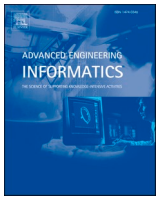
Advanced Engineering Informatics 65 (2025) 103126

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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



A spatial-temporal neural network based on ResNet-Transformer for predicting railroad broken rails

Xin Wang, Junyan Dai, Xiang Liu *

Department of Civil and Environmental Engineering, Rutgers, The State University of New Jersey, Piscataway, NJ, USA

ARTICLE INFO

Keywords:
Broken rail
Freight railroad
Spatial-temporal modeling
ResNet-Transformer
Time-independent data
Time-dependent data

ABSTRACT

Broken rails are a primary factor considered in railroad capital planning investments. This paper develops a spatial–temporal neural network model based on ResNet-Transformer architecture to predict the occurrence of broken rails one year in advance. The railroad data for this research includes infrastructure data, operational data, condition-related data, and maintenance activities. First, this research captures detailed spatial correlations and temporal dependencies, ensuring that each aspect is considered for its specific impact on rail integrity. Then, utilizing the ResNet architecture, the proposed model captures spatial correlations among static rail characteristics. Subsequently, the Transformer architecture is utilized for effectively handling long-term temporal data patterns and dependencies that reflect dynamic changes over time. An experiment was conducted based on railroad data collected from one major freight railroad covering about 20,000 miles of track spanning seven years, from 2013 to 2021. AUC values of the proposed model for the training, validation, and test set are 0.84, 0.81, and 0.81, respectively, demonstrating that the model has a relatively good performance and generalizes reasonably well to the validation and test set. The results indicate that the proposed model outperforms traditional machine learning approaches such as XGBoost, especially in identifying high-risk segments. When screening 10% of the highest-risk rail segments, the model can capture 41.6% of broken rails, compared to only 33.1% detected by XGBoost and 38.0% detected by ResNet-only model. This enhanced detection capability highlights the model's effectiveness in utilizing complex pattern recognition across both spatial and temporal data. The proposed spatial–temporal model not only aids in proactive maintenance to improve the safety and reliability of rail transportation but also contributes to more strategic capital planning in the railroad industry.

1. Introduction

Rail transportation contributes significantly to economic development and connectivity due to its efficient and cost-effective mode, particularly for transporting large volumes of goods over long distances. Broken rails are the main factor leading to freight-train derailments in terms of both the number of trains derailed and the number of cars derailed, which significantly jeopardizes the safety and efficiency of rail transportation [28]. In addition to causing severe accidents, broken rails can result in indirect losses due to service interruption during the repair and maintenance of rail infrastructure. According to the Federal Railroad Administration (FRA) accident records, derailment takes up 70.9 % of the average annual financial loss in the railroad network [4]. Given the substantial safety and economic implications, broken rails have been a primary consideration in railroad capital planning investments, such as the determination of the inspection intervals and maintenance-of-way tasks that are usually planned one year in advance. Therefore, the prediction of broken rail risk is of great interest and practical value, which can assist railroad staff in preventive maintenance that aims to reduce risk and service disruptions by informing the optimal planning of inspection and maintenance activities. In this way, rail transportation reliability and efficiency can be enhanced, as well as increase economic competitiveness within the industry.

Broken rails can be caused by a combination of infrastructure, operational, condition-related, and environmental factors. Infrastructure-related causes, such as curvature [17], grade [19], rail [14], turnout [7], and signal information [37], have been identified as significant contributors to rail fractures. For instance, curvature affects the lateral forces exerted on the rails, particularly in small curves with greater stress concentrations, leading to a higher probability of rail fatigue and wear [17]. Similarly, the grade of the track impacts the distribution of force along the rails, with steeper gradients imposing greater

* Corresponding author.
E-mail address: xiang.liu@rutgers.edu (X. Liu).

https://doi.org/10.1016/j.aei.2025.103126
Received 13 August 2024; Received in revised form 31 October 2024; Accepted 8 January 2025
Available online 13 January 2025

strain on the rail [19]. Operational factors consist of the traffic volume [6] and the maximum allowable speed of the vehicles [9]. Traffic volume directly influences the frequency and intensity of train passages, determining the cumulative loading on the rails over time. Consequently, higher traffic volumes are usually associated with accelerated deterioration of the track infrastructure [6]. Similarly, faster operating speed corresponds to a higher probability of rail failure due to increased dynamic force between the wheel and rail. Condition-related factors refer to track geometry exceptions [31], historical rail defects and broken rail detected [24,38], and vehicle-track interactions (VTI) [8]. These factors provide valuable insights into the current state of the track system and its susceptibility to failures. Track geometry exceptions can lead to significant increases in the dynamic loads on rails, decreasing the rail service life [31,36]. Furthermore, researchers found that broken rails are more likely to occur in cold seasons as tensile thermal stresses reach peak values in cold temperatures [27,43].

Due to the significant safety and economic impacts, many researchers have employed data-driven approaches to predict the occurrence of broken rails to guide proactive measures and ensure the safety and reliability of rail transportation. Early prediction of broken rail can be found in Dick et al. [13]. They developed a multivariate statistical model based on logistic procedures to quantitively predict the probability of broken rails. Later, a multilayer perceptron (MLP) neural network was proven to outperform the logistic regression model in terms of accuracy [33]. A fuzzy model [35] and a Markov model [2] were developed to predict the frequency of broken rails over long track sections. They can provide good prediction performance with basic factors that are relatively easy to collect. A defect-based risk analysis methodology for estimating broken rail risk was developed [23], where the crack information served as the index for evaluating the squat severity. Survival analysis allows researchers to model the time until an event occurs, such as rail failure, while accounting for censoring (i.e., rails that have not failed by the end of the study period). Thus, Ghofrani et al. [18] utilized the survival analysis approach to study the impact of covariates on rail life defined by the total cumulative tonnage, identifying factors (e.g., past rail defects and past geometry defects) that can significantly affect rail durability. Data treatment methods were incorporated into the gradient boosting classifier to address the imbalanced classification issue in predicting broken rail occurrences [19,37]. A feature-level attention-based bidirectional recurrent neural network (RNN) was proposed to enhance the performance of rail break prediction by capturing the temporal dependencies within the in-series train data, where the time-series generative adversarial network was employed to mitigate imbalanced data problem [42].

With the development of artificial intelligence, deep learning algorithms have been widely utilized and demonstrate their capability in handling various tasks such as sequential data analysis [11], image recognition [12,21], and natural language processing [34,25]. Neural networks have various architectures, each with unique strengths in processing data. MLP is a type of feed-forward neural network, where each neuron in one layer is fully connected to every neuron in the subsequent layer. It is commonly utilized for feature-based data due to its simplicity and efficient processing capabilities [33]. Convolutional neural network (CNN) is designed for structured grid data such as image and video data by conducting convolution operations to acquire local spatial dependence in the data [12]. It can also be applied to analyze multi-channel sequential data such as traffic prediction [3] and demand prediction [44]. Nevertheless, as networks go deeper, traditional CNN architecture suffers from vanishing gradients, which can hinder the optimization process and make it challenging to train deep networks effectively. ResNet (Residual Network), a type of CNN-based model, was proposed to address this issue by introducing residual connections [20]. It has demonstrated state-of-the-art performance in large-scale image classification tasks and sensor data analysis [15,39].

LSTM is a type of RNN architecture designed to address the vanishing gradient problem in traditional RNNs by learning long-term dependencies in sequential data by maintaining a memory cell and selectively updating it over time [22]. It can be used for tasks associated with long sequential data such as speech recognition and time series prediction. However, fluctuations and dependencies between distant elements pose significant challenges for traditional models like RNNs and CNNs. With the advent of transformer architecture, a revolutionary shift occurred in NLP and sequential data processing [34]. Transformers are designed to efficiently capture long-range dependencies in sequential data through self-attention mechanisms. Unlike RNNs, where computations of input data are inherently sequential, Transformers process the entire sequence simultaneously, enabling parallelization and alleviating the sequential bottleneck. Additionally, the self-attention mechanism is capable of weighing the importance of each element in a sequence concerning all other elements. This allows them to focus on relevant parts of the input sequence, such as sudden changes in the data. This is particularly important for attenuating the impact of noise or irrelevant information and can be useful for sensor data where distant data points may still be relevant. Transformer demonstrated promising results in various time-series tasks including anomaly detection, regression, and classification by accounting for both short-range and long-range dependencies and interactions [40,1]. For example, PatchTST addresses the memory and computational limitations of Transformers in time series by using a patching strategy, allowing it to effectively handle long look-back windows. This design significantly improves the model's accuracy in representing sequential patterns and dependencies in time-series data [32]. In contrast, DSformer introduces a dual approach with double sampling and attention-based blocks to enhance local and global feature extraction across multiple variables, which is especially useful for multivariate time series predictions, such as traffic or weather forecasting [41].

