2017 by the researchers of Google, Vaswani et al. [26]. This model utilizes a pure attention mechanism to encode input sequences, and its encoder and

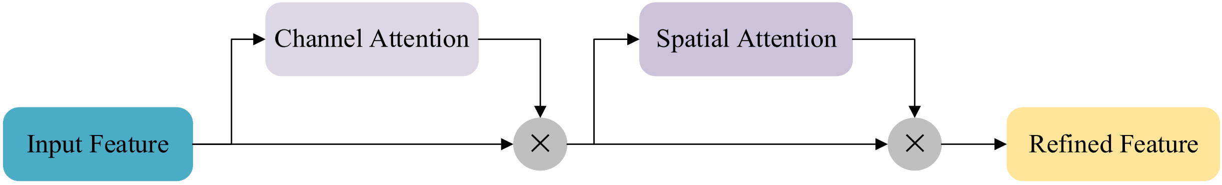


Fig. 3. Convolutional block attention module architecture.

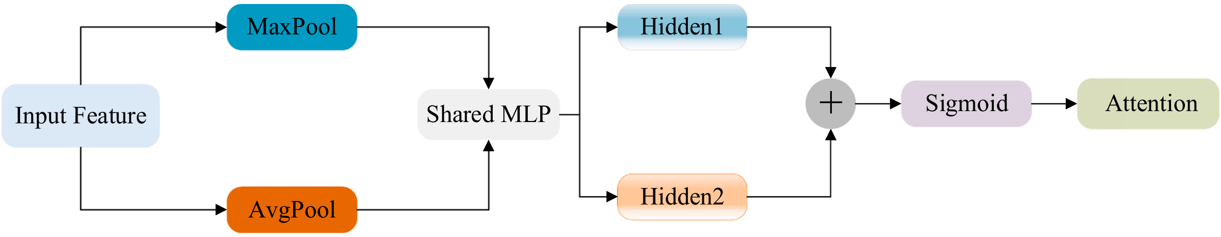


Fig. 4. Channel attention module architecture.

decoder consist of attention mechanisms and feedforward neural networks. It abandons the previously commonly used recurrent neural network structures. The Transformer structure is illustrated in Fig. 2. In comparison to recurrent neural network models, the Transformer model has the following advantages:

(1) Highly parallelized structure: In contrast to recurrent neural networks, the Transformer model has a highly parallelized structure, significantly improving the training speed of the model.
(2) Self-attention mechanism: The Transformer model introduces a self-attention mechanism, allowing it to capture global information better, leading to significant improvements in translation quality and other tasks.

Since the essence of the railway accident causation prediction task is a text classification task, we mainly utilize the Encoder part of the Transformer model. The Transformer's Encoder comprises multiple identical layers, each having two sub-layers. The first sub-layer is a multi-head attention aggregation layer, and the second is a position-wise feedforward network layer. Specifically, when calculating the self-attention of the encoder, the queries, keys, and values all come from the output of the previous encoder layer. Inspired by residual networks, each sub-layer adopts residual connections. Therefore, in the Transformer, for any input $x \in R^d$ at any position in a sequence, the output is required to satisfy $y \in R^d$, so that the residual connection fulfills $x + y \in R^d$, R^d is d-dimensional real vector. After the residual connection, layer normalization is immediately applied. Therefore, the Transformer encoder generates a d-dimensional representation vector for each position corresponding to each input sequence.

The Transformer model primarily utilizes multi-head attention, where each attention head is capable of learning distinct tasks. The support of self-attention increases the receptive field between sentences, allowing the current word to relate to any word in the sentence during information integration. However, this attention mechanism is considered flawed in this study. It primarily focuses on the relationships between words, neglecting the meaning of individual words and the issue of the importance of words within sentences. Therefore, Convolutional Block Attention is introduced to address this problem.

4.2. Lexical encoder

CBAM (Convolutional Block Attention Module) [27] is a lightweight attention model initially used in image recognition. It performs attention operations on both spatial and channel dimensions. In this study, we introduced it into natural language processing because we observed that when computing attention scores in the text domain, it effectively considers the meaning of words themselves and the importance of each word.

The CBAM structure, as shown in Fig. 3, consists of two sub-modules: the Channel Attention Module (CAM) and the Spatial Attention Module (SAM). In Fig 3, the input features undergo sequential processing involving channel attention and spatial attention, respectively.

The computation process of the channel attention module is illustrated in Fig 4. Firstly, the input features are subjected to max pooling and average pooling operations for the purpose of aggregating feature information, thereby generating two distinct channel context descriptors. Then, these two pieces of information are subsequently fed into a shared network (Shared MLP), which consists of two one-dimensional convolution operations. Following the shared network operation, two different channel features, Hidden1 and Hidden2, are obtained respectively. Finally, the two-channel features are fused together and passed through the Sigmoid activation function to acquire channel attention.

The computation process of the spatial attention module is depicted in Fig 5. The input of spatial attention is the output of channel attention. Initially, the input undergoes max pooling and average pooling operations for the aggregation of channel features. The term "Concatenate" refers to concatenating the results of max pooling and average pooling, which are then fed into the convolutional layer (Conv layer). Finally, the

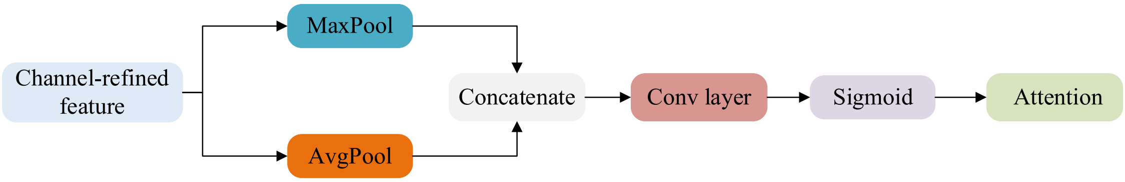


Fig. 5. Spatial attention module architecture.

result of the convolutional layer is passed through the Sigmoid activation function to derive the final spatial attention.

Assuming an accident report description is represented as a sentence $F = \{f_1, f_2, f_3, \ldots, f_n\}$ consisting of n words, where f_i represents the i^{th} word, $i \in \{1, 2, \cdots, n\}$. Firstly, each word is transformed into a word vector w_i through Word Embedding, where $w_i \in R^d$. After adding positional encoding, $F \in R^{n*d}$. Subsequently, F is passed through both the Multi-Head Attention and the Convolutional Block Attention. The Convolutional Block Attention is responsible for learning the semantics and importance of individual words.

