Rail break prediction and cause analysis using imbalanced in-service train data

Zeng, Cheng; Huang, Jinsong; Wang, Hongrui; Xie, Jiawei; Huang, Shan

DOI
10.1109/TIM.2022.3214494

Publication date
2022

Document Version
Final published version

Published in
IEEE Transactions on Instrumentation and Measurement

Rail Break Prediction and Cause Analysis Using Imbalanced In-Service Train Data

Cheng Zeng, Jinsong Huang, Hongrui Wang, *Member, IEEE*, Jiawei Xie, and Shan Huang

***Abstract*— Timely detection and identification of rail breaks are crucial for safety and reliability of railway networks. This article proposes a new deep learning-based approach using the daily monitoring data from in-service trains. A time-series generative adversarial network (TimeGAN) is employed to mitigate the problem of data imbalance and preserve the temporal dynamics for generating synthetic rail breaks. A feature-level attention-based bidirectional recurrent neural network (AM-BRNN) is proposed to enhance feature extraction and capture two-direction dependencies in sequential data for accurate prediction. The proposed approach is implemented on a three-year dataset collected from a section of railroads (up to 350 km) in Australia. A real-life validation is carried out to evaluate the prediction performance of the proposed model, where historical data are used to train the model and future "unseen" rail breaks along the whole track section are used for testing. The results show that the model can successfully predict nine out of 11 rail breaks three months ahead of time with a false prediction of nonbreak of 8.2%. Predicting rail breaks three months ahead of time will provide railroads enough time for maintenance planning. Given the prediction results, a Shapley additive explanations (SHAP) method is employed to perform a cause analysis for individual rail break. The results of cause analysis can assist railroads to plan appropriate maintenance to prevent rail breaks.**

***Index Terms*— Cause analysis, deep learning-based approach, in-service train data, rail break prediction, real-life validation.**

I. Introduction

RAIL break is a major threat to the reliability of the railroads, leading to increased risk of accidents and cost of maintenance. Currently, visual inspections and track circuit systems are usually used to detect rail breaks. However, visual inspections are slow and laborious, and the results depend on human operators. In addition, due to long inspection interval, rail breaks may occur with the growing of undetected defects after an inspection. The drawback of track circuit system is that the rail break has already occurred, resulting in a post-fix rather than preventing the rail breaks in the first place. Thus, developing a predictive model to identify locations at high risk for rail break's occurrence has economic and safety benefits, enabling more effective maintenance to prevent rail breaks, service interruptions, and potential derailments.

Previous research on predictive models for rail breaks can be divided into two categories: 1) probabilistic models for analyzing the risk of rail breaks and 2) machine learning-based models for predicting the location where rail breaks are likely to occur.

Chattopadhyay and Kumar [1] applied the Weibull distribution to model the relationship between the probability of rail breaks and annual tonnage, where the break probability function monotonically increased with annual tonnage. Based on the fuzzy logic process, Vesković et al. [2] developed a fuzzy model to predict the frequency of broken rails in a large region. Bai et al. [3] employed the theory of Markov stochastic processes to estimate the rail break time in a long period, such as five years. A regression model based on survival analysis was developed by Ghofrani et al. [4] to predict the risk of rail breaks between two successive rail inspections. Later, Ghofrani et al. [5] developed an improved version of the prediction model to predict the arrival rate of rail breaks. Those probabilistic models perform well in analyzing the risk of rail breaks over a large region or long-term period. However, they are incapable to predict the location where rail breaks are prone to occur, which makes predictive maintenance difficult.

Dick et al. [6] developed a logistic regression model to quantify the probability of rail breaks at any particular location in a two-year period. Track characteristics data, such as rail age, annual tonnage, mainline turnouts, and degree of curve, were selected for inclusion in the model. Based on the same dataset, Schafer and Barkan [7] proposed a hybrid method by using logistic regression to select important parameters and an artificial neural network (ANN) to predict rail breaks. A recent study proposed by Zhang et al. [8] used multisource of data, including track characteristics and traffic information to predict rail breaks. Ghofrani et al. [9] applied a gradient boosting machine to analyze the risk of rail breaks by considering geometry data and monthly average temperature. While the track characteristics data are potentially relevant to the appearance of rail breaks, such data remain constant over a large region, which may not be suitable for identifying rail breaks at a specific location. The foot-by-foot track geometry data used by Ghofrani et al. [9], however, is difficult to capture potential rail breaks in an early stage due to the long data-collection interval.

By equipping with various types of sensors on in-service trains, the entire rail networks can be monitored continuously in a timely and cost-efficient manner [10], [11], [12], [13]. It has been shown that monitoring data collected

Manuscript received 4 July 2022; revised 14 September 2022; accepted 2 October 2022. Date of publication 17 October 2022; date of current version 31 October 2022. This work was supported by the Australian Government through the Australian Research Council's (ARC) Linkage Projects Funding Scheme under Project LP160101254. The Associate Editor coordinating the review process was Dr. Rajarshi Gupta. *(Corresponding author: Jinsong Huang.)*

Cheng Zeng, Jinsong Huang, Jiawei Xie, and Shan Huang are with the Discipline of Civil, Surveying and Environmental Engineering, The University of Newcastle, Callaghan, NSW 2308, Australia (e-mail: jinsong.huang@newcastle.edu.au).

Hongrui Wang is with the Section of Railway Engineering, Delft University of Technology, 2628 CN Delft, The Netherlands.

Digital Object Identifier 10.1109/TIM.2022.3214494

from in-service train can provide useful insights into the vehicle–track interaction and can be associated with the quality of the track condition [10], [14], [15], [16], [17], [18]. Also, in-service train data have been used to estimate track stiffness [11] and predict track substructure defects such as mud pumping [19]. Such huge amount of monitoring data has been collected over the past several years by railroad operators, but, to the best of our knowledge, the potential of in-service train data has not been exploited for the prediction of rail breaks. There are several reasons for this lack of exploration, and they all present major challenges when dealing with in-service train data for rail break prediction.