To combine the strengths of different network architectures, the hybrid models have attracted continuous attention from researchers and engineers, which achieves enhanced performance across diverse tasks. CNN-LSTM models were applied to analyze the geometry of railroad tracks, where CNN serves as the feature extractor to capture spatial patterns, while LSTM processes the sequential data [29,36]. However, due to the fixed architecture, CNN-LSTM may not adapt dynamically to different data characteristics, posing challenges in scalability and adaptability. Compared with CNN-RNN architecture, ResNet-Transformer hybrid architectures offer advantages in efficiently capturing long-range dependencies within the data while providing flexibility in architecture [45].

In the field of railroad engineering, recent advances in deep learning have significantly enhanced predictive capabilities for rail infrastructure, where various architectures and hybrid models have been explored to improve prediction performance and handle the complexity of the data. Modeling and predicting broken rails are critical parts of the advanced railroad management system and capital planning. However, accurate prediction of broken rails has been challenging due to the complex impact of both spatial and temporal dependencies. The characteristics of both central and adjacent track segments can impact the broken rail occurrence for segments of interest. In central segments, changes in the infrastructure, such as the degree of a curve or the presence of rail defects, can compromise rail integrity [17,24]. For instance, increased curve degrees can amplify the lateral forces exerted on the rails, accelerating wear and potentially leading to breaks. Existing rail defects in the central segment may propagate under the strain of these forces, increasing the likelihood of a rail break. Additionally, information from adjacent track segments can be crucial in predicting broken rails in a given segment. Considering the specific example of reverse curves (S-shaped curves), the abrupt change in rail direction between two curves can lead to increased dynamic stress [30]. This not only affects the segment directly but can also influence adjacent segments by altering the flow and distribution of forces along the track. Beyond the spatial dependencies, temporal dependencies also exacerbate the challenge of predicting broken rail. Rail conditions evolve over

time due to various factors such as geometry conditions and maintenance activities [19,37]. The failure to account for the dynamic change in rail conditions could hinder the ability to differentiate between normal segments and segments with recurring/transient patterns of broken rail.

The aforementioned studies typically employed feature integration techniques to combine related information over each track section. On one hand, this process generates appropriate input formats for data-driven models and can reduce computational costs. On the other hand, integration methods introduce subjectivity and lead to the oversight of crucial information pertinent to identifying broken rails. First of all, the spatial information of impact factors is ignored due to information combination. For example, when integrating the curve feature, either the maximum or average curve degree strategy could be applied to represent the curve information over a track section. The compound or reverse curve track comprising two or more simple curves with the same or opposite direction of curvature poses a significant challenge in this regard. This oversight of spatial dependence is particularly concerning for segments affected by intricate interplay influence factors, such as curve tracks with geometry exceptions. Furthermore, feature integration techniques overlook temporal variations. Although these techniques can summarize data over time, they may mask patterns of degradation or unusual occurrences that are crucial for predicting rail breaks. This masking effect can lead to models that are less sensitive to the onset of critical failures, resulting in late or missed predictions. While the spatial and temporal dependencies have not been explored in broken rail prediction, ResNet-Transformer presents a promising approach for this application due to its capability to capture intricate patterns in the data. They combine the residual connections of ResNet with the self-attention mechanisms of Transformers, enabling them to effectively model complex spatial and temporal relationships.

In summary, accurately predicting broken rail occurrences requires a model that can simultaneously account for the complex spatial and temporal dependencies present in rail infrastructure data. Previous studies on broke rail prediction usually relied on subjective feature extraction methods, which are not capable of capturing all relevant information and could introduce biases. They neglect spatial and temporal information necessary for comprehending the relationships among various factors influencing rail degradation. Addressing the dependencies is crucial for developing an accurate predictive model.

The ResNet-Transformer architecture is particularly well-suited to this task due to its ability to effectively capture both spatial and temporal dependencies. ResNet (Residual Network) excels in extracting deep spatial features by utilizing skip connections that mitigate the vanishing gradient problem, thus allowing the network to learn more intricate patterns in the data [26]. Meanwhile, the Transformer component is adept at handling sequential data and capturing long-range dependencies through self-attention mechanisms, which enables the model to weigh the importance of different time steps dynamically [5]. This combined architecture leverages the strengths of both ResNet and Transformer, making it a promising tool for accurately predicting broken rails by comprehensively analyzing the complex spatial–temporal interactions inherent in railroad data. Nevertheless, typical ResNet models usually generate long sequential outputs, which incur high computation costs when passing the corresponding outputs to the Tranformer model for further processing.

To address the above problems, this paper develops a spatial–temporal model based on pruned ResNet-Transformer for broken rail prediction, which effectively accounts for the impact of characteristics from adjacent track segments and historical sequential information. First, in an effort to overcome the problems associated with the consideration of spatial and temporal dependencies, this research utilizes microscale track segments to enable the capturing of track property variations (e.g., curve degree and the presence of turnouts) and incorporating of time-series input of track changes over time (e.g., rail defects and geometry exceptions over time). This method helps mitigate information loss during the subjective feature extraction process. Second, the pruned ResNet is employed to generate a reduced feature representation of the spatial data. By pruning the ResNet, we reduce the dimensionality of the output, which decreases the computational cost associated with processing long sequential outputs. This efficient representation maintains essential spatial features while discarding redundant information. Finally, the Transformer component processes the pruned features to capture long-range temporal dependencies. Through its self-attention mechanisms, the Transformer dynamically weighs the importance of different time steps, ensuring that both recent and historical data are integrated effectively into the prediction model. In summary, this integrated approach leverages the strengths of both ResNet and Transformer architectures, combining detailed spatial analysis with advanced sequential processing.

The main contributions of this paper are summarized as follows:

1) **Development of a Pruned ResNet-Transformer Model**: This paper introduces a novel spatial–temporal prediction model for broken rails that integrates a pruned ResNet architecture with Transformer.
2) **Enhanced Spatial and Temporal Feature Integration**: This study utilizes microscale track segments to capture detailed spatial variations in rail properties (e.g., curvature and turnout presence). Additionally, the proposed approach incorporates time-series changes of track to consider the temporal variations that influence broken rail occurrences.
3) **Mitigation of Information Loss**: The proposed method addresses the limitations of subjective feature extraction techniques by preserving crucial spatial and temporal information.
4) **Practical Implications for Railroad Management**: The proposed model provides a robust tool for railroad operators and maintenance planners to forecast broken rails more accurately.

2. Data Description

This research collaborates with a major U.S. freight railroad company to collect relevant data. The studied railroad data covers about 20,000 miles of track spanning nine years, from 2013 to 2021. It can be classified into four categories: operational data, infrastructure data, condition-related data, and maintenance activities, which are summarized in Table 1.

Infrastructure data offers insights into the layout and characteristics of the railroad network in this study. It encompasses parameters such as curve and grade data, which describe the horizontal and vertical alignment of the track. Additionally, rail data provides information on rail weight, installation dates, rail types (new or re-laid), and whether the rail is jointed or continuously welded. Turnout data describes

Table 1
Summary of Data Collected for Broken Rail Prediction.

Category	Data	Description
Infrastructure data	Curvature	Horizontal alignment of the track
	Grade	Vertical alignment of the track
	Rail	Rail weight, new rail versus re-laid rail, and laid date
	Turnout	Turnout location
	Signal	Location and type of traffic signal
Operational data	Traffic	Monthly tonnage and car pass data
	Speed (Track class)	Maximum allowable speed
Condition-related data	Rail defects	Detected rail defect occurrences
	Geometry exceptions	Track geometry exceptions
	Broken rails	Detected broken rail occurrences
	VTI exceptions	Vehicle-track interaction exception data
Maintenance activity	Ballast cleaning	Activities that remove debris, dirt, and worn-out ballast
	Grinding	Restore the profile and smoothness of the rails

turnout direction, frog type, size, and other relevant details. Signal data is a binary variable in our research, indicating whether specific track segments fall within signalized territory.

In terms of operational data, it provides essential information regarding the traffic and the speed of the railroads. The traffic data specifies the monthly gross traffic tonnage and the number of passing cars for each track segment. The speed data indicates the maximum allowed speed (measured in miles per hour) of the train passing by each track segment.

Condition-related data in this study includes the track geometry exception, VTI exception, rail defects, and broken rail data. Track geometry represents the geometric data of the track, including profile, alignment, cross-level, warp, and gauge. Once the amplitude of a specific geometry measurement (e.g., warp) exceeds the corresponding threshold in the FRA's Track Safety Standards [16], it is defined as a geometry exception. Then, the relevant features of the exception including the exception type, found data, and location information are in the geometry exception database. On the other hand, The VTI system measures the car body, truck frame, and axle accelerations during the operation of inspection vehicles. The VTI exception contains the location, found date, exception type, and exception priority. Broken rail is a special type of rail defect, which denotes that a rail is separated into two or more pieces. When a broken rail or rail defect is identified, the failure/defect-related information such as failure/defect type, track location, rail side, and found date would be investigated.

Finally, the maintenance activities encompass ballast cleaning and grinding, aiming at ensuring the smooth operation and integrity of the railroad infrastructure. Ballast cleaning involves removing contaminants from the track's ballast layer to maintain proper drainage and stability. Grinding focuses on restoring the rail profile to enhance ride quality and reduce wear on rolling stock. Following these activities, updated location information is recorded in the maintenance system to track maintenance efforts and their impact on the railroad's overall condition.