In the Convolutional Block Attention module, as convolutional operations are used, we first need to resize F, resulting in $F^{'} \in R^{d*n}$, to ensure that our word vectors are not split during one-dimensional convolution. $F^{'}$ is then subsequently inputted into Convolutional Attention Module (CAM), and the calculation expression in CAM is represented by formula 2 and formula 3.

$$score_{cam} = \sigma(SharedMLP(MaxPool(F^{'})) + SharedMLP(AvgPool(F^{'}))) \in R^{d*1} \tag{2}$$

$$F^{''} = (F^{'} \odot score_{cam}) \in R^{d*n} \tag{3}$$

In formula 2, *SharedMLP* consists of two one-dimensional convolution operations, *MaxPool* is the maximum pooling operation, $MaxPool(x) \in R^{d*1}$; *AvgPool* is the average pooling operation, σ is an activation function (Sigmoid). Through this formula, we obtain a set of attention scores $score_{cam}$.

In formula 3, the multiplication of $score_{cam}$ with the word vectors in F' enables the model to learn the importance representation of different feature dimensions within the word vector. For instance, for one word with a vector denoted as $w_i = \{a_1{}^i, a_2{}^i, \ldots, a_d{}^i\} \in R^{d*1}$, and its score computed by formula 2 is $score_{cam} = \{s_1, s_2, \ldots, s_d\} \in R^{d*1}$, multiplying w_i with the corresponding positions in $score_{cam}$, the formula 3 assigns higher scores to more significant dimensions, thereby resulting in larger product values. The score increases as the dimension becomes more important, ensuring effective semantic learning of the word itself. The model achieves optimal semantic learning for word embeddings.

The computed result $F^{''}$ derived from the CAM is subsequently propagated to the Spatial Attention Module (SAM), where in the calculation process in the SAM is expressed by formula 4 and formula 5.

$$score_{sam} = \sigma(Conv([MaxPool(F^{''}); AvgPool(F^{''})])) \in R^{1*n} \tag{4}$$

$$F^{'''} = (F^{''} \odot score_{sam}) \in R^{d*n} \tag{5}$$

In formula 4, *MaxPool* is the maximum pooling operation, $MaxPool(x) \in R^{d*1}$, *AvgPool* is the average pooling operation, $AvgPool \in R^{1*n}$. By applying this formula, we derive an additional set of attention scores $score_{sam}$.

In formula 5, the sentence-level semantic learning is achieved by multiplying the attention score $score_{sam}$ with $F^{''}$. For instance, $score_{sam} = \{s_1, s_2, \ldots, s_n\} \in R^{1*n}$, where each score s_i is multiplied with the i^{th} word vector in $F^{''}$. Consequently, the intermediate form of the word vector in $F^{'''}$ is expressed as $w_i = \{a_1{}^i, a_2{}^i, \ldots, a_d{}^i\} \times s_i, \; i \in \{1, 2, \cdots, n\}$. This calculation ensures that all feature dimensions of a given word are assigned equal weightage, guaranteeing different weights for distinct words at a sentence level and assigning lower weights to less significant words. Finally, converting the scale of the output $F^{'''}$ into the scale of F, yields a final output $F_{cbam} \in R^{n*d}$ of the CBAM module.

The output of CBAM is combined with the output of Multi-Head Attention (MHA) F_{mha}, resulting in $F_{attention} = [F_{mha}; F_{cbam}] \in R^{n*2d}$, where $F_{mha} \in R^{n*d}$ represents the output of Multi-Head Attention. Through a linear layer and a ReLU layer, attention features are aggregated to ensure that the output satisfies the requirements of the original Transformer structure, denoted as $x \in R^d$. The subsequent computational process remains consistent with the traditional Transformer-Encoder, without modification in this study.

4.3. Contextual information encoder

Traditional Transformers employ positional encoding to address parallelization challenges and ensure accurate word positioning. However, unlike word vectors, positional encoding lacks the linear transformation property in the semantic space. It functions more as a manually designed index, which is inadequate for effectively representing position information. To overcome this limitation, we utilize BiGRU [28], a bidirectional recurrent neural network that extracts position-related information, compensating for the traditional Transformer's lack of accurate position information and enhanceing the capture of contextual information.

First, the accident report F undergoes Word Embedding, representing each word as a word vector w_i, where $w_i \in R^d$, $F = \{w_1, w_2, \ldots, w_n\}$, where n is the number of words, $i \in \{1, 2, \cdots, n\}$. The word embeddings w_i are passed through the BiGRU layer to obtain the true hidden states $\overrightarrow{h_i}$ and the backward hidden states $\overleftarrow{h_i}$. The final hidden states h_i is derived by concatenation, as shown in formula 6, formula 7, and formula 8.

$$\overrightarrow{h_i} = \overrightarrow{GRU}(w_i) \in R^d \tag{6}$$

$$\overleftarrow{h_i} = \overleftarrow{GRU}(w_i) \in R^d \tag{7}$$

$$h_i = \left[\overrightarrow{h_i}; \overleftarrow{h_i}\right] \in R^{2d} \tag{8}$$

As a result, the embedded representation $F = \{w_1, w_2, \ldots, w_n\}$ of the accident report text enters the GRU to accquire the hidden states, as shown in formula 9.

$$\{h_1, h_2, \ldots, h_n\} = BiGRU(\{w_1, w_2, \ldots, w_n\}) \tag{9}$$

The significance of words in the accident report description varies when it comes to predicting the cause of the accident. Hence, a self-attention mechanism enables the model to emulate human perception methods and behavioral characteristics, ultimately acquiring the ability to discern between crucial and insignificant components. The hidden states $H = \{h_1, h_2, \ldots, h_n\}$ are fed into the self-attention mechanism module, where we employ the Dot-Product Attention [26], and derive the output G_h as shown in formula 10.

$$G_h = softmax\left(\frac{HH^T}{\sqrt{d}}\right)H \tag{10}$$

After this calculation, we obtain $G_h \in R^{2d}$. As shown in Fig. 1, the output of the improved Transformer-Encoder is passed through another BiGRU, resulting in $G_h^{'} \in R^{2d}$. Concatenating G_h and $G_h^{'}$, we get $G_h^{''} = [G_h; G_h^{'}] \in R^{4d}$, which is then fed into the CNN module to aggregate the feature information from the BiGRU layer and the improved Transformer layer. This process determines the most relevant and supportive accident feature information for the input, thereby generating the final prediction output.