Data imbalance is one of the main challenges in rail break prediction because the number of available rail break events is very limited, and thus, rail break events are heavily outnumbered by nonbreak events. Undersampling of nonbreak events was commonly used in previous studies [6], [7] for balancing the dataset. In this case, most nonbreak events were not used and their information got lost. Ghofrani et al. [9] used the synthetic minority oversampling technique (SMOTE) to handle the data imbalance by generating more rail break events. However, as the strategy for SMOTE is based on data interpolation, it may insert lots of noise and is less effective in creating synthetic data subject to the underlying distribution of the real data [20]. Recently, Goodfellow et al. [21] proposed a general, flexible, and powerful framework for estimating generative models called generative adversarial network (GAN). By capturing characteristics from the original data, GAN is capable of generating new data sharing the same distributions with the original one. Yang et al. [22] used GAN to generate scarce fault data of harmonic drive for fault diagnosis. Zhang et al. [23] applied GAN to generate the signals in machine faulty states for machinery fault diagnosis. Wang et al. [24] utilized a deep convolutional GAN (DCGAN) to generate fault samples to balance the dataset for fault diagnosis of rotating machines. Zhong et al. [25] and Lyu et al. [26] constructed DCGANs for anomaly detection of catenary support components in railways. Liu et al. [27] proposed a long short-term memory-based GAN (LSTM-GAN) for fault diagnosis in machine health monitoring. As one can see, no previous studies have applied GAN to in-service train data for rail break prediction. The in-service train data associated with rail breaks are in the format of time sequences. The amplitude of in-service train data will change over time. A generative model that considers temporal dynamics in data is necessary when generating the rail break sample.

Another challenge is that the in-service train data are time sequence, and learning the temporal dependencies that are characteristics of the rail breaks is difficult for conventional machine learning methods [6], [7], [8], [9]. In recent years, deep learning technology has achieved excellent performance in image recognition [28], [29], language translation [30], and signal analysis [31], [32], [33] as it has a strong ability for automatic feature extraction from different types of data such as images, text, and time sequences. Zhong et al. [34] proposed an improved deep convolutional neural networks (CNNs) to process image data for defect detection of catenary split pins in high-speed railways. Chen et al. [35] applied a CNN to learn and classify sensor data for fault diagnosis in railway switch systems. However, the CNN model is originally designed based on image analysis and thus may not be effective in capturing temporal dynamics in time sequences data. Recurrent neural networks (RNNs) proposed by Graves and Schmidhuber [36] is a commonly used deep learning model for sequential data; de Bruin et al. [37] used LSTM to detect the circuit faults from the collected signals in track circuits. However, one-way LSTM was used, which may not perform well for in-service train data. A recent state-of-the-art attention-based framework called Transformer [38] is proposed to extract task-related feature information from massive amounts of data and automatically give correct predictions due to its powerful automatic feature learning capabilities. Wang et al. [39] applied a transformer-based framework for high-speed train wheel wear prediction from collected vibration signals.

Most deep learning-based predictive models only provide a binary result, i.e., break or nonbreak. In real-world applications, however, engineers prefer to know which causes lead to a break, i.e., which parameters are most important when the model makes a certain prediction, so that they can plan maintenance accordingly to prevent rail breaks. However, the vast majority of deep learning-based models are treated as black box and lack of explainability traits. Recent studies are being developed with a focus on explainable artificial intelligence [40], [41]. Those explainable methods can provide information to understand how the model performs prediction and thus identify the causes of rail breaks [42], [43].

This article proposes a new deep learning-based approach for rail break prediction and cause analysis using monitoring data from in-service trains. To mitigate the problem of data imbalance and preserve the temporal dynamics in data, a time-series GAN (TimeGAN) is employed to learn the distribution of the real rail break samples and generate synthetic rail break samples. To make an accurate prediction, a feature-level attention-based bidirectional recurrent neural network (AM-BRNN) is proposed to enhance feature extraction and capture two-direction dependencies from sequential data. The proposed approach is implemented on a three-year dataset collected from a section of railroads (up to 350 km) in Australia to show its applicability. Most studies evaluate and compare the prediction models using cross-validation approaches, where training and testing data are randomly selected from the same dataset that consists of only historical rail breaks. In addition to cross validation, a real-life validation is carried out to evaluate the prediction performance of the proposed model, where historical data are used to train the model and future "unseen" rail breaks along the whole track section are used for testing. The results show that the model can successfully predict nine out of 11 rail breaks three months ahead of time with a false prediction of nonbreak of 8.2%. Predicting rail breaks three months ahead of time will provide railroads enough time for maintenance planning. Given the prediction results, a Shapley additive explanations (SHAP) method is employed to perform the cause analysis for individual rail break. The results of cause analysis can be used for planning maintenance to prevent rail breaks.

TABLE I
DESCRIPTION OF PARAMETERS

No.	Parameters	Description
1	Va	Maximum vertical acceleration at front and rear bogie
2	Va_F	Maximum vertical acceleration at front bogie
3	Va_R	Maximum vertical acceleration at rear bogie
4	Vastd_F(L)	The standard deviation of vertical acceleration at front bogie (left side frame)
5	Vastd_F(R)	The standard deviation of vertical acceleration at front bogie (right side frame)
6	Vastd_R(L)	The standard deviation of vertical acceleration at rear bogie (left side frame)
7	Vastd_R(R)	The standard deviation of vertical acceleration at rear bogie (right side frame)
8	SD	Maximum suspension displacement at front and rear bogie
9	SD_F	Maximum suspension displacement at front bogie
10	SD_R	Maximum suspension displacement at rear bogie
11	SDdif_F	The difference of suspension displacement at front bogie
12	SDdif_R	The difference of suspension displacement at rear bogie
13	SDavg_F	The average of suspension displacement at front bogie
14	SDavg_R	The average of suspension displacement at rear bogie
15	Twist1	Difference between two cross-level measurements at a certain distance apart (2 m chords)
16	Twist2	Difference between two cross-level measurements at a certain distance apart (14 m chords)
17	CRV	The degree to which a curve deviates from a straight line
18	Pro(L)	Vertical profile (left rail)
19	Pro(R)	Vertical profile (right rail)
20	Align	GPS correction
21	BCR	Brake cylinder pressure
22	Force	Train handling dynamics
23	Railage	Rail age
24	Tonnage	Annual tonnage
25	Ton*age	The product of annual tonnage and rail age
26	Speed	Speed of in-service train
27	Temp_min	Daily minimum temperature

The main contributions of this article are summarized as follows.