3. Methodology

A novel spatial–temporal deep learning methodological framework is proposed in this paper to predict railroad broken rails using multi-source data, as illustrated in Fig. 1. This framework mainly contains two parts: data preprocessing and model development. Data preprocessing generates the model's input and corresponding output. The model's input contains the spatial and temporal features to consider the impact of the adjacent track segments and historical railroad conditions. The prediction target is defined as the broken rail occurrence of the central track segment by the time of next year. Subsequently, in the model development phase, ResNet is applied to extract spatial information from the raw data while the Transformer captures the temporal correlations and sequential patterns in the latent representations.

3.1. Data preprocessing

This section generates the input and output of the deep learning model with consideration of both spatial and temporal features by incorporating railroad information from both adjacent track segments and historical data, as shown in Fig. 2. The data information from the adjacent track segments helps to identify the spatial impact of railroad characteristics on the broken rail. Historical railroad information provides insight into the data trends and patterns that develop over time. To this end, the whole railroad network is first delineated into micro track segments with identical lengths of 0.01 miles. This fine-grained segmentation ensures detailed spatial analysis and enables the model to

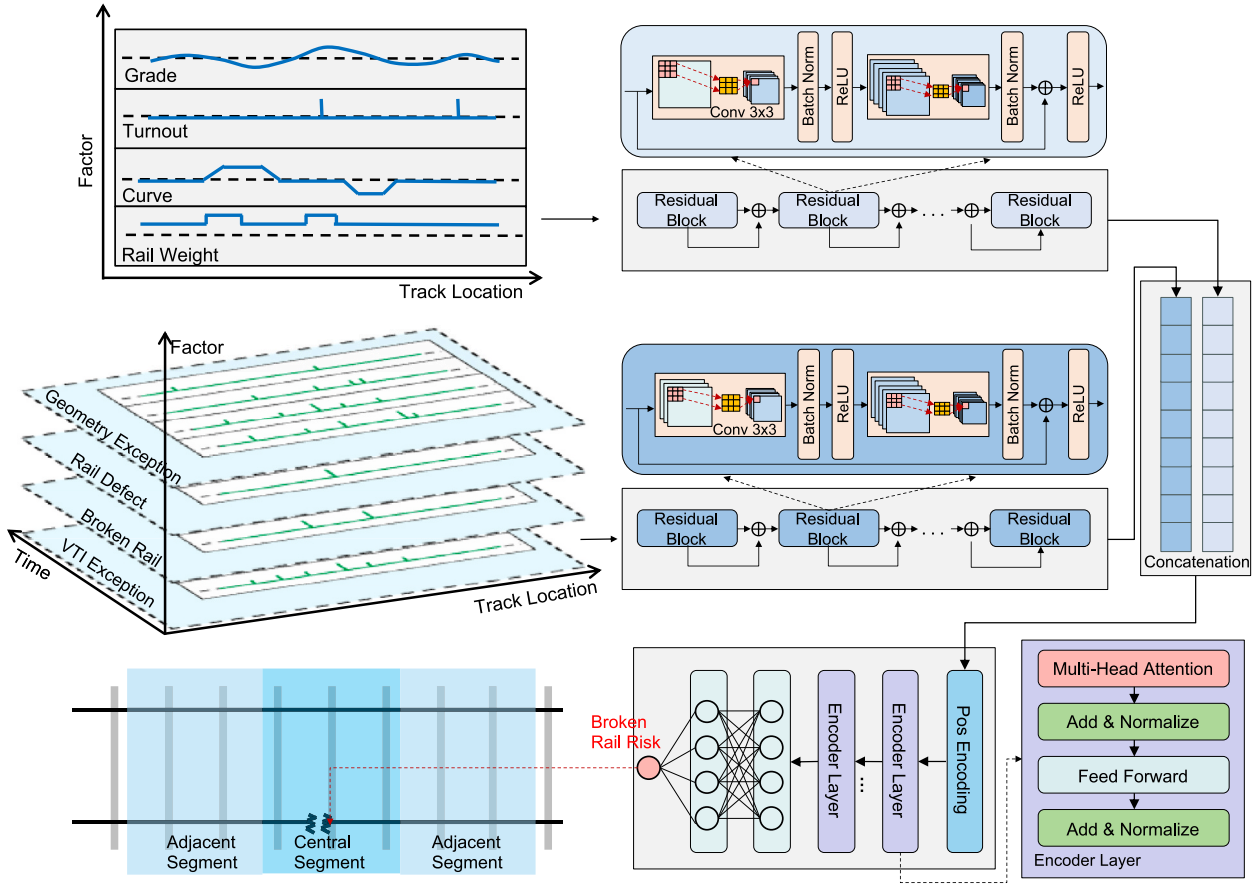


Fig. 1. Methodological Framework for Broken Rail Prediction.

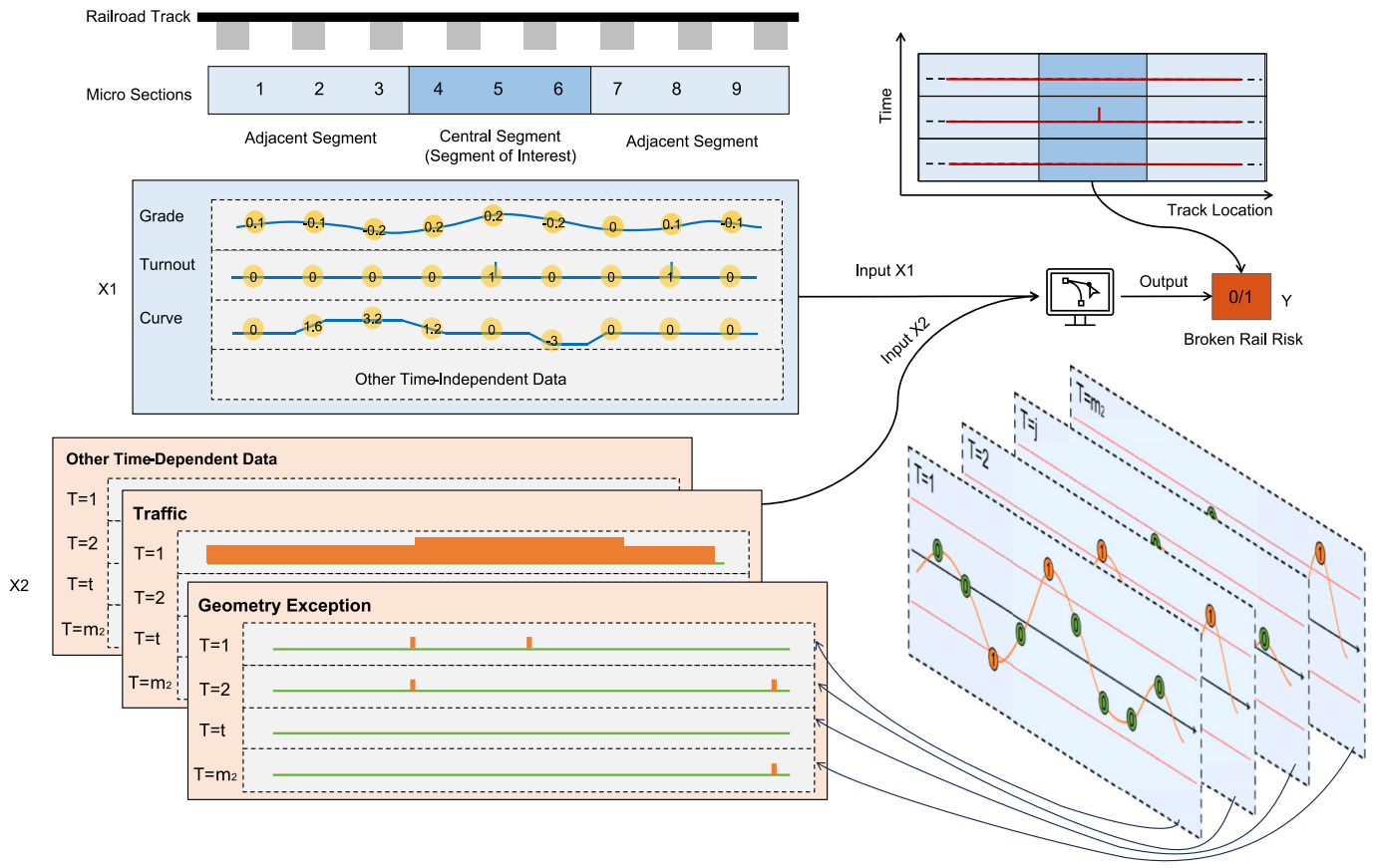


Fig. 2. Data Preprocessing for Preparing Inputs and Outputs of the Deep Learning Model.

consider changes in railroad conditions over short track sections.

This research divides the datasets into two types according to time attributes: time-independent data and time-dependent data are separately processed. Time-independent data refers to railroad data that typically remain constant over time, which includes seven railroad characteristics: curve degree, grade, signal type, number of turnouts, rail age, rail weight, and maximum allowed speed. Only spatial feature is considered when processing the time-independent data. To be specific, this study sets the length of the segment of interest to 0.1 miles. Besides, the neighboring track segments within 0.1 miles are integrated to account for spatial dependencies.