5. Experiments

This study's developed Transformer-based railway accident cause analysis model was trained on a server equipped with an AMD EPYC 7642 48-Core Processor CPU and an RTX 3090 GPU with 24GB VRAM. Python 3.8 was the programming language, and PyTorch 2.0.0 served as the deep learning framework. The specific parameters for the model we trained are as follows: word embeddings are set to 128 dimensions, 4 layers of BiGRU, and 4 layers of enhanced Transformer-Encoder blocks. The initial learning rate for our model is set to 2e-4, with a corresponding decrease based on the number of iterations. We trained the

Table 1
Categorization of accident cause.

Code	Accident Cause
MY01	Track Foundation Issues
MY02	Track Geometry Irregularities and Abnormal Track Gauge
MY03	Track Break
MY05	Track Joint Issues
MY06	Switch and Turnout Equipment Defects
MY07	Extreme Environmental Conditions Accident
MY08	Accidents Caused by Road-Related Users
MY09	Foreign Object on Track or Wheelsets
MY10	Station Operation Issues
MY11	Railway Operation Issues Caused by External Human Interference
T217	Mismatched rail-head contour
T303	Guard rail loose/broken or mislocated
T305	Retarder worn, broken, or malfunctioning
T307	Spring/power switch mechanism malfunction
T308	Stock rail worn, broken or disconnected
T404	Catenary system defect
M201	Load shifted
M203	Overloaded car
M204	Improperly loaded car
M406	Fire, other than vandalism, involving on-track equipment

Table 2
The top 10 significant accident factors.

Feature name	Description
STATION	Nearest city and town
TRKNAME	Track identification
TRNSPD	Speed of train in miles per hour
WEATHER	Weather conditions
LATITUDE	Latitude in decimal degrees
LONGITUDE	Longitude in decimal degrees
RAILROAD	Railway code
TEMP	Temperature in degrees Fahrenheit
STATE	FIPS state code
TRNNBR	Train id number

model for 30 epochs, utilizing Xavier initialization for the randomization of model parameters. The optimizer used is Adam. Throughout the training process, we selected the model with the highest accuracy on the validation set as the final saved model. To enhance the model's reusability, we have made the code and dataset publicly available [29].

5.1. Date set

The data used in this study is selected from the U.S. FRA Rail Equipment Accident (REA) accident/incident database. The database [7] records railway equipment accident reports in tabular form since 1975. The accident reports document 145 accident factors, including the time and location of the accident, environmental conditions, track status, accident causes, accident types, and more. The reasons for accidents are diverse, with each reason being independent or interconnected, leading to different types of accidents. Through our analysis of the types of accidents reported to the Federal Railroad Administration in the United States, we have identified 62 types of accident causes. Due to the large number of types, we present only 20 types of causal factors in Table 1. All types are shown in Appendix A. The codes beginning with MY are types we categorized ourselves, while the rest are accident types classified by the Federal Railroad Administration.

The database of railway equipment accidents from the United States Federal Railroad Administration contains numerous accident factors, many of which are redundant elements. Therefore, prior to inputting the data into the training model, it is necessary to manually eliminate irrelevant accident elements. Initially, we identified the top 10 significant accident factors based on the literature [22], as shown in Table 2. Subsequently, we selected the remaining 31 important accident factors from the accident table. Ultimately, a total of 41 relevant accident factors, as presented in Appendix B., were inputted into the model.

Since accident factors are discrete data, in order to allow the model to understand these accidents element features fully, we need to preprocess the data when inputting it into the model by adding prompt words, as shown in Fig. 6. The accident factors in this figure are systematically organized, with different special tag words ("<Date>", "<County>", etc.) added before each accident factor text. This facilitates the model's comprehension of relevant semantics in subsequent words through these prompt words during learning, thereby enhancing its ability to understand and comprehend text.

After filtering and cleaning the railway accident database from the United States Federal Railroad Administration, we obtained 99,891 usable records. The dataset was then divided into training, validation, and test sets with an 8:1:1 ratio. The validation set was utilized for fine-tuning model hyperparameters and preliminary assessments of the model's capabilities.

5.2. Evaluation metrics

The model's predictive performance was evaluated using three metrics: *Precision, Recall*, and *F1-score. Precision* measures the ratio of correctly predicted positive samples to the total number of samples predicted as positive. It quantifies the proportion of positive samples among all instances predicted as positive. *Recall* measures the ratio of correctly predicted positive samples to the total number of actual positives. It assesses the proportion of positive samples among all actual positives. *F1-score* represents a harmonic mean between recall and precision values. The calculation formulas are provided below.

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F1 - score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{13}$$

In formula 11 and formula 12, *TP* is the number of positive samples that are rightly classified, *FP* is the number of negative samples that are wrongly classified, and *FN* is the number of positive samples that are wrongly classified.

Given the multi-class classification of this problem, we employ

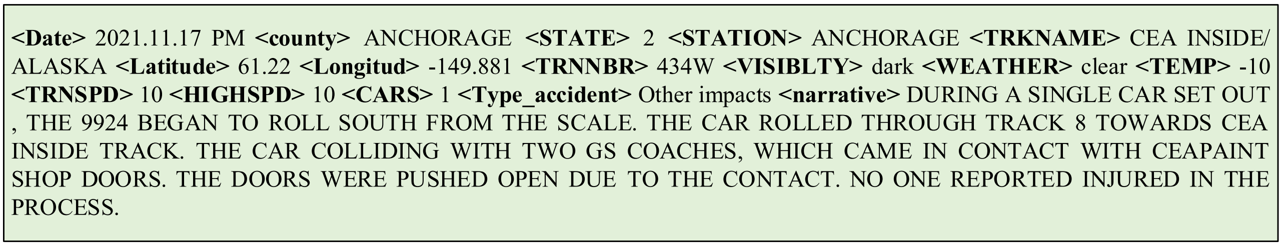

<Date> 2021.11.17 PM **<county>** ANCHORAGE **<STATE>** 2 **<STATION>** ANCHORAGE **<TRKNAME>** CEA INSIDE/ ALASKA **<Latitude>** 61.22 **<Longitud>** -149.881 **<TRNNBR>** 434W **<VISIBLTY>** dark **<WEATHER>** clear **<TEMP>** -10 **<TRNSPD>** 10 **<HIGHSPD>** 10 **<CARS>** 1 **<Type_accident>** Other impacts **<narrative>** DURING A SINGLE CAR SET OUT , THE 9924 BEGAN TO ROLL SOUTH FROM THE SCALE. THE CAR ROLLED THROUGH TRACK 8 TOWARDS CEA INSIDE TRACK. THE CAR COLLIDING WITH TWO GS COACHES, WHICH CAME IN CONTACT WITH CEAPAINT SHOP DOORS. THE DOORS WERE PUSHED OPEN DUE TO THE CONTACT. NO ONE REPORTED INJURED IN THE PROCESS.