1) For the first time, this article designs an AM-BRNN for predicting rail breaks using in-service train data. The proposed AM-BRNN achieves better predictive performance in terms of efficiency and accuracy than current models. Unlike previous studies, this article validates the proposed approach against future "unseen" rail breaks along 350-km railway tracks. The proposed approach is proven to be able to predict rail breaks ahead of time so that maintenance can be performed to reduce rail breaks.
2) To overcome the data imbalance problem, this article customizes a TimeGAN for generating synthetic rail break samples. The customized TimeGAN model is proven to be able to generate better quality rail break samples than current models because the temporal dynamics in in-service train data are preserved.

II. DATA DESCRIPTION AND PREPARATION

A. Data Description

Dataset available for the current study consists of in-service train data; track characteristic data, including rail age, annual tonnage, and train speed; and temperature data. The dataset is collected from a section of railroads (up to 350 km) in Australia for three years from 2018 to 2021.

1) In-Service Train Data: The in-service trains collect sampling data along the track with daily service. The sampling data are associated with dynamic train responses (e.g., suspension displacement (SD) and vertical acceleration), track geometry data (e.g., track twist and track curvature), and train driving parameters (e.g., brake cylinder pressure and in-train forces). The collected sampling data are preprocessed into 22 parameters at a 1-m interval with the corresponding locations. The details of the 22 parameters (corresponding to parameter Nos. 1–22) are shown in Table I. A total of 27 parameters will be used for rail break prediction, as can be seen from Table I. Parameters Nos. 23–27 are explained in Sections II-A2 and II-A3. Fig. 1 shows the changes of historical in-service train data over time and location, where Fig. 1(a) and (c) shows SD and Fig. 1(b) and (d) shows the vertical acceleration (Va). In Fig. 1, the horizontal axis is location in kilometers and the vertical axis is date. Color represents the magnitude of dynamic responses, where red means high values, while green means low values. The black crosses in Fig. 1 represent actual rail break events that had occurred at certain locations and on certain dates, where Fig. 1(a) and (b) indicates rail break 1 and Fig. 1(c) and (d) indicates rail break 2. The actual rail break records were provided by the railway track corporation. It can be seen from Fig. 1(a) and (b) that SD increased before rail break 1 occurred, but Va remained normal, whereas it can be seen from Fig. 1(c) and (d) that Va increased, but SD remained normal before rail break 2 occurred. This may be because rail breaks are usually caused by different factors, and thus, high responses are reflected in different parameters.

2) Track Characteristics Data: Rail age is closely related to the risk of rail break. However, due to the effect of different maintenance and replacement (M&R) activities on the rail conditions, the rail age should be adjusted accordingly. In this article, it is assumed that all rails were put into use on 01/01/2000 as the actual starting time was not recorded. If M&R activities, including rail joint replacement, rail defect removal, renewal, or rerailing, were performed, the date when the rail is put into use is reset as the recent M&R activities

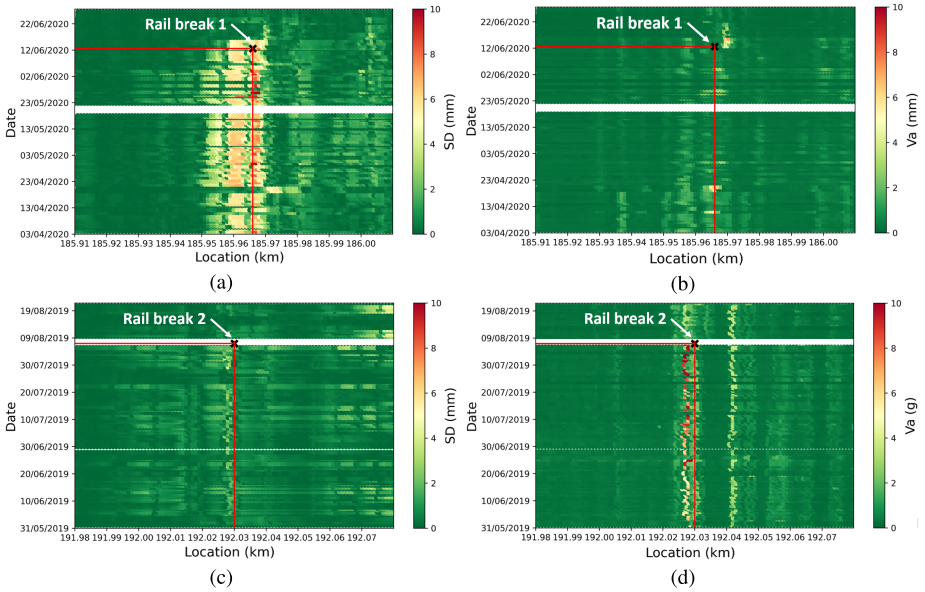


Fig. 1. Changes of SD and vertical acceleration (Va) over time and location for rail breaks 1 and 2, respectively. (a) SD for rail break 1. (b) Vertical acceleration for rail break 1. (c) SD for rail break 2. (d) Vertical acceleration for rail break 2.

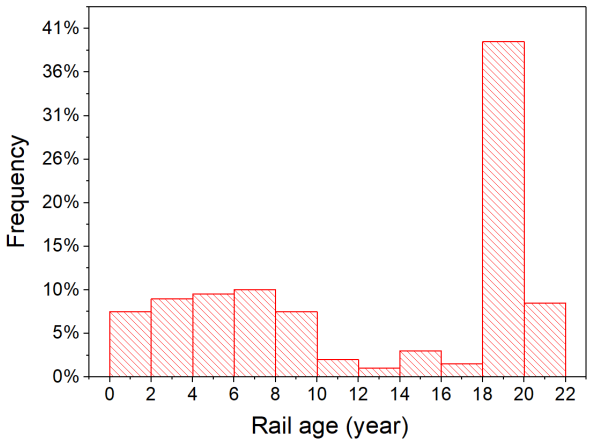


Fig. 2. Distribution of rail breaks with respect to rail age.