On the other hand, time-dependent data pertains to variables that change over time and are crucial for understanding the dynamic aspects of railroad operations and conditions. It comprises eight time-dependent railroad characteristics: traffic tonnage, number of cars, rail defect, geometry exception, broken rail, VTI exception, ballast cleaning, and grinding. Beyond the spatial dependencies mentioned above, temporal features are considered to include temporal patterns and trends that influence the occurrence of broken rails. Time-dependent data from the past three years is incorporated into the input for the deep learning model, recorded monthly to capture temporal fluctuations and trends.

Therefore, in this study, for a given dataset $D = \{(X1, X2, Y)\}$, Y is the prediction output of the model. $X1$ and $X2$ denote time-independent inputs and time-dependent inputs, respectively, as illustrated in Eq. (1)–(3).

$$X1 = \left\{ x1_{ij}^{p} | i = 1, 2, \cdots n_1; j = 1, 2, \cdots n_2 \right\} \tag{1}$$

$$X2 = \left\{ x2_{ijk}^{p} | i = 1, 2, \cdots m_1; j = 1, 2, \cdots m_2; j = 1, 2, \cdots m_3 \right\} \tag{2}$$

$$Y = \left\{ y_p | p = 1, 2, \cdots \right\} \tag{3}$$

where, $x1_{ij}^{p}$ ($x1_{ij}^{p} \in \mathbb{R}^{n_1 \times n_2}$) is a 2-D matrix representing the p^{th} instance of time-independent input $X1_p$. The dimensions n_1 and n_2 correspond to the number of micro track segments and the number of time-independent variables, respectively. In this research, $n_1 = 31$ and $n_2 = 7$. $x2_{ijk}^{p}$($x2_{ijk}^{p} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$) indicate a 3-D matrix denoting the p^{th} sample of time-dependent input $X2_p$. m_1 is the number of micro track segments, hence $m_1 = n_1 = 31$. m_2 is set to 36 accounting for the number of historical records in the past three years. $m_3 = 8$ shows the number of time-dependent variables. y_p is a binary variable indicating whether there is a broken rail associated with the segment of interest (central segment) by the time of next year.

3.2. Model development

A deep learning model is proposed for railroad rail prediction, utilizing two types of inputs and the binary outputs described previous section. The modeling process employs a ResNet-Transformer architecture designed to handle the spatial and temporal dependencies inherent in the data. Initially, the ResNet generates feature representations from the data. These representations are then fed into a Transformer architecture to capture temporal and long-term relationships in the data. The following sections illustrate the detailed spatial–temporal modeling process, followed by the loss function of the proposed model.

3.2.1. Spatial modeling

The ResNet architecture has proven its ability to extract detailed and hierarchical spatial features from complex datasets. It processes the data through residual blocks, which are the fundamental building blocks of the ResNet network. Each residual block contains two convolutional layers with the same number of output channels. Each convolutional layer is followed by a batch normalization layer and a Rectified Linear

Unit (ReLU) activation function. The design incorporates a skip connection that adds the input directly before the final ReLU activation function. Therefore, it can create very deep neural networks by allowing gradients to flow through the network more effectively during training, which addresses the problem of vanishing gradients. Nevertheless, traditional ResNet models such as ResNet50 have relatively long sequential encoded spatial representations (e.g., 2048). To minimize the model parameters and save computation costs, a pruned ResNet is implemented using ResNet34 architecture as the basic framework by reducing the layers, as shown in Fig. 3. We reduced the number of residual blocks, primarily based on empirical observations and domain-specific needs for handling railroad track data. The model uses a ResNet34 architecture as the base, reducing its layers and parameters to balance spatial feature extraction with computational feasibility. This pruned version maintains the essential characteristics of ResNet while being more efficient in terms of computational resources.

In this research, two types of input data, i.e., time-independent data $X1$ and time-dependent data $X2$, are separately processed using pruned ResNet34. First, the 2D convolutional operations $conv(\bullet)$ is utilized to extract the multi-scale features and maintain spatial hierarchies by conducting elementwise multiplication of filter weights with the input data, as shown in Eq. (4)

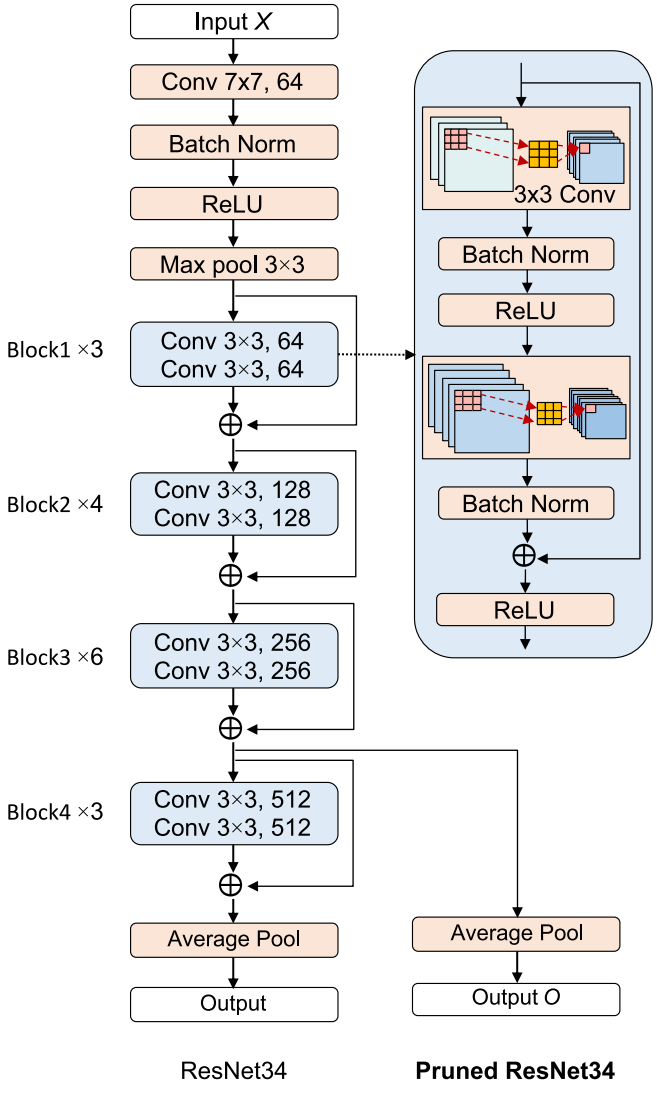


Fig. 3. Architecture of the Proposed Pruned ResNet.

$$H = Conv(X) = \sum_{p \in M} X \times C + b \tag{4}$$

where X denotes either input of $X1$ or $X2$, M is the feature map, C and b represent convolutional kernel and bias term, respectively. Then, the batch normalization operation $BN(\bullet)$ is applied to normalize the activations to improve training stability and performance. Thus, we can have the output of the first convolutional layer Z_1, calculated using Eq (5)–(7).

$$Z_1 = f(BN(H)) = f(\gamma \widehat{H} + \beta) \tag{5}$$

$$\widehat{H} = \frac{H - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{6}$$

$$f(u) = \max(0, u) \tag{7}$$

Where, $f(\bullet)$ is the ReLU activation function of the convolutional layers. $BN(\bullet)$ denotes the batch normalization operation. $\widehat{H}$ is the normalized input. ϵ is a small constant added for numerical stability. σ^2 and μ represent the variance and the mean of the input H. γ and β are learnable parameters that scale and shift the normalized value.

Next, the result Z_1 is passed into the other convolutional layer and batch normalization layer to generate the corresponding output Z_2. This step ensures that the feature maps are appropriately scaled, which facilitates effective gradient flow through the network.

$$Z_2 = BN(conv(Z_1)) \tag{8}$$

Finally, a skip connection is introduced by adding the original input X to the output of the second convolutional layer, Z_2. This addition helps to mitigate the vanishing gradient problem by allowing the gradient to flow more directly through the network. The combined output is then passed through another ReLU activation function $f(\bullet)$ to complete the residual block. The final output O of this process is given by:

$$O = f(Z_2 + X) \tag{9}$$

Notably, when using time-independent data $X1$ and time-dependent data $X2$ as inputs, the final outputs correspond to $O1$ and $O2$, which share the same dimension—a 1D vector with a length of 256. $O1$ encapsulates the spatial information derived from the static features of the railroad. This allows for the detailed analysis of the railroads' structural properties, which are crucial for assessing its long-term stability and integrity. $O2$ focuses on the dynamic aspects of the railroads, such as fluctuations in traffic load and varying track conditions. Therefore, the pruned ResNet34 integrates both static and dynamic spatial features, setting a robust foundation for the subsequent temporal modeling using the Transformer architecture.

3.2.2. *Temporal modeling*

In the temporal modeling phase, the objective is to capture the sequential patterns in the data. The Transformer architecture is utilized for its superior performance in handling sequential data and capturing long-term dependencies. The time-independent and time-dependent outputs (i.e., $O1$ and $O2$) are combined into a new representation as the input of the Transformer part, as presented in Eq. (10)

$$X' = concat(O1, O2) \tag{10}$$

where $concat(\bullet)$ denotes concatenation operation that integrates the vector horizontally. X' ($X' \in \mathbb{R}^{256\times 2}$) is the input of the Transformer with length of 256 and 2 types of features.