Fig. 6. Data Preprocessing.

Table 3
Prediction results of accident causes.

Model	Precision			Recall			F1-score		
	Macro	Weighted	Avg	Macro	Weighted	Avg	Macro	Weighted	Avg
TextCNN	24.92	43.28	34.10	10.23	43.13	26.68	10.24	36.24	23.24
BiLSTM	57.11	66.28	61.70	44.18	66.73	55.46	47.19	65.38	56.29
BiGRU	64.26	68.43	66.35	47.75	67.67	57.71	51.30	66.35	58.83
BERT	59.35	68.01	63.68	51.78	67.86	59.82	54.06	67.44	60.75
Ours	64.82	69.65	**67.24**	51.62	68.85	**60.24**	54.97	68.05	**61.51**

$Macro_{Precision}$, $Macro_{Recall}$, and $Macro_{F1_{score}}$ as the performance evaluation indicators. Corresponding formula is shown formulas 14–16. In the following formulas, n represents the total number of accident categories, and i represents the i^{th} category.

$$Macro_{Precision} = \frac{1}{n}\sum_{i=1}^{n} P_i \tag{14}$$

$$Macro_{Recall} = \frac{1}{n}\sum_{i=1}^{n} R_i \tag{15}$$

$$Macro_{F1_{score}} = \frac{1}{n}\sum_{i=1}^{n} F1_i \tag{16}$$

To address the category imbalance issue in the dataset, we also used $Weighted_{Precision}$, $Weighted_{Recall}$, and $Weighted_{F1_{score}}$ as the performance evaluation indicators, calculated as follows formulas 17–19, where w_i represents the weight value of i^{th} category. Specifically, when calculating the Macro average, we initially assigned equal weights to each category. However, it is not appropriate to assign equal weights in the presence of sample imbalance. Instead, a more suitable approach would be assigning different weights to each category based on its respective sample size.

$$Weighted_{Precision} = \sum_{i=1}^{n} w_i P_i \tag{17}$$

$$Weighted_{Recall} = \sum_{i=1}^{n} w_i R_i \tag{18}$$

$$Weighted_{F1_{score}} = \sum_{i=1}^{n} w_i F1_i \tag{19}$$

5.3. Results and discussions

We compared our model with the following baseline models, and the results are shown in Table 3:

- TextCNN (Re-training from scratch) [30]: Encodes text using multiple sizes of convolutional kernels to capture semantic information at different levels.
- BiLSTM (Re-training from scratch) [31]: Composed of forward and backward LSTMs to better capture long-range dependencies in the text.

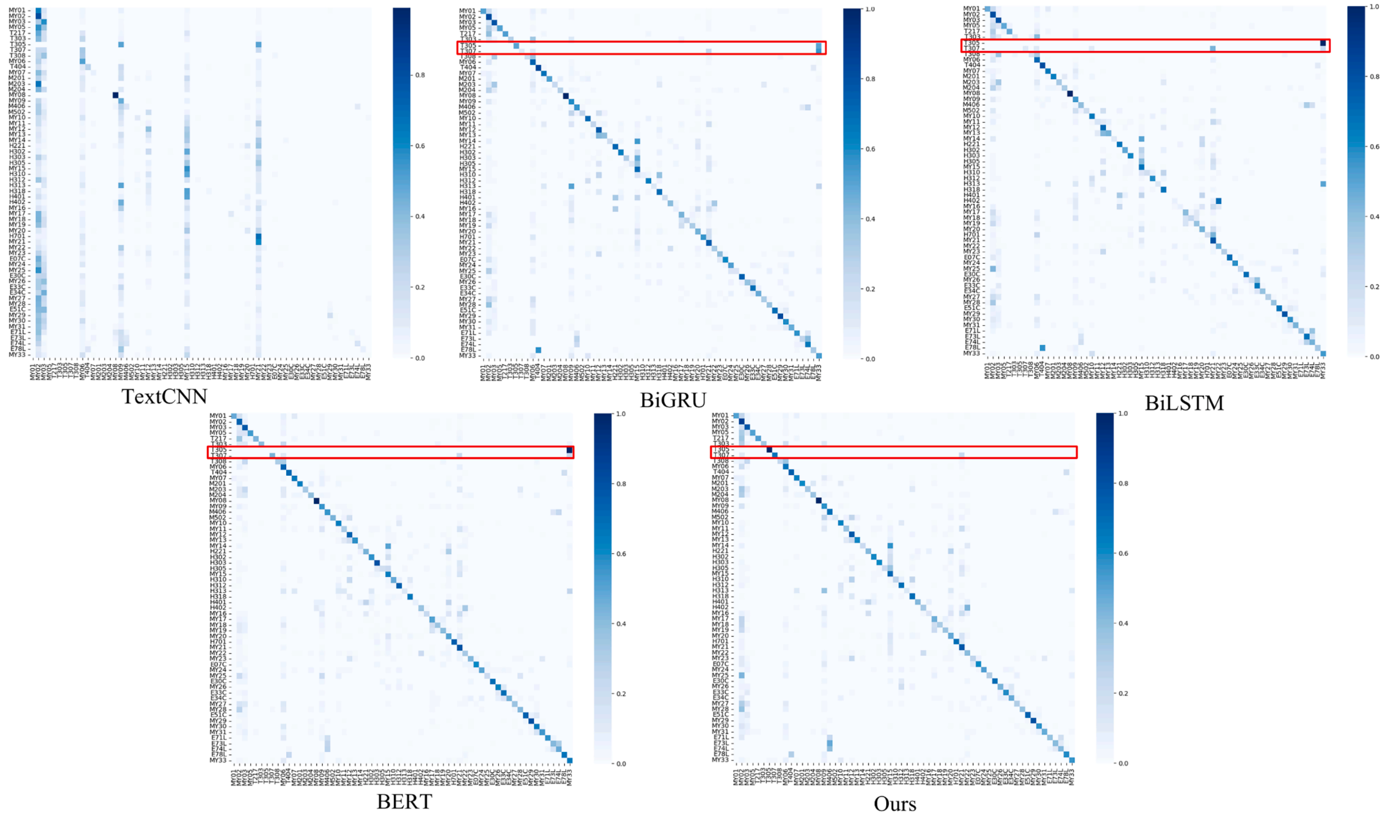
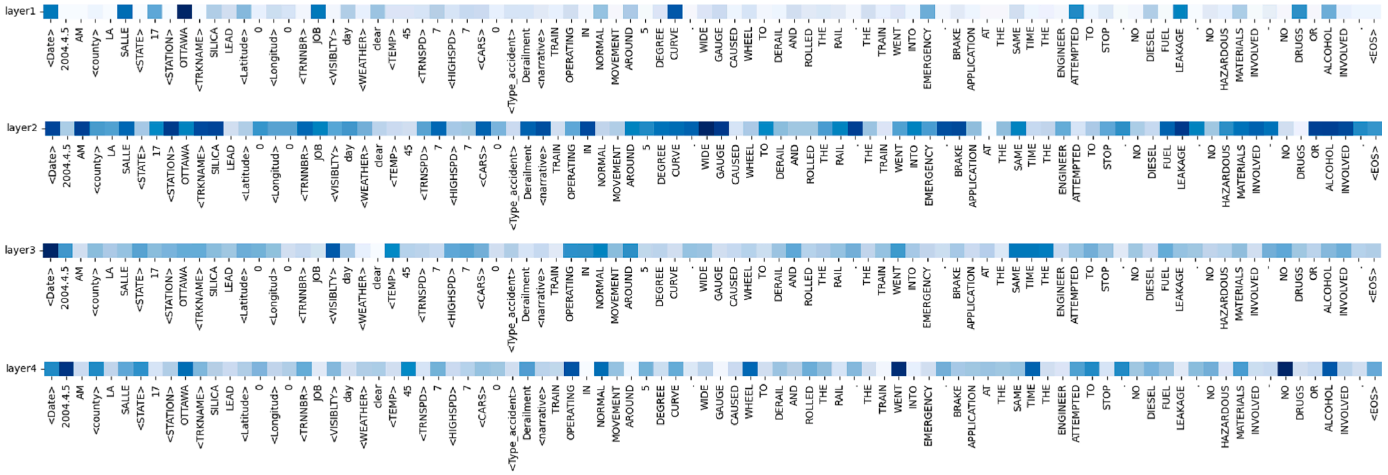