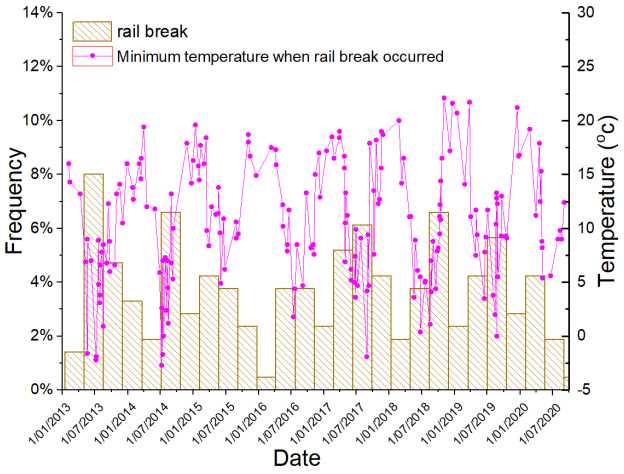


Fig. 3. Minimum temperatures on the day of rail breaks happened and the distribution of historical rail break events in every three months.

date because those M&R activities will restore the track to almost brand-new condition. Fig. 2 shows the distribution of historical rail breaks with respect to rail age. It can be seen from Fig. 2 that rail break can happen anytime. Almost 41% rail breaks happened between 18 and 20 years of rail age.

Since the studied railway network is composed of heavy haul lines, the effect of tonnage on rail is particularly important and should be considered. Annual tonnage, which is the total weight of goods and trains that passes each track section, is collected to provide tonnage information for prediction. In addition, the product of annual tonnage and rail age is also considered as it was found significant in the previous studies [6], [7].

The magnitude of the in-service train data is influenced by train speed, as the sensors are installed on the axle boxes of the in-service train's bogies. In normal running, the higher the speed, the higher the dynamic responses such as vertical acceleration [44]. To consider this effect, the train speed is collected as one of the parameters.

3) Temperature Data: It has been shown that the temperature has significant effects on the health of rails due to thermal expansion and contraction [5]. In extreme cold, rail suffers from tensile stress, which may increase the probability of rail break. Thus, daily minimum temperature data are collected. The scatterplot in Fig. 3 denotes the minimum temperature on the day of rail break. The histograms in Fig. 3 show the distribution of historical rail breaks in every three months. In Fig. 3, the horizontal axis is date, the right vertical axis is the magnitude of temperature, and the left vertical axis is frequency. As shown in Fig. 3, the rail breaks happened more frequently in cold seasons than in warm seasons.

B. Data Preparation

To locate the potential rail breaks and facilitate the effective M&R, the railway track needs to be divided into segments. If a segment is too short, a large number of segments will

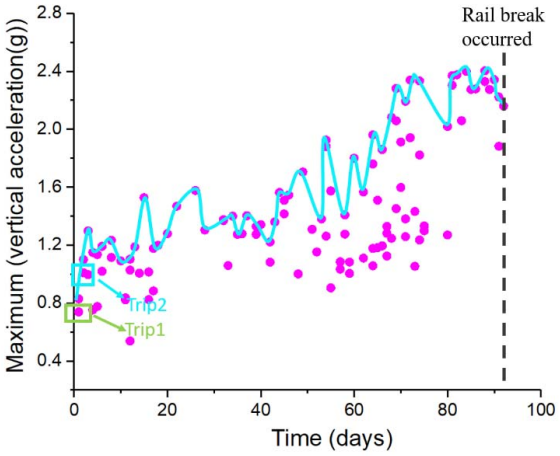


Fig. 4. Illustration of the changes of vertical accelerations over a certain section of track.

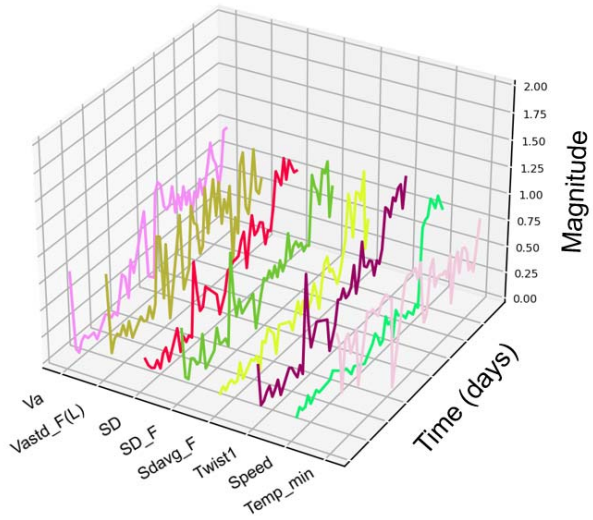


Fig. 5. Changes of several parameters over time for a given segment after standardization.

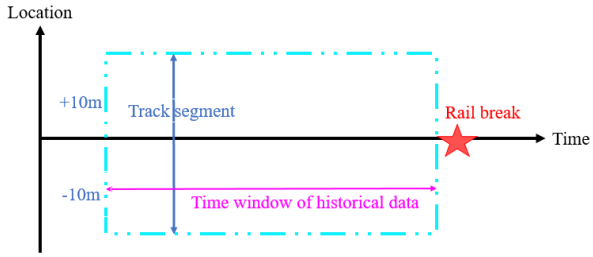


Fig. 6. Illustration of a break sample.

increase the total computational time. If a segment is too long, further inspection is required to locate potential rail break. To consider both computational efficiency and practical constraints, the length of a segment is set to be 20 m. As suggested in [45], the changes of parameters over time should be considered when predicting rail conditions. Data used in this study can be processed as time sequences to preserve the changes over time because most data is collected daily. As in-service trains collect data almost every day, the time interval is set to be one day. Fig. 4 presents an illustration of the vertical accelerations changing over time in a certain segment. Each point that is plotted in Fig. 4 represents the maximum vertical acceleration of a single trip. For some days with multiple trips, the maximum vertical acceleration varies from train to train and from trip to trip. Therefore, the maximum value of maximum acceleration in each time interval (one day) is adopted, as denoted by the curve in Fig. 4.

All in-service train parameters and train speed are processed as time sequences as aforementioned. Daily temperature data and rail age are recorded based on segment and hence are naturally time sequences. As the annual tonnage of a segment does not vary with time, there is no need for processing.

The aforementioned different types of parameters have different magnitudes. To reduce the impact of differences in magnitude, the max–min standardization method is used to transform the data to the range from 0 to 1. Fig. 5 shows the processed time sequences of several parameters for a given segment after standardization.