Transformers process sequences in parallel rather than sequentially (like RNNs do), meaning they don't inherently "know" the position of each data point in the sequence. To capture the sequential nature of the data, positional encoding is introduced to the Transformer model. It

provides each element in the sequence with unique positional information, enabling the model to interpret the order of temporal data effectively. This research applies a sinusoidal function as the position encoding function introduced by Vaswani et al. [34]. The sinusoidal approach to positional encoding offers two key benefits. First, it provides encoding continuity, where the smooth, continuous nature of sinusoidal functions allows the model to interpret gradual transitions across sequence positions, which is essential for capturing long-range dependencies in temporal data. Second, it ensures interpositional consistency by using different frequencies for sine and cosine functions. This feature enables the model to learn and generalize relationships between positions based on their relative distances, helping it detect patterns over varying time lags, even for unseen positions during training. The positional encoding process is defined as follows:

$$PE(pos,i)=\begin{cases} sin\left(\dfrac{pos}{10000^{\frac{2i}{d}}}\right) if i is even \\ cos\left(\dfrac{pos}{10000^{\frac{2i}{d}}}\right) if i is odd \end{cases} \tag{11}$$

where, pos is the position index in the sequential data, i is the dimension index, and d is the total number of dimensions. The function generates a matrix where each position is encoded with a unique combination of sine and cosine functions. This position matrix is added to the input embeddings X' to provide information about the positions of elements in the sequence.

$$X'_{enc}=X'+PE(pos,i) \tag{12}$$

where X'_{enc} denotes input tensor with position information. Subsequently, X'_{enc} is passed through several Transformer blocks, each consisting of a multi-head self-attention mechanism and a feed-forward neural network. The multi-head self-attention mechanism calculates attention scores for different parts of the sequence, allowing the model to focus on relevant information from various positions. Its implementation involves multiple layers of normalization, attention, and feed-forward networks with residual connections. For head h, the attention mechanism computes queries Q_h, keys K_h, and values V_h from the input tensor X'_{enc} using learned projection matrices, which are computed as Eq. (13)–(15).

$$Q_h=X'_{enc}W_Q^h \tag{13}$$

$$K_h=X'_{enc}W_K^h \tag{14}$$

$$V_h=X'_{enc}W_V^h \tag{15}$$

where Q_h determines the current element for which attention weights are being calculated. K_h represents all elements in the sequence against which the current element is compared. V_h contains the actual information of the elements to be weighted and combined. The attention information for each single head s_h is then obtained by Eq. (16).

$$s_h=softmax\left(\frac{Q_h(K_h)^T}{\sqrt{d_k}}\right)V_h \tag{16}$$

where $softmax(\bullet)$ represents softmax function, and d_k denotes the dimension of the key (and queries). Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. The outputs from all heads are concatenated and linearly transformed to form the multi-head attention results S, as presented in Eq. (17).

$$S=concat(s_1,s_2,\cdots,s_H)W_o \tag{17}$$

where W_0 is the output projection matrix, and H is the total number of heads. This research H is set to 8. Subsequently, a residual connection adds the input X'_{enc} to the attention output S, followed by layer normalization. The residual connections ensure that the model can effectively learn and propagate gradients during training. After passing through the Transformer blocks, the tensor is subjected to global average pooling to reduce its dimensionality and summarize the sequence information. This step aggregates information across the entire sequence, resulting in the pooled output tensor O_{pool}, as presented in Eq. (18)–(19).

$$O_{pool}=pool(O_s,C_{pool}) \tag{18}$$

$$O_S=LN(X'_{enc}+DP(S)) \tag{19}$$

where O_S is the result of each Transformer block, and O_{pool} denotes the final output of Transformer part. $pool(\bullet)$ represents the average pooling function. C_{pool} is the pooling size of the maximum pooling function. $LN(\bullet)$ denotes the layer normalization function, which normalizes the inputs across the batch dimension. It computes the mean and variance for each feature across all samples in a mini-batch. $DP(\bullet)$ is the dropout function that randomly set a fraction of input units to zero at each update during the training process, which prevents overfitting.

The pooled output O_{pool} is then fed into MLP for final classification. The MLP consists of multiple dense layers and dropouts for regularization. The final output layer uses a sigmoid activation function to produce the binary classification output, as shown in Eq. (20).

$$\hat{y}_p=f_s(DP(O_{pool})) \tag{20}$$

where $\hat{y}_p$ is the prediction output of the proposed model. $f_s(\bullet)$ denotes the sigmoid activation function that fully connects the corresponding input to a single unit ranging from 0 to 1.

3.2.3. Loss function

The training process of proposed deep learning models focuses on minimizing the difference between the predicted and actual labels. In the context of railroad engineering, broken rail events are rare, meaning that the majority of track segments do not experience broken rails. This leads to a highly imbalanced dataset, which poses a significant challenge, as standard binary cross-entropy loss might result in the model becoming biased towards the majority class (non-broken rails). This bias could lead to poor forecasting of broken rails, which is critical because misclassifying broken rails can result in severe consequences such as derailments and accidents.

Given the scarcity of broken rail instances, addressing the class imbalance is essential to ensure that the model can reliably predict these high-risk events. To achieve this, a custom-weighted binary cross-entropy loss function is used. This custom function assigns a higher weight to the minority class (broken rails), highlighting the importance of accurately predicting these critical cases. By doing so, the model is trained to place greater emphasis on learning patterns associated with broken rails, improving its sensitivity to these rare but impactful occurrences. The custom-weighted binary cross-entropy loss function is formulated as Eq. (21).

$$\mathscr{L}=-\sum_{p=1}^{P}\left[\omega\cdot y_p\cdot log\left(\hat{y}_p+\varepsilon\right)+\left(1-y_p\right)\cdot log\left(1-\hat{y}_p+\varepsilon\right)\right] \tag{21}$$

where, $\mathscr{L}$ is the weighted binary cross-entropy loss function. y_p represents the true labels, with 1 indicating a broken rail and 0 indicating a non-broken rail in the p^{th} track segment. $\hat{y}_p$ denotes the predicted probabilities of having broken rail for p^{th} track segment. The term ε epsilon∈ is a small constant added to prevent taking the logarithm of zero. The custom weight (ω) is set to a higher value for the positive class (broken rails), which allows the model to focus more on critical instances during training.

4. Experiment results

This section first introduces the evaluation metrics and baseline model, XGBoost, which has been widely employed in previous studies. Then, the implementation details of training the proposed deep learning model are presented. Finally, the results of the proposed model are demonstrated and compared with previous studies from the literature review.

4.1. Evaluation metric and baseline model

Evaluation metric is a measure used to assess the performance of the deep learning model. In binary classification, results can be categorized into four types, forming what is known as a confusion matrix: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). In the context of railroad engineering, broken rail incidents are rare but have severe consequences, such as service interruptions or derailments. Misclassifications, particularly FN, can lead to missed detections, while FP can result in unnecessary inspections. Both of these outcomes have significant implications for rail safety and operational efficiency. An effective evaluation metric must balance the needs of both the majority class (intact rails) and the minority class (broken rails).

The Receiver Operating Characteristic (ROC) curve is a crucial tool for evaluating binary classifiers. It plots the True Positive Rate (TPR, also known as sensitivity) against the False Positive Rate (FPR) across various threshold values. The ROC curve helps visualize the trade-offs between TPR and FPR for different decision thresholds. The Area Under the ROC Curve (AUC) is a summary metric that provides a single scalar value representing the model's overall ability to discriminate between the positive and negative classes. The AUC value ranges from 0 to 1, with higher values indicating better performance. The AUC is computed using Eq. (22). In broken rail prediction, a high AUC value indicates that the model effectively distinguishes between broken and intact rails despite the class imbalance.

$$AUC = \int_0^1 TPR(FPR)d(FPR) \tag{22}$$

In addition to AUC, this study customizes an evaluation metric for the railroad industry to optimize inspection efforts by accurately identifying segments prone to defects. This metric aims to maximize the percentage of actual rail defects detected within the top-risk segments, thus enhancing inspection efficiency and safety. The custom metric is defined as Eq. (23)–(24).

$$E = \int_t^1 TPR(t)P'(t)dt \tag{23}$$

$$P(t) = \frac{TP + FP}{TN + FP + FN + TP} \tag{24}$$

where E represents the percentage of total broken rails correctly predicted when track segments with predicted risk probability greater than threshold t are classified as true positives. $P(t)$ denotes the percentage of the railroad inspected given threshold t. This metric underscores the importance of prioritizing high-risk segments for inspection, ensuring that the model not only discriminates well between broken and intact rails but also effectively targets segments most likely to have broken rails.