TextCNN
BiGRU
BiLSTM
BERT
Ours

Fig. 7. Results of the confusion matrix comparison for the models.



Accident Description: <Date> 2004.4.5 AM <county> LA SALLE <STATE> 17 <STATION> OTTAWA <TRKNAME> SILICA LEAD <Latitude> 0 <Longitud> 0 <TRNNBR> JOB <VISIBLTY> day <WEATHER> clear <TEMP> 45 <TRNSPD> 7 <HIGHSPD> 7 <CARS> 0 <Type_accident> Derailment <narrative> TRAIN OPERATING IN NORMAL MOVEMENT AROUND 5 DEGREE CURVE . WIDE GAUGE CAUSED WHEEL TO DERAIL AND ROLLED THE RAIL . THE TRAIN WENT INTO EMERGENCY . BRAKE APPLICATION AT THE SAME TIME THE ENGINEER ATTEMPTED TO

Fig. 8. Distribution of word importance in SAM layer.

- BiGRU (Re-training from scratch) [32]: Composed of forward and backward GRUs to encode text in both directions, preserving the advantages of GRU and further learning contextual information.
- BERT (Version: Bert-Base-Uncased) [33]: Bidirectional Encoder Representation from Transformers, a pre-trained language representation model introduced by Google.

In Table 3, We used the average value of Macro and Weighted as the final evaluation results. From the table, it can be seen that our model achieved the best performance in all three evaluation metrics. Compared with the best baseline model BERT, our model improved by 3.56, 0.42, and 0.76 in Avg-Precision, Avg-Recall, and Avg-F1-score, respectively.

The confusion matrices for each baseline model and our model were plotted to evaluate the performance of the model in each category, as depicted in Fig. 7. A darker color on the diagonal indicates better accident predictions for that specific category. The results presented in Fig. 7 show that the TextCNN model exhibited the poorest performance, as it failed to accurately predict each accident category and produced scattered predictions randomly. In contrast, BiGRU, BiLSTM, BERT, and our model demonstrated superior predictive capabilities.

However, in accident categories with relatively few training data, such as T305 (Retarder worn, broken, or malfunctioning) and T307 (Spring/power switch mechanism malfunction) shown in Fig. 7, BiGRU, BiLSTM, and BERT models all fail to achieve satisfactory results. The error rates are relatively high, and they often misclassify them as MY33 (Classification Yard Automatic Control System Failure). The analysis of this issue is as follows: MY33 represents a collective term for four types of accidents, namely, Classification yard automatic control system switch failure, Classification yard automatic control system retarder failure, Classification yard automatic control system - Inadequate or insufficient control, and Power switch failure. The descriptions of accidents in T305 and T307 often contain words such as "switch," "retarder," and "power." Due to the small number of accidents and the lack of detailed descriptions in the dataset, the models easily become confused during prediction, leading to incorrect judgments. In contrast, our model performs well predicting these two types of accidents. This indicates that our model can effectively discover meaningful features in the imbalanced accident data with a few instances, enabling accurate predictions. It also indirectly validates the effectiveness of the Lexical Encoder CBAM.