The prediction model is trained in the supervised mode and hence requires labeled dataset. According to the historical records, there were 346 rail breaks that occurred in the track section in the past three years. The data, which are within 10 m before and after the break location and recorded before the corresponding break date, are labeled as a "break" sample, as shown in Fig. 6. A time window is introduced to determine how many days of data should be used for model development. If the time window is too long, the learned features will be lost through the long process and the computational time will increase. If the time window is too short, the information used in the prediction will be insufficient. According to a previous study [19], using 56 days as time window can obtain the best performance in mud pumping prediction on railway track. Thus, the time window is also set to be 56 days in this study. The data that are outside 10 m of the break location are labeled as "nonbreak." The time window of nonbreak is also 56 days that are randomly selected and consecutive. The number of samples in the rail break class and nonbreak class is 346 and 5249, respectively.

III. Methodology

The proposed approach mainly includes imbalanced data handling, rail break prediction, and cause analysis. Fig. 7 shows the workflow of the proposed approach. First, TimeGAN is used to generate synthetic rail break samples to balance the training dataset. Then, the balanced training dataset is used to develop the proposed AM-BRNN model for rail break prediction. Given the prediction results, SHAP is employed to identify the input parameters that are most likely to be the causes of rail breaks.

A. Imbalanced Data Handling

The dataset is observed to be heavily imbalanced, in which the proportion of samples for the minority data class that indicates rail break is about 0.06. Machine learning algorithms that do not consider data imbalance perform poorly as imbalanced datasets induce a bias in favor of the majority class.

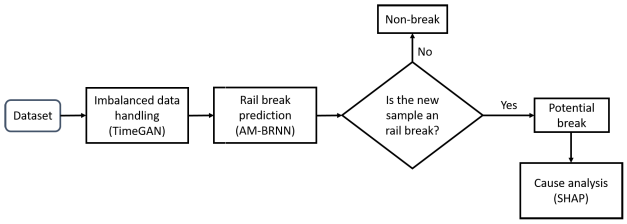


Fig. 7. Workflow of the proposed approach.

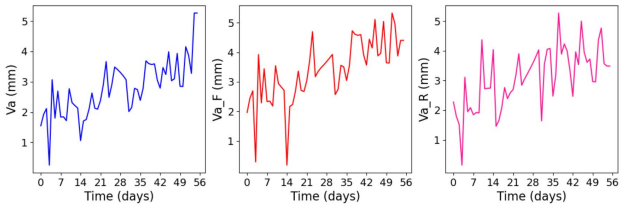


Fig. 8. Several parameters changing over time for a given rail break instance.

Hence, it is necessary to balance the dataset when developing rail break prediction models. Given the limited number of available rail break events, generating rail break sample is a natural choice. The GAN model is adopted in this study because it is capable of generating new samples sharing the same distribution with the original ones. A real rail break sample is the time sequence that changes of 27 parameters over 56 days. The amplitude of parameters will change over time. Fig. 8 shows several parameters for a given rail break. It can be seen from Fig. 8 that the amplitude of vertical acceleration increases gradually over time. Thus, the temporal dynamics in data should be considered when generating the rail break sample. However, existing GANs and their variants do not adequately consider the temporal dynamics in data when they are used for generating time sequences. To address this issue, a TimeGAN [46] is applied to generate synthetic rail break samples. In TimeGAN, a stepwise supervised loss is introduced to capture the temporal dynamics in data by using the real-time sequences as supervision. By doing this, TimeGAN can generate rail break data that preserve temporal dynamics. Another key aspect of the dataset is that the 27 parameters are usually not independent but correlated, especially the parameters collected from the same type of onboard sensors, i.e., vertical acceleration at the front bogie and the rear bogie. This can also be found in Fig. 8 that the variation of vertical acceleration at the front bogie (Va_F) and the rear bogie (Va_R) is very similar. Those similar parameters may provide redundant information. Several studies have explored the benefit of combining autoencoders (AEs) with adversarial training, such as improving generative capability [47]. Thus, an AE is used to offer a reversible mapping between input parameters and latent representations, thereby reducing the high dimensionality of the adversarial learning space.

1) Autoencoder: AE consists of encoder and decoder, where the encoder E is used to encode the input into the latent representation and the decoder R is used to decode the latent representation into the input. The real rail break data are fed into the AE for reconstruction. The encoder and decoder are all implemented via RNNs, where the used RNN consists of LSTM layer and fully connected layer. The reconstruction can be achieved by minimizing the loss $\mathcal{L}_R$

$$\mathcal{L}_R = \mathbb{E}_{x_{1:T}\sim p}\left[\sum_{t=1}^{T} \|x_t - R\left(E(x_t)\right)\|_2\right] \tag{1}$$

where T is the length of time steps and x is the original input time sequence.

2) Adversarial Component: It has two subnetworks: a generator G and a discriminator D. The generator takes noise as input and aims to generate a synthetic sample as close as possible to the real rail break sample from the latent representation. Then, the discriminator distinguishes whether the generated sample comes from the real dataset or from the generator. In the original TimeGAN framework [46], the generator was implemented via a standard RNN. BRNNs have shown better performance in learning temporal dynamics than the standard ones [48]. Thus, BRNN is used as the network structure of the generator in this study. Generator and discriminator are all implemented via BRNNs, where the used BRNN consists of bidirectional LSTM layer. Generator and discriminator compete in a two-player min–max game with loss $\mathcal{L}_U$

$$\min_{\theta_g}\max_{\theta_d}\mathcal{L}_U = \mathbb{E}_{x_{1:T}\sim p}\left[\sum_{t=1}^{T}\log D(x_t)\right] + \mathbb{E}_{z_{1:T}\sim \widehat{p}}\left[\sum_{t=1}^{T}\log(1 - D(G(z_t)))\right] \tag{2}$$

where θ_g denotes the learnable parameters in G, θ_d denotes the learnable parameters in D, and z is random noise with uniform distribution.