For comparative purposes, the XGBoost algorithm is selected as the baseline in this research. XGBoost is an ensemble learning technique that combines multiple weak models, typically decision trees, to create a strong predictive model [10]. It improves performance by iteratively adding models that correct the errors of the combined ensemble. XGBoost has been widely applied in various domains, including fault prediction, due to its robustness in handling large datasets and its ability to model complex relationships. Thus, using XGBoost as a benchmark allows for a direct comparison to evaluate the improvements and effectiveness offered by the proposed deep learning approach.

Notably, the proposed model is able to utilize the 2D matrix and 3D matrix as inputs to consider both time-independent and time-dependent variables, while XGBoost is typically used for tasks with feature-based data structures. Therefore, we used feature extraction techniques to convert the 2D and 3D matrices into feature-based representations suitable for XGBoost. This approach allows the comparison between the proposed deep learning model and the baseline XGBoost algorithm.

4.2. Implementation details

The proposed model was implemented using Keras version 2.11.0 on a server with an NVIDIA RTX 4090 GPU, an AMD Ryzen 3700X CPU @ 3.59 GHz, and 112 GB of system memory. Key hyperparameters, including the learning rate, batch size, and dropout rate, were selected to ensure effective training and prevent overfitting. Specifically, a learning rate of 0.0001 was chosen after evaluating performance across different rates (i.e., 0.001, 0.0005, and 0.0001), with 0.0001 yielding the best balance between convergence speed and model accuracy. Similarly, we analyzed batch sizes of 16, 32, and 64, with a batch size of 64 providing optimal GPU utilization and stable gradient updates. A dropout rate of 0.3 was selected to mitigate overfitting.

The model's parameters were optimized with the Adam optimizer, and an early stopping criterion with a patience value of 4 epochs was used to cease training if the validation loss did not improve for four consecutive epochs. Temporal splitting divided the dataset into training, validation, and testing sets in a 6:2:2 ratio for our experiments. The training and validation sets were used to determine the learnable parameters of the model, while the test set was reserved for evaluating the model's performance on unseen data.

4.3. Results

The proposed spatial–temporal neural network model was trained using training and validation sets. In each set, the inputs comprise both time-independent and time-dependent data, and the corresponding output is a binary variable indicating a broken rail event. Notably, it took 110 GB of system memory during data preprocessing to generate the entire input and output dataset. The proposed model utilizes 18 GB of GPU memory during the training process. The training and validation loss curves are shown in Fig. 4. It can be observed that after 2nd epochs, the model's validation loss showed no improvement over four successive epochs. Consequently, the model training process ceased at 6th epoch,

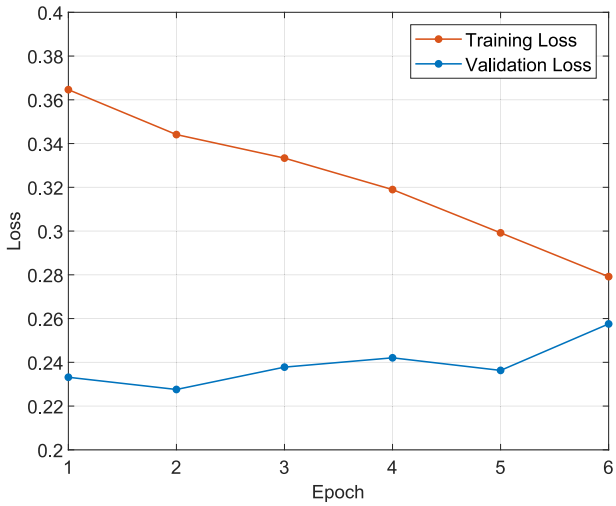


Fig. 4. Training and the Validation Loss Curve of the Proposed Model.

and the corresponding time needed for the model to converge (i.e., the whole training process) is 13,128 s. However, the inference speed of the trained model is much faster, taking 56 s to predict broken rails over the whole network (20,000 miles of railroad track).

Subsequently, the trained model was utilized to predict the broken rail probability in the test set to evaluate its generalization capability. AUC values for the training, validation, and test set are 0.84, 0.81, and 0.81, respectively, which demonstrates that the model has a relatively good performance and generalizes reasonably well to the validation and test set.

Further assessment of the model's performance was conducted using the customized evaluation metric *E*, which was specifically designed to optimize rail inspection efforts by focusing on segments most prone to broken rail. This metric maximizes the percentage of actual broken rails detected within the top-risk segments, enhancing both inspection efficiency and safety. The calculation of *E* involves integrating the *TPR* across a range of thresholds, weighted by the proportion of the railroad inspected at each threshold, to highlight the model's efficiency in identifying the most critical rail segments. Both the performance of the proposed model and XGBoost are presented in Fig. 5. It reveals that the proposed model effectively prioritizes high-risk segments, thereby maximizing the use of inspection resources. When 10 % of the railroad network is screened, 41.6 % of broken rails can be captured using the proposed model while the traditional machine learning approach (i.e., XGBoost) only captures 33.1 % of broken rails. This demonstrates the effectiveness of the methodology proposed in this paper for guiding the inspection and maintenance activities for the railroad. It also highlights the strengths of the spatial–temporal deep learning model in capturing complex patterns and dependencies that might be challenging for traditional machine learning algorithms like XGBoost to identify.

Additionally, the ablation experiment was conducted to evaluate the contributions of key components of our model, specifically focusing on the impact of the ResNet architecture. Given that inputs consist of both 2D and 3D formats to represent time-independent and time-dependent data, respectively, a Transformer-only structure would not align with the input data format requirements. As a result, a Transformer-only variant is not included in our ablation experiments. The same ResNet structure (Section 3.2.1) was employed to ensure consistency in our evaluations. The result indicates that the ResNet-only structure can capture 38.0 % of broken rails when 10 % of the network is inspected. While the ResNet architecture excels at extracting features from spatial data, it may not fully leverage the temporal dependencies inherent in the combined 2D and 3D input formats.

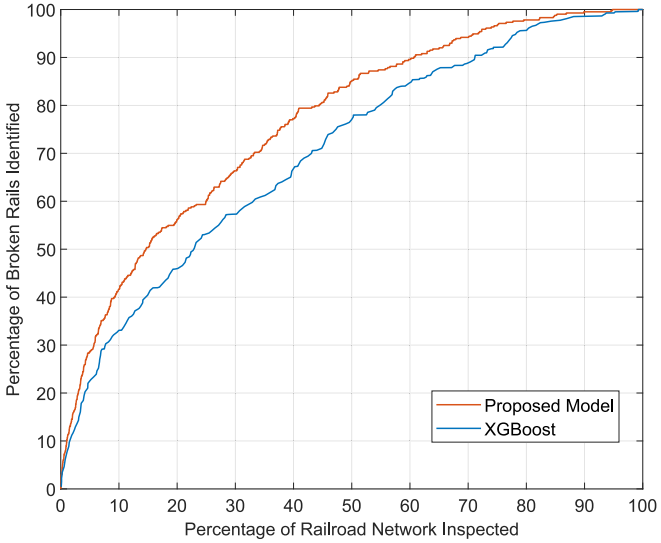


Fig. 5. Performance of the Proposed Model and XGBoost.

The spatial–temporal neural network model presented in this study allows for processing and learning from both static and dynamic rail characteristics, enabling the model to make more accurate predictions about rail failures. Time-independent characteristics, such as rail weight, curvature, and grade, provide a static snapshot of the infrastructure's inherent risks, influenced by design properties. These characteristics are crucial as they define the baseline conditions under which the rails operate. On the other hand, time-dependent characteristics, such as traffic volume, speed, and historical maintenance activities, introduce a dynamic aspect that reflects the operational environment's influence on rail integrity over time. By combining these datasets, the model gains a holistic view of the rail environment, enhancing its ability to detect patterns and anomalies that purely time-independent or time-dependent models might miss.

The utilization of both ResNet and Transformer architectures synergistically enhances the model's performance. ResNet helps mitigate the vanishing gradient problem, allowing the model to learn from deeper layers without performance degradation, which is crucial for capturing complex spatial relationships in static data. Transformers, renowned for their effectiveness in handling sequential data, give the model the ability to analyze sequences of time-dependent data across extended timeframes. This capability is crucial for understanding the temporal patterns and dependencies that affect rail conditions.

4.4. Application of research

The findings from this research have significant implications for the management of broken rails and can be extended to other predictive modeling problems. As an example of improving spot maintenance and management of broken rails, Fig. 6 illustrates the risk mapping over a portion of the studied railroad network. The map highlights segments with a higher probability of rail breaks in red, suggesting these areas as priorities for upcoming inspections and maintenance. This visual representation helps in strategic planning and allocation of resources, ensuring that the most vulnerable sections receive attention to prevent potential derailments and enhance overall safety. The proposed predictive model not only supports existing maintenance strategies but also introduces a proactive approach to rail management. Railroads can use predictive insights to prioritize and optimize maintenance schedules, focusing resources on the most critical areas. This targeted maintenance could ensure the best use of resources and reduce the incidence of unplanned maintenance and associated costs. By reducing emergency repairs and improving planning, the model directly contributes to significant cost savings and improves the overall economics of railroad operations. Additionally, the integration of the proposed predictive model enables rail companies to refine their asset management and capital planning decisions. By accurately forecasting rail conditions, companies can better allocate investments for infrastructure upgrades and replacements, thus managing long-term asset life-cycle costs more effectively.