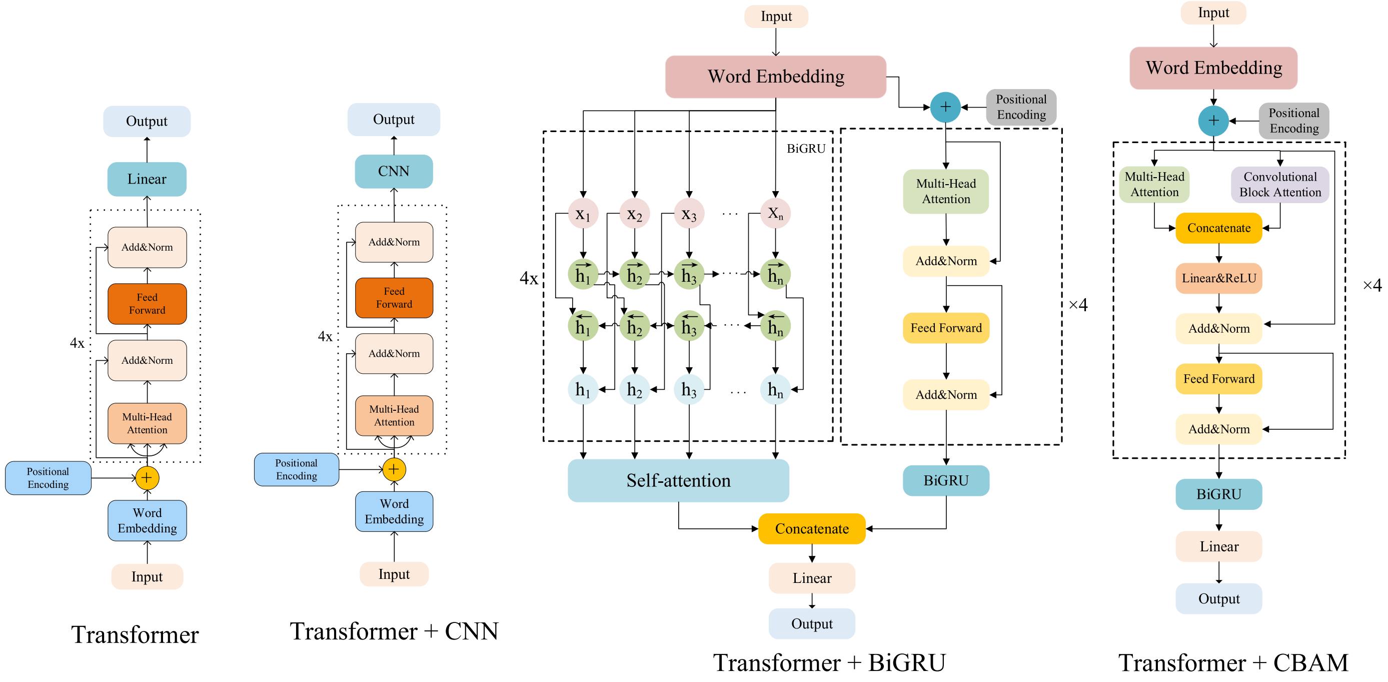
Compared to the best baseline model, our model achieved improvements of 3.56%, 0.42%, and 0.76% in Avg-Precision, Avg-Recall, and Avg-F1-score, respectively. We investigated the reasons behind these results, considering that our task involves predicting the causes of railway accidents, where various factors exhibit a discrete distribution. It becomes challenging to represent complete contextual dependencies due to the discrete nature of accident-related factors. The BERT model excels in extracting complex dependencies between contexts, but in this scenario, BERT fails to demonstrate satisfactory performance. Additionally, our model utilizes Word2Vec embeddings, which, to some extent, focuses more on semantic understanding at the word level. Furthermore, the inclusion of the Convolutional Block Attention Module (CBAM) as a lexical Encoder in our model enhances word-level semantic understanding, making it better suited for the discrete distribution of railway accident factors.

In Table 3, the baseline model TextCNN exhibits the poorest performance, attributed to the limitations imposed by its convolutional kernels. Despite TextCNN's ability to consider relationships between texts, it is constrained by fixed convolutional kernels. As these kernels have a fixed scope, different words can only perceive surrounding words within a fixed stride, making it challenging to overcome these limitations. Moreover, railway accident factors are discrete, requiring consideration of numerous relationships between words. This limitation renders TextCNN essentially incapable of comprehending railway accident data. While TextCNN can broaden its field of view by increasing the number of convolutional layers, this strategy is effective in datasets with complete contextual relationships but less impactful in our discrete data context. In contrast, our model and the other three baseline models can focus on the entire input sentence, allowing them to grasp the meaning within railway accident texts to a certain extent. Consequently, they demonstrate significantly superior results compared to TextCNN.

In terms of interpretability, we also conducted a corresponding study. As depicted in Fig. 8, we visualized the attention scores $score_{sam}$ of the Self-Attention Mechanism (SAM) layer across all Lexical Encoder layers for an accident example. This visualization demonstrates the distribution of importance scores assigned to each word in different Lexical Encoder layers, indicating their significance in generating the judgment result. The intensity of color reflects the model's emphasis on specific words, thereby providing partial interpretability for prediction outcomes. In Fig. 8, the accident cause is identified as "Track Geometry Irregularities and Abnormal Track Gauge". Notably, from layer 2 onwards, our model exhibits a focus on the words "WIDE GAUGE", while in layer 4 it pays attention to the words "WHEEL". These focal points vary across different layers. By comprehensively analyzing these focal points and their relationship with the accident cause, our model infers that this particular accident was caused by "Track Geometry Irregularities and Abnormal Track Gauge". This interpretation offers valuable insights into predicting results.

Table 4
Ablation experiment results.

Model	Precision			Recall			F1-score		
	Macro	Weighted	Avg	Macro	Weighted	Avg	Macro	Weighted	Avg
Transformer	60.22	67.19	63.71	47.66	66.89	57.28	51.18	65.81	58.50
Transformer+CNN	57.89	66.95	62.42	48.05	67.12	57.59	50.43	66.09	58.26
Transformer+BiGRU	62.27	68.08	65.18	48.71	68.02	58.37	52.37	67.11	59.74
Transformer+CBAM	57.57	67.17	62.37	50.63	67.06	58.85	51.81	66.35	59.08
Ours	64.82	69.65	**67.24**	51.62	68.85	**60.24**	54.97	68.05	**61.51**



Transformer Transformer + CNN Transformer + BiGRU Transformer + CBAM

Fig. 9. Ablation experiment models.

To demonstrate the effectiveness of our model improvements, we conducted ablation experiments, and the experimental results are shown in Table 4. In the ablation experiments, the following models depicted in Fig. 9 were used:

- Transformer: The traditional Transformer-Encoder model.
- Transformer+CNN: Integration of CNN into the output part of the traditional Transformer-Encoder model.
- Transformer+BiGRU: Addition of our Context Information Encoder to the traditional Transformer-Encoder model.
- Transformer+CBAM: Inclusion of the CBAM attention model and the Semantics Encoding module in the traditional Transformer-Encoder model.