3) AR Learning Objective: A supervised AR learning objective is introduced to explicitly encourage the generator to capture the stepwise conditional distributions in the data using the real data as supervision. Thereby, the transition dynamics from real sequences can be learned. In an alternating fashion, the generator is trained in the closed-loop mode, where the generator receives sequences of real sample in latent space $l_{1:t-1}$ to generate the next latent vector l_t. Gradients can be computed on a supervised loss $\mathcal{L}_S$, which is defined as

$$\mathcal{L}_S = \mathbb{E}_{x_{1:T}\sim p}\left[\sum_{t=1}^{T} \|l_t - G\left(l_{t-1}, z_t\right)\|_2\right]. \tag{3}$$

Fig. 9 shows the architecture of TimeGAN. TimeGAN training consists of three main phases: AE training to provide latent space, supervised training to learn conditionals $\widehat{p}(x_t|x_{1:t-1})$, and joint training to learn distribution $\widehat{p}(x_{1:t})$. First, purely as a reversible mapping between the original data and latent space, E and R are trained together to enable accurate reconstruction of the original data from its latent representation. Then, G is trained exclusively using the supervised loss to capture the temporal dynamics in the data. Finally, the joint training phase consists of training G, D, E, and R using their respective loss functions. Since the parameters may be correlated as they are collected from sensors mounted on the same in-service train, the TimeGAN is trained once using all parameters. The input

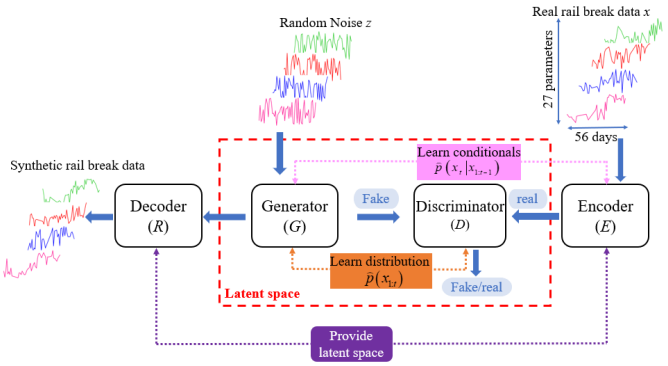


Fig. 9. Architecture of the TimeGAN.

data are the time sequences that the changes of 27 parameters are over 56 days, with the shape of [56 × 27].

B. Rail Break Prediction

Since the in-service train data are in the format of time sequences, learning the temporal dependencies is difficult for the conventional machine learning techniques. To capture the temporal dependencies, BRNN is a highly preferred choice [48]. The recurrent connections in BRNN consider both forward and backward information and hence are able to capture two-direction dependencies. In the recurrent layers, LSTM is used, which is well suited for capturing long-range dependencies on multiple time scales [49].

Besides, the in-service train data are collected from multiple sensors and hence are high-dimensional with various parameters. Accurate prediction relies on effective feature extraction. Recent studies have introduced attention mechanisms into the neural network framework [30], [38] to selectively focus on important information by setting different weights on input or feature dimension. In computer vision, channelwise attention is usually applied to a high-level feature map to exploit the interchannel relationship of features [50], [51], [52]. A similar technique can be found in speech recognition where time series is involved. Cheng et al. [53] proposed a DNN-based model for speech enhancement using attention mechanisms on the feature dimension. They suggested that the attention model can make full use of the key information in features and improve accuracy. Wang et al. [54] proposed a MaskNet model to estimate the click-through rate (CTR), in which feature-level attention was introduced to dynamically highlight the informative elements in the hidden layer. Inspired by the previous studies, this article introduces feature-level attention into the network to assign the weight for the features. In this way, the model can strengthen the utilization of feature information and further improve the prediction performance.

1) Bidirectional LSTM: Bidirectional LSTM layers utilize two parallel channels (forward and backward) simultaneously and concatenate the hidden states of the two LSTMs as the representation of each time step t. The hidden state h_t and memory cell c_t in LSTM are the function (represented as g^{LSTM}) of their previous status h_{t-1}, c_{t-1} and input vector W_t. The hidden state of each location h_t in bidirectional LSTM considers the forward and backward information, and

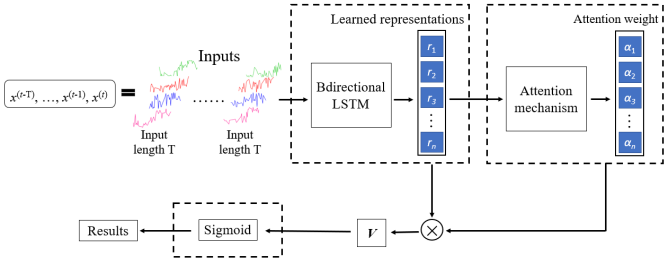


Fig. 10. Basic framework of the AM-BRNN model.

its form is given as follows:

$$\overrightarrow{c}_t, \overrightarrow{h}_t = g^{\text{LSTM}}\left(\overrightarrow{c}_{t-1}, \overrightarrow{h}_{t-1}, W_t\right) \tag{4}$$

$$\overleftarrow{c}_t, \overleftarrow{h}_t = g^{\text{LSTM}}\left(\overleftarrow{c}_{t-1}, \overleftarrow{h}_{t-1}, W_t\right). \tag{5}$$

The representation of the entire sequences is $[\overrightarrow{h}_T, \overleftarrow{h}_T]$, where T is the length of time sequences. At each time step t, the representation is $h_t = \overrightarrow{h}_t \oplus \overleftarrow{h}_t$, which is a concatenation of the hidden states of the forward LSTM and backward LSTM. In this way, the forward and backward information can be considered simultaneously.

2) Attention Mechanism: The attention module takes as inputs the features r from bidirectional LSTM layers $r = f_{\text{ext}}(x) = [r_1, r_2, \ldots, r_{\text{Nunit}}]$, where Nunit is dimensions of the features. Next, the attention mechanism generates a positive weight α_i for r_i, which can be considered as the importance of the corresponding features. α_i is calculated using an attention model, which takes r_i as input, and a softmax function to get a normalized importance weight. In this study, a one-layer neural network is used as an attention model

$$u_i = f(W_{\text{att}} r_i + b_{\text{att}}) \tag{6}$$

$$\alpha_i = \frac{e^{u_i}}{\sum_{k=1}^{\text{Nunit}} e^{u_k}} \tag{7}$$

where W_{att} denotes the weight matrix, b_{att} denotes the bias term, and u_i represents a hidden representation of r_i through a one-layer neural network. When the attention for r_i is generated, the enhanced feature vector v can be obtained as

$$v = \sum_{i=1}^{\text{Nunit}} \alpha_i r_i. \tag{8}$$

The basic framework of AM-BRNN model is shown in Fig. 10. After enhanced features are generated, a sigmoid function is adopted for the final rail break prediction, which converts the outputs of the network to the probabilities of each class.