In summary, our proposed spatial–temporal neural network model provides a significant advancement in railway maintenance technology. It not only supports existing maintenance strategies but also introduces a proactive approach to rail management. By identifying potential problem areas before failures occur, the model assists in transitioning from reactive to preventive maintenance strategies. This shift is expected to reduce downtime and associated costs, improve safety margins, and extend the lifespan of rail infrastructure.

Beyond the case of broken rail prediction, the ResNet-Transformer model offers a generalized framework that can be applied to various predictive modeling problems across different domains. Its ability to capture and analyze complex spatial–temporal dependencies makes it suitable for applications in areas such as predictive maintenance of other critical infrastructure (e.g., pipelines, highways, and bridges), weather forecasting, and even in healthcare for predicting patient outcomes based on a combination of spatial data (e.g., imaging) and temporal data

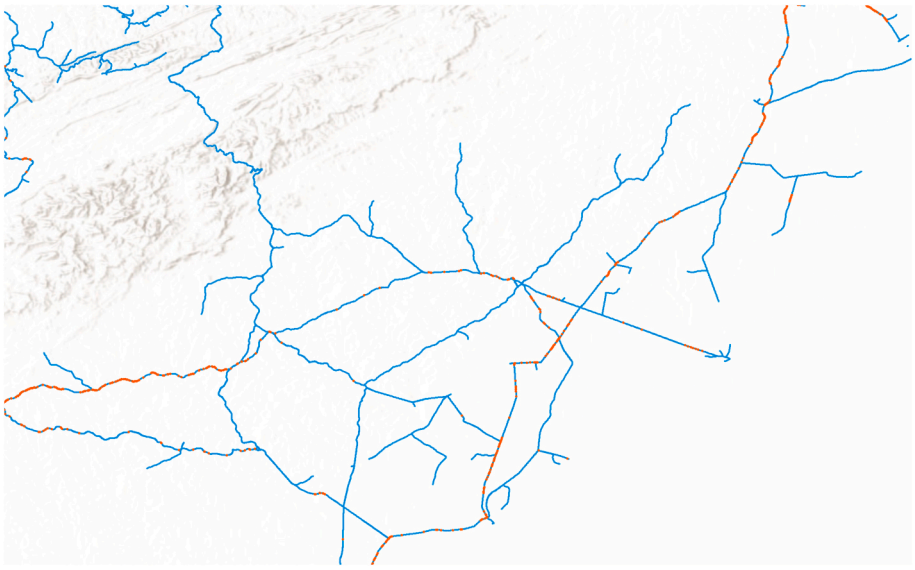


Fig. 6. Predicted Top 10% Risk (Red Segments) of Broken Rails over Part of Studied Railroad Network.

(e.g., patient history). The versatility and robustness of this model highlight its potential to address a wide range of predictive challenges, driving advancements and innovations in multiple fields.

5. Conclusion and future work

5.1. Conclusion

Predicting the occurrence of broken rails can assist railroads in proactive planning aimed at improving the safety and sustainability of rail transportation services while simultaneously minimizing the life-cycle costs of the rail. This study developed a spatial–temporal neural network model based on ResNet-Transformer architecture to predict broken rails one year in advance, integrating both time-independent and time-dependent characteristics of rail data. Time-independent data provides a foundational understanding of the rail infrastructure's static conditions, while time-dependent data captures dynamic and seasonal factors that influence the likelihood of rail failures. The effectiveness of the spatial–temporal model is validated using data collected from one major freight railroad covering around 20,000 miles.

In the data preprocessing phase, we utilized 2D and 3D data structures as inputs of the model to offer a detailed representation of the spatial correlation and temporal dependencies in the data. This allows for a comprehensive consideration of the impact that contributes to rail integrity.

In the modeling phase, the strengths of ResNet in processing spatial patterns and Transformers in managing temporal sequences were combined to create a robust model that effectively addresses both dimensions of the data. The ResNet architecture allows the model to learn the spatial relationships across the adjacent track segments, ensuring that the spatial context influencing the central track segment is effectively captured. The output from the ResNet consists of high-dimensional spatial feature representations, which are then aggregated to form a comprehensive spatial feature map for each track segment. This spatial feature map is subsequently passed to temporal modeling using the Transformer. The self-attention mechanism enables the model to weigh the importance of different time steps and track segments, effectively capturing both short-term fluctuations and long-term trends in the data. The network outputs a binary prediction indicating the likelihood of a broken rail occurrence in the central track segment by the next year. The integration of spatial and temporal modeling ensures that the framework leverages comprehensive insights from both dimensions, enhancing the accuracy and robustness of the broken rail prediction model. Additionally, as most track segments did not experience broken rails incurring imbalanced data issues, this research utilizes a weighted cross-entropy function as the loss function of the proposed model to measure the difference between actual rail conditions and prediction rail conditions.

The application of this hybrid model has demonstrated notable improvements over traditional machine learning methods, such as XGBoost, in predicting broken rails. For instance, when screening 10 % of the railroad network, the proposed model was able to detect 41.6 % of broken rails, a significant increase over the 33.1 % detected by XGBoost. This superior performance highlights the model's capability to pinpoint high-risk segments more accurately, thereby enhancing safety and optimizing maintenance operations. These results also emphasize the importance of considering spatial correlation and temporal dependencies through ResNet-Transformer to enhance the effectiveness of predictive models for broken rail prediction.

Furthermore, the findings from this study underscore the potential of advanced neural network models to transform rail maintenance strategies from reactive to proactive, ensuring that maintenance efforts are not only more effective but also more economically efficient. The proactive approach facilitated by the model helps prevent rail failures, thus minimizing disruptions and reducing the life-cycle costs associated with rail maintenance.

5.2. Limitation and future work

This section addresses the limitations of the current study and proposes directions for future research to enhance the spatial–temporal neural network model used for predicting broken rails.

One limitation of the current study is the data diversity and volume, as the model was primarily trained and tested using data from a specific major freight railroad. This may affect the model's generalizability across different rail networks that operate under varying conditions. Therefore, expanding the diversity of data sources, such as incorporating infrastructure data and operational data from different geographic regions, and varied rail network types, could also help improve the model's adaptability and accuracy. This would not only test the model's robustness but also enhance its generalization across globally diverse rail systems.

Additionally, due to computational constraints associated with the large datasets and the complexity of our spatial–temporal neural

network model, cross-validation was not implemented during the training process. This limitation restricts the ability to provide a comprehensive statistical analysis, such as the T test, to evaluate the stability and generalizability of model's performance. Instead, current research relied on a single training-validation split, which may not fully represent the model's robustness across different data subsets. In future work, we aim to address this limitation by exploring the feasibility of cross-validation to obtain more reliable estimates of model performance. This would enable us to conduct more rigorous statistical analyses and enhance our understanding of variability and significance in our findings.

Last but not least, the complexity and computational demands of the ResNet-Transformer architecture, while beneficial for capturing intricate spatial and temporal data patterns, require significant computational resources. This might limit the model's deployment in resource-constrained environments. Optimization of the model to reduce its computational load without compromising its predictive accuracy would make it more accessible and practical for broader applications. Improving the model's ability to integrate and process real-time data could significantly enhance its utility. Enabling real-time data processing would allow the model to respond more dynamically to changes in rail conditions, offering the potential to prevent failures as they develop, rather than relying solely on predictions based on historical patterns. This capability would shift the model's use from purely predictive to a more preventive rail maintenance tool.

CRediT authorship contribution statement

Xin Wang: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Junyan Dai:** Writing – review & editing, Validation, Software, Resources. **Xiang Liu:** Writing – review & editing, Validation, Supervision, Project administration, Funding acquisition, Conceptualization.

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

The authors do not have permission to share data.

References

[1] M.W. Ahmad, M.U. Akram, M.M. Mohsan, K. Saghar, R. Ahmad, W.H. Butt, Transformer-based sensor failure prediction and classification framework for UAVs, Expert Syst. Appl. 248 (2024) 123415, https://doi.org/10.1016/j.eswa.2024.123415.

[2] L. Bai, R. Liu, F. Wang, Q. Sun, F. Wang, Estimating railway rail service life: A rail-grid-based approach, Transp. Res. A Policy Pract. 105 (2017) 54–65.

[3] T. Bogaerts, A.D. Masegosa, J.S. Angarita-Zapata, E. Onieva, P. Hellinckx, A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data, Transp. Res. Part C Emerging Technol. 112 (2020) 62–77, https://doi.org/10.1016/j.trc.2020.01.010.

[4] R. Bridgelall, D.D. Tolliver, Railroad accident analysis using extreme gradient boosting, Accid. Anal. Prev. 156 (2021) 106126, https://doi.org/10.1016/j.aap.2021.106126.

[5] S. Chang, L. Wang, M. Shi, J. Zhang, L. Yang, L. Cui, Extended attention signal transformer with adaptive class imbalance loss for Long-tailed intelligent fault diagnosis of rotating machinery, Adv. Eng. Inf. 60 (2024) 102436, https://doi.org/10.1016/j.aei.2024.102436.