The results presented in Table 4 demonstrate that, when compared to the traditional Transformer-Encoder model, our enhanced model exhibits superior overall performance. This signifies the effectiveness of our model improvement strategy, which effectively addresses the limitations of the traditional Transformer-Encoder model and achieves optimal outcomes across various performance metrics.

6. Conclusions and future work

This study proposed a novel deep-learning model framework based on Transformer for accurate prediction of railway accident causes. To enhance the learning of semantic information within accident text, Convolutional Block Attention was introduced, while BiGRU was employed to address the deficiency of traditional Transformers in capturing position representation information and further capturing contextual information. This deep learning framework has potential applications in effectively addressing other prediction problems related to transportation safety in future scenarios.

The proposed model architecture is based on the traditional Transformer framework, with the incorporation of Convolutional Block Attention, BiGRU, and CNN. The model was trained using accident data from the U.S. FRA Rail Equipment Accident (REA) accident/incident database. To ensure effective training, keyword prompting techniques were employed during data preprocessing to facilitate better understanding and learning of the meaning associated with each discrete accident element by the model. For training purposes, 80% of the data was utilized while 10% was allocated for hyperparameter tuning and another 10% for model validation.

The performance evaluation demonstrated that our model outperformed several baseline models across various metrics. Compared to BERT, which served as our best baseline model, our proposed approach achieved improvements of 3.56%, 0.42%, and 0.76% in Avg-Precision, Avg-Recall, and Avg-F1-score respectively. Particularly noteworthy is its superior predictive performance when dealing with a small number of accident samples compared to baseline models' effectiveness. These results indicates that our model can effectively extract feature information even from a limited number of accident samples surpassing what can be achieved by baseline models.

The current study has limitations, as the data only covers a single cause for accidents and real railway accidents often have multiple factors involved. Additionally, the accident data table contained irrelevant elements and lacked descriptions, making it difficult for our model to learn meaningful knowledge. Therefore, there is a significant need for a high-quality accident dataset in predicting railway accident causes.

Future studies should focus on predicting accidents with multiple causes and collecting superior datasets from domestic and international sources. Our model has not been compared with other models in the field of railway accidents due to the lack of open-source models specifically designed for predicting causation. We will continue refining our model and expanding comparisons to include more specialized methods within this domain.

CRediT authorship contribution statement

Bin Jiang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization, Investigation, Resources. **Keming Wang:** Writing – review & editing, Resources, Methodology, Funding acquisition, Formal analysis, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This research was supported by the Natural Science Foundation of Sichuan province (No. 2022NSFSC0464), Chengdu Soft Science Research Project (2023-RK00-00084-ZF). Special thanks are given to the anonymous reviewers for their careful reading and insightful suggestions, which helped improve the quality of this work. The authors also thank Haochen Shi for his help suggestions for revision.

Appendix A. Categorization of accident cause

Code	Accident Cause
MY01	Track Foundation Issues
MY02	Track Geometry Irregularities and Abnormal Track Gauge
MY03	Track Break
MY05	Track Joint Issues
MY06	Switch and Turnout Equipment Defects
MY07	Extreme Environmental Conditions Accident
MY08	Accidents Caused by Road-Related Users
MY09	Foreign Object on Track or Wheelsets
MY10	Station Operation Issues
MY11	Railway Operation Issues Caused by External Human Interference
MY12	Ineffective Use of Brakes
MY13	Failure to Properly Secure the Engine
MY14	Radio Communication Failure
MY15	Push-Pull Train Operation Issues
MY16	Unauthorized Movement or Violation of Regulations
MY17	Excessive Buffing and Slack Action in Train Movement or Composition
MY18	Excessive Lateral Tractive Force on Curves
MY19	Violation of Train Operating Procedures
MY20	Train Overspeed
MY21	Improper Operation of Switch
MY22	Human Factors
MY23	Brake System Malfunction Issues
MY24	Underframe Structure Issues
MY25	Center Plate Issues
MY26	Coupling System Issues
MY27	Lateral Support Clearance Issues
MY28	Railway Vehicle Suspension System Issues
MY29	Bearing Issues
MY30	Rail and Wheel Fracture, Wear
MY31	Wheel Issues
MY33	Classification Yard Automatic Control System Failure
T217	Mismatched rail-head contour
T303	Guard rail loose/broken or mislocated
T305	Retarder worn, broken, or malfunctioning
T307	Spring/power switch mechanism malfunction
T308	Stock rail worn, broken or disconnected
T404	Catenary system defect
M201	Load shifted
M203	Overloaded car
M204	Improperly loaded car
M406	Fire, other than vandalism, involving on-track equipment
M502	Vandalism of on-track equipment, e.g., brakes released
H221	Automatic block or interlocking signal displaying a stop indication - failure to comply.
H302	Cars left foul
H303	Derail, failure to apply or remove
H305	Instruction to train/yard crew improper
H310	Failure to couple
H312	Passed couplers (other than automated classification yard)

(*continued on next page*)

(*continued*)

Code	Accident Cause
H313	Retarder, improper manual operation
H318	Kicking or dropping cars, inadequate precautions
H401	Failure to stop train in clear
H402	Motor car or on-track equipment rules, failure to comply
H701	Spring Switch not cleared before reversing
E07C	Rigging down or dragging
E30C	Knuckle broken or defective
E33C	Coupler retainer pin/cross key missing
E34C	Draft gear/mechanism broken or defective (including yoke)
E51C	Broken or bent axle between wheel seats
E71L	Traction motor failure (LOCOMOTIVE)
E73L	Oil or fuel fire (LOCOMOTIVE)
E74L	Electrically caused fire (LOCOMOTIVE)
E78L	Pantograph defect (LOCOMOTIVE)

Appendix B. Accident factors

Feature Name	Description
STATION	Nearest city and town
TRKNAME	Track identification
TRNSPD	Speed of train in miles per hour
WEATHER	Weather conditions
LATITUDE	Latitude in decimal degrees
LONGITUDE	Longitude in decimal degrees
RAILROAD	Railway code
TEMP	Temperature in degrees Fahrenheit
STATE	FIPS state code
TRNNBR	Train id number
INCDTNO	Railroad code (Reporting RR)
YEAR4	Four character year identification
MONTH	Month of incident
DAY	Day of incident
AMPM	am or pm
TYPE	Type of accident
CARS	Cars carrying hazmat
CARSDMG	Hazmat cars damaged or derailed
CARSHZD	Cars that released hazmat
VISIBLTY	Daylight period
TRNSPD	Speed of train in miles per hour
TYPSPD	Train speed type
TRNDIR	Train direction
TONS	Gross tonnage, excluding power units
HIGHSPD	Maximum speed reported for equipment involved
NARRLEN	Length of narrative
NARR1	Narrative
NARR2	Narrative
NARR3	Narrative
NARR4	Narrative
NARR5	Narrative
NARR6	Narrative
NARR7	Narrative
NARR8	Narrative
NARR9	Narrative
NARR10	Narrative
NARR11	Narrative
NARR12	Narrative
NARR13	Narrative
NARR14	Narrative
NARR15	Narrative

References

[1] J. Liu, F. Schmid, K. Li, et al., A knowledge graph-based approach for exploring railway operational accidents, Reliab. Eng. Syst. Saf. 207 (2021) 107352.