C. Cause Analysis

A full perspective of rail break prediction should not only give an estimation of rail break but also provide knowledge about its cause, allowing the domain experts to perform predictive maintenance accordingly. In this article, the most important input parameters that lead to a predicted rail break are referred to as causes. Note that we do not imply these to be the root causes but rather that they are the causes due to which the algorithm flagged a rail break. Different explainable

methods can be used to estimate the importance of each parameter in making a certain prediction (rail break or nonbreak). One of the most used explainable methods is SHAP [40], which is a model-agnostic representation of parameter importance. In the SHAP method, the parameter importance on the prediction model is represented using SHAP values. The SHAP method requires the sum of the SHAP values to match the output of prediction model (i.e., AM-BRNN) for specific input. By calculating the SHAP values of each parameter for an individual rail break, it is possible to determine which parameters are most important for making the prediction of rail break. In this case, a cause analysis for individual rail break can be achieved. Since the exact computation of SHAP values is still challenging, various methods, including Tree SHAP, Kernel SHAP, and Deep SHAP, are used to approximate the SHAP values. In this study, Deep SHAP is used together with a deep learning prediction model. The detailed mathematical formulation of SHAP can be retrieved in [40].

IV. Results and Discussion

This section verifies the effectiveness of the proposed AM-BRNN and TimeGAN using data collected from a section of railway tracks in Australia. The generation performance of the TimeGAN is first evaluated through data quality assessment of newly generated samples. Then, the prediction performance of the AM-BRNN is evaluated through five-fold cross validation. Furthermore, the complexity analysis of the proposed model is conducted to check its potential to be deployed in standalone devices in field for real-time prediction. Finally, a real-life validation is performed by using the proposed approach. Given the prediction results, a cause analysis is performed on a certain predicted rail break using the SHAP method.

A. Data Generation and Quality Assessment

The TimeGAN is aimed to generate synthetic samples for rail break events for solving the data imbalance problem. For the validation of the method, the quality of the synthetic samples needs to be evaluated. In this article, the quality assessment on the synthetic samples is conducted from three aspects as follows.

1) Diversity: Synthetic samples should be distributed to cover the real ones. Principal component analysis (PCA) is used to flatten the temporal dimension for both the real and synthetic samples to assess the diversity of synthetic samples. The results visualize how closely the distribution of synthetic samples resembles that of the real in 2-D space, providing a qualitative assessment of diversity

2) Fidelity: Synthetic samples should be indistinguishable from the real ones. To do this, a post hoc classification model is trained to discriminate whether the sample is from the real or synthetic datasets. First, each real sample is labeled as real and each synthetic sample is labeled as not real. Then, as a standard supervised task, an LSTM classifier is trained to classify the real and not real classes. The classification error on the randomly selected test set is reported as discriminative score, which provides a quantitative assessment of fidelity.

3) Usefulness: Synthetic samples should preserve the temporal dynamics characteristics of the real ones. In particular, synthetic samples are expected to capture conditional distributions over time. Thus, an LSTM regression model is trained to predict next-step temporal vectors over each input sequence using the synthetic samples. Then, the trained model is evaluated on the real samples. Performance is measured using the mean absolute error (MAE), which is also reported as predictive score. This provides a quantitative assessment of usefulness.

Based on the 346 real samples of historical rail break events, the TimeGAN model is trained to generate 346 synthetic rail break-related samples for checking the quality of the synthetic samples. In addition, to demonstrate the superiority of the TimeGAN model, two conventional generation methods, including SMOTE [55] and dynamic time warping barycentre averaging (DBA) [56], and three GAN-based generation methods, including GAN [22], DCGAN [23], [24], [26], and LSTM-GAN [27], are applied to generate synthetic samples separately for comparison.

PCA is used to compare the similarity of synthetic samples and real rail break samples. The results are shown in Fig. 11, where circles denote synthetic samples and crosses denote real samples. Generally, a good overlap between the synthetic samples and the real samples indicates high similarity. It can be seen from Fig. 11 that synthetic samples (circles) generated by SMOTE, DBA, and GAN overlap partially the real samples, whereas the synthetic samples generated by DCGAN, LSTM-GAN, and TimeGAN almost cover all the real samples. However, compared to TimeGAN, DCGAN and LSTM-GAN generate more synthetic samples that are far away from the cloud of real samples.

To further evaluate the synthetic samples, discriminative score and prediction score are calculated to measure the fidelity and usefulness of the synthetic samples quantitatively. Discriminative score is a probability score between 0 and 1. Generally, a score close to zero implies good fidelity, which means that the synthetic samples are very similar to the real ones. Similarly, a low value of predictive score signifies good usefulness of the synthetic samples. As can be seen from Table II, all GAN-based methods generate better synthetic samples in comparison to conventional methods, including SMOTE and DBA based on discriminative scores. However, GAN and DCGAN perform slightly worse than LSTM-GAN and TimeGAN based on predictive scores. This may be because there is no RNN block in GAN and DCGAN that can preserve temporal dynamics in data. In general, TimeGAN-generated samples achieve the best (lowest) discriminative score of 0.283 and the predictive score of 0.154.

Based on the above analysis, it can be concluded from Fig. 11 and Table II that TimeGAN demonstrates consistent superiority over five other methods in generating synthetic rail break samples.

B. Prediction Model Evaluation

To evaluate the performance of the proposed AM-BRNN with TimeGAN, a fivefold cross-validation procedure is

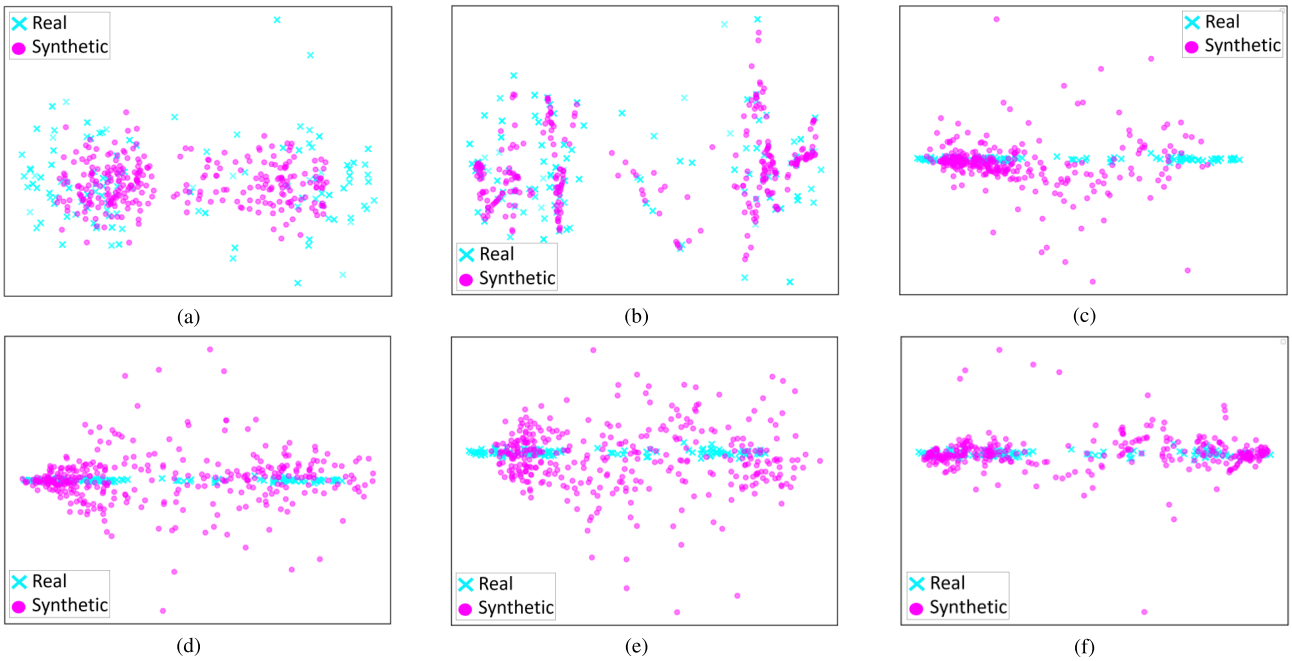


Fig. 11. PCA visualizations on results of six different generation methods. Crosses denote real samples, and circles denote synthetic samples. (a) DBA. (b) SMOTE. (c) GAN. (d) DCGAN. (e) LSTM-GAN. (f) TimeGAN.

TABLE II
DISCRIMINATIVE SCORES AND PREDICTIVE SCORES OF SIX DIFFERENT GENERATION METHODS (BOLD NUMBERS INDICATE THE BEST PERFORMANCE)

Generation method	Discriminative score (Lower the better)	Predictive score (Lower the better)
DBA	0.458	0.156
SMOTE	0.387	0.161
GAN	0.386	0.257
DCGAN	0.329	0.221
LSTM-GAN	0.341	0.182
TimeGAN	**0.283**	**0.154**

carried out. First, the dataset (346 rail break and 5249 non-break samples) is randomly divided into five folds of approximately equal size: one of the five folds is treated as a test dataset and the remaining four folds are treated as a training dataset. To avoid rail break information leakage during the data generation process, the TimeGAN is only applied to the training dataset to generate synthetic samples to get the balanced training dataset. Then, the AM-BRNN is developed based on the balanced training dataset. The prediction performance is evaluated on the originally imbalanced test dataset. This process is repeated five times and the average of the five times is recorded as the final result. Since TimeGAN has shown superior generation performance in Section IV-A, TimeGAN is used to generate synthetic samples to balance the training dataset used for the prediction model.

Given that the test dataset is imbalanced, using the standard accuracy as an evaluation metric may lead to a prediction model looking promising with high accuracy but fails to be valid in predicting rail breaks. Hence, both false positive rate (FPR) and false negative rate (FNR) are used for measuring the performance of the prediction model for break and nonbreak, respectively. For ease of understanding, false prediction of break and false prediction of nonbreak are introduced to replace the machine learning terminology FPR and FNR, respectively. Besides, balanced accuracy is introduced as a compact evaluation metric to reflect the general performance of prediction models. The balanced accuracy is especially useful when the test dataset is imbalanced.

False prediction of break is a ratio of falsely predicted break samples to the total break samples

$$\text{False prediction of break} = \frac{FN}{P} \tag{9}$$

where P indicates the total number of real break samples and FN denotes the number of break samples falsely predicted as nonbreak.

False prediction of nonbreak is a ratio of falsely predicted nonbreak samples to the total nonbreak samples

$$\text{False prediction of non-break} = \frac{FP}{N} \tag{10}$$

where N denotes the total number of real nonbreak samples and FP denotes the number of nonbreak samples falsely predicted as break.

Balanced accuracy is the average of correct predictions obtained on break and nonbreak classes

$$\text{Balanced accuracy} = \frac{1}{2} \times \left(\frac{TP}{P} + \frac{TN}{N} \right) \tag{11}$$

where TP indicates the number of break samples correctly predicted as break and TN denotes the number of nonbreak samples correctly predicted as nonbreak.

To demonstrate the superiority of the AM-BRNN, four deep learning methods that were used in fault diagnosis in railway systems, including multilayer perceptron (MLP) neural

TABLE III
STRUCTURES OF THE MLP, 1DCNN, CNN-LSTM, AND TRANSFORMER

Deep learning method	Details
MLP	FCN size 128, 1 FCN layer; FCN size 64, 1 FCN layer.
1DCNN	1-D convolutional kernel size 128, 1 unit; 1-D convolutional kernel size 64, 1 unit; FCN size 64, 1 FCN layer.
CNN-LSTM	1-D convolutional kernel size 128, 1 unit; 1-D convolutional kernel size 64, 1 unit; LSTM size 64, 1 LSTM; FCN size 64,1 FCN layer.
Transformer	1 Transformer encoder layer; FCN size 64, 1 FCN layers.

network [9], CNNs (1DCNN) [35], CNN-LSTM [31], and Transformer [39], are applied for comparison. The structures of MLP, 1DCNN, CNN-LSTM, and Transformer are presented in Table III. All prediction models are trained via mini-batch stochastic gradient descent with the Adam optimizer. The size of the minibatch is 64. The learning rate is set to be 0.001. The training process is set to be stopped early when the loss does not decrease over an average of the last ten epochs. The prediction models are smooth and differentiable so that the learnable parameters can be learned by standard backpropagation with binary cross entropy as the objective function.

The performances are compared in Table IV. The proposed model achieves the second lowest false prediction of break of 9.3%. Although Transformer can achieve the