[6] G. Chattopadhyay, S. Kumar, Parameter estimation for rail degradation model, International Journal of Performability Engineering 5 (2009) 119.

[7] J. Chen, P. Wang, J. Xu, R. Chen, Simulation of vehicle-turnout coupled dynamics considering the flexibility of wheelsets and turnouts, Veh. Syst. Dyn. 61 (2023) 739–764, https://doi.org/10.1080/00423114.2021.2014898.

[8] M. Chen, Y. Sun, W. Zhai, High efficient dynamic analysis of vehicle–track–subgrade vertical interaction based on Green function method, Veh. Syst. Dyn. 58 (2020) 1076–1100, https://doi.org/10.1080/00423114.2019.1607403.

[9] R. Chen, J. Chen, P. Wang, J. Fang, J. Xu, Impact of wheel profile evolution on wheel-rail dynamic interaction and surface initiated rolling contact fatigue in turnouts, Wear 438–439 (2019) 203109, https://doi.org/10.1016/j.wear.2019.203109.

[10] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in, in: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794, https://doi.org/10.1145/2939672.2939785.

[11] T. de Bruin, K. Verbert, R. Babuska, Railway track circuit fault diagnosis using recurrent neural networks, IEEE Trans. Neural Networks Learn. Syst. 28 (2017) 523–533, https://doi.org/10.1109/TNNLS.2016.2551940.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L.i. Fei-Fei, in: ImageNet: A Large-Scale Hierarchical Image Database, in, 2009, pp. 248–255, https://doi.org/10.1109/CVPR.2009.5206848.

[13] Dick, C.T., Barkan, C.P., Chapman, E., Stehly, M.P., 2002. Predicting the occurrence of broken rails: a quantitative approach, in: Proceedings of the 2002 Annual Conference, American Railway Engineering and Maintenance of Way Association (AREMA), Washington DC.

[14] C.T. Dick, C.P. Barkan, E.R. Chapman, M.P. Stehly, Multivariate statistical model for predicting occurrence and location of broken rails, Transp. Res. Rec. 1825 (2003) 48–55.

[15] C. Fan, J. Wang, W. Huang, X. Yang, G. Pei, T. Li, Z. Lv, Light-weight residual convolution-based capsule network for EEG emotion recognition, Adv. Eng. Inf. 61 (2024) 102522, https://doi.org/10.1016/j.aei.2024.102522.

[16] FRA, D. of T. and the F.R.A., 1998. Track Safety Standards, Final Rule, 49 CFR Part 213.

[17] T. Gao, Q. Wang, K. Yang, C. Yang, P. Wang, Q. He, Estimation of rail renewal period in small radius Curves: a data and mechanics integrated approach, Measurement 185 (2021) 110038, https://doi.org/10.1016/j.measurement.2021.110038.

[18] F. Ghofrani, N.K. Chava, Q. He, Forecasting risk of service failures between successive rail inspections: a data-driven approach, Journal of Big Data Analytics in Transportation 2 (2020) 17–31.

[19] F. Ghofrani, H. Sun, Q. He, Analyzing risk of service failures in heavy haul rail lines: a hybrid approach for imbalanced data, Risk Anal. (2021).

[20] K. He, X. Zhang, S. Ren, J. Sun, in: Deep Residual Learning for Image Recognition, IEEE, Las Vegas, NV, USA, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.

[21] S. Ho Ro, Y. Li, J. Gong, A machine learning approach for Post-Disaster data curation, Adv. Eng. Inf. 60 (2024) 102427, https://doi.org/10.1016/j.aei.2024.102427.

[22] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.

[23] A. Jamshidi, S. Faghih-Roohi, S. Hajizadeh, A. Núñez, R. Babuska, R. Dollevoet, Z. Li, B. De Schutter, A big data analysis approach for rail failure risk assessment, Risk Anal. 37 (2017) 1495–1507.

[24] A. Jamshidi, S.F. Roohi, A. Núñez, R. Babuska, B. De Schutter, R. Dollevoet, Z. Li, Probabilistic defect-based risk assessment approach for rail failures in railway infrastructure, IFAC-PapersOnLine 49 (2016) 73–77.

[25] Y. Li, W. Ji, S.M. AbouRizk, in: Automated Abstraction of Operation Processes from Unstructured Text for Simulation Modeling, IEEE, Orlando, FL, USA, 2020, pp. 2517–2525, https://doi.org/10.1109/WSC48552.2020.9383953.

[26] J. Liu, Z. Yang, H. Qi, T. Jiao, D. Li, Z. Wu, N. Zheng, S. Xu, Deep learning-assisted automatic quality assessment of concrete surfaces with cracks and bugholes, Adv. Eng. Inf. 62 (2024) 102577, https://doi.org/10.1016/j.aei.2024.102577.

[27] Liu, X., Dick, C.T., Lovett, A., Saat, M.R., Barkan, C.P., 2013. Seasonal effect on the optimization of rail defect inspection frequency, in: Rail Transportation Division Conference. American Society of Mechanical Engineers, p. V001T01A008.

[28] X. Liu, M.R. Saat, C.P.L. Barkan, Analysis of causes of major train derailment and their effect on accident rates, Transp. Res. Rec. 2289 (2012) 154–163, https://doi.org/10.3141/2289-20.

[29] S. Ma, L. Gao, X. Liu, J. Lin, Deep learning for track quality evaluation of high-speed railway based on vehicle-body vibration prediction, IEEE Access 7 (2019) 185099–185107, https://doi.org/10.1109/ACCESS.2019.2960537.

[30] T. Michálek, M. Kohout, On the problems of lateral force effects of railway vehicles in S-curves, Veh. Syst. Dyn. 60 (2022) 2739–2757, https://doi.org/10.1080/00423114.2021.1917631.

[31] R. Mohammadi, Q. He, F. Ghofrani, A. Pathak, A. Aref, Exploring the impact of foot-by-foot track geometry on the occurrence of rail defects, Transp. Res. Part C Emerging Technol. 102 (2019) 153–172, https://doi.org/10.1016/j.trc.2019.03.004.

[32] Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J., 2022. A time series is worth 64 words: Long-term forecasting with Transformers. https://doi.org/10.48550/ARXIV.2211.14730.

[33] D. Schafer, C. Barkan, A hybrid logistic regression/neural network model for the prediction of broken rails. In: Proceedings of the 8th World Congress on Railway Research, 2008.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Curran Associates Inc, 2017.

[35] S. Veskoović, J. Tepić, M. Ivić, G. Stojić, S. Milinković, Model for predicting the frequency of broken rails, Metalurgija 51 (2012) 221–224.

[36] X. Wang, Y. Bai, X. Liu, Prediction of railroad track geometry change using a hybrid CNN-LSTM spatial-temporal model, Adv. Eng. Inf. 58 (2023) 102235, https://doi.org/10.1016/j.aei.2023.102235.
[37] X. Wang, X. Liu, Z. Bian, A machine learning based methodology for broken rail prediction on freight railroads: a case study in the United States, Constr. Build. Mater. 346 (2022) 128353, https://doi.org/10.1016/j.conbuildmat.2022.128353.
[38] X. Wang, X. Liu, T.L. Euston, Relationship between track geometry defect occurrence and substructure condition: a case study on one passenger railroad in the United States, Constr. Build. Mater. 365 (2023) 130066, https://doi.org/10.1016/j.conbuildmat.2022.130066.
[39] L. Wen, X. Li, L. Gao, A transfer convolutional neural network for fault diagnosis based on ResNet-50, Neural Comput & Applic 32 (2020) 6111–6124, https://doi.org/10.1007/s00521-019-04097-w.
[40] Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., Sun, L., 2022. Transformers in time series: A survey. https://doi.org/10.48550/ARXIV.2202.07125.
[41] C. Yu, F. Wang, Z. Shao, T. Sun, L. Wu, Y. Xu, in: Dsformer: A Double Sampling Transformer for Multivariate Time Series Long-Term Prediction, in, Birmingham United Kingdom, 2023, pp. 3062–3072, https://doi.org/10.1145/3583780.3614851.
[42] C. Zeng, J. Huang, H. Wang, J. Xie, S. Huang, Rail break prediction and cause analysis using imbalanced in-service train data, IEEE Trans. Instrum. Meas. 71 (2022) 1–14, https://doi.org/10.1109/TIM.2022.3214494.
[43] Z. Zhang, X. Liu, H. Hu, Statistical analysis of seasonal effect on freight train derailments, J. Transp. Eng., Part a: Systems 147 (2021) 04021073, https://doi.org/10.1061/JTEPBS.0000583.
[44] S. Zhong, W. Lyu, D. Zhang, Y. Yang, in: Bikecap: Deep Spatial-Temporal Capsule Network for Multi-Step Bike Demand Prediction, IEEE, Bologna, Italy, 2022, pp. 831–841, https://doi.org/10.1109/ICDCS54860.2022.00085.
[45] S. Ziyabari, Z. Zhao, L. Du, S.K. Biswas, Multi-branch Resnet-Transformer for short-term spatio-temporal solar irradiance forecasting, IEEE Trans. on Ind. Applicat. 59 (2023) 5293–5303, https://doi.org/10.1109/TIA.2023.3285202.