[2] The Guizhou bullet train derailed while ascending the platform last month, colliding with debris flow at the tunnel mouth 1 km away: what caused the derailment? [EB/OL]. https://www.163.com/dy/article/H91GCG9N05322ICO.html, 2022-06-04.

[3] K. Wang, X. Wang, Z. Wang, et al., Logical consistency verification of state sensing in safety-critical decision: A case study of train routing selection, IET Intelligent Transport Systems 16 (8) (2022) 1042–1057.

[4] W.T. Hong, G. Clifton, J.D Nelson, Railway accident causation analysis: current approaches, challenges and potential solutions, Accid. Anal. Prevent. 186 (2023) 107049.

[5] D.R. Zang, K.M. Wang, X.Y Feng, Causal analysis of high-speed railway accidents based on knowledge graph and fault tree, Railway Comput. Appl. 32 (7) (2023) 14–18.

[6] F. Ghofrani, Q. He, R.M.P. Goverde, et al., Recent applications of big data analytics in railway transportation systems: a survey, Transport. Res. Part C 90 (2018) 226–246.

[7] Railroad Equipment Accident [EB/OL] https://data.transportation.gov/dataset/Railroad-Equipment-Accident-Incident-Source-Data-F/aqxq-n5hy, 2023-11-22.

[8] M. Heidarysafa, K. Kowsari, L. Barnes, et al., Analysis of railway accidents' narratives using deep learning, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2018, pp. 1446–1453.

[9] R. Wang, X. Ma, L. Jia, et al., The method about the prediction and analysis of causes for railway accidents based on graph convolutional neural network//2023, in: 7th International Conference on Transportation Information and Safety (ICTIS), IEEE, 2023, pp. 496–501.

[10] L. Gao, H. Wu, Verb-based text mining of road crash report, in: TRB 92nd Annual Meeting, 2013.

[11] S. Das, A. Dutta, I. Tsapakis, Topic models from crash narrative reports of motorcycle crash causation study, Transp. Res. Rec. 2675 (9) (2021) 449–462.

[12] S. Das, S. Datta, H.A. Zubaidi, et al., Applying interpretable machine learning to classify tree and utility pole related crash injury types, IATSS Res. 45 (3) (2021) 310–316.

[13] K.M. Kwayu, V. Kwigizile, K. Lee, et al., Discovering latent themes in traffic fatal crash narratives using text mining analytics and network topology, Accid. Anal. Prevent. 150 (2021) 105899.

[14] B. Wali, A.J. Khattak, N Ahmad, Injury severity analysis of pedestrian and bicyclist trespassing crashes at non-crossings: a hybrid predictive text analytics and heterogeneity-based statistical modeling approach, Accid. Anal. Prevent. 150 (2021) 105835.

[15] M. Bareiss, C. Smith, H.C Gabler, Finding and understanding pedal misapplication crashes using a deep learning natural language model, Traffic. Inj. Prev. 22 (1) (2021) 169–172.

[16] X. Zhang, P. Srinivasan, S. Mahadevan, Sequential deep learning from NTSB reports for aviation safety prognosis, Saf. Sci. 142 (2021) 105390.

[17] Z. Lao, D. He, Z. Jin, et al., Few-shot fault diagnosis of turnout switch machine based on semi-supervised weighted prototypical network, Knowl. Based. Syst. 274 (2023) 110634.

[18] S. Soleimani, A. Mohammadi, J. Chen, et al., Mining the highway-rail grade crossing crash data: a text mining approach, in: 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2019, pp. 1063–1068.

[19] X. Zhou, P. Lu, Z. Zheng, et al., Accident prediction accuracy assessment for highway-rail grade crossings using random forest algorithm compared with decision tree, Reliab. Eng. Syst. Saf. 200 (2020) 106931.

[20] R. Bridgelall, D.D. Tolliver, Railroad accident analysis using extreme gradient boosting, Accid. Anal. Prevent. 156 (2021) 106126.

[21] L. Gao, P. Lu, Y. Ren, A deep learning approach for imbalanced crash data in predicting highway-rail grade crossings accidents, Reliab. Eng. Syst. Saf. 216 (2021) 108019.

[22] H. Meng, X. Tong, Y. Zheng, et al., Railway accident prediction strategy based on ensemble learning, Accid. Anal. Prevent. 176 (2022) 106817.

[23] B. Song, X. Ma, Y. Qin, et al., Railroad accident causal analysis with unstructured narratives using bidirectional encoder representations for transformers, J. Transport. Saf. Secur. 15 (7) (2023) 717–736.

[24] Y. Bengio, R. Ducharme, P. Vincent, A neural probabilistic language model, Adv. Neural Inf. Process. Syst. (2000) 13.

[25] Mikolov T., Chen K., Corrado G., et al. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[26] A. Vaswani, N. Shazeer, N. Parmar, et al., Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[27] S. Woo, J. Park, J.Y. Lee, et al., Cbam: convolutional block attention module, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3–19.

[28] Chung J., Gulcehre C., Cho K. H., et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.

[29] Transformer-based Model for Root Cause Analysis of Railway Accidents [EB/OL] https://github.com/chinajb2000731/TrainAccident, 2023-11-22.

[30] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1746–1751.

[31] Huang Z., Xu W., Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv :1508.01991, 2015.

[32] L. Zhou, X. Bian, Improved text sentiment classification method based on BiGRU-attention, J. Phys. 1345 (3) (2019) 0320.

[33] J. Devlin, M.-W. Chang, K. Lee, et al., Bert: pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT 2019, 2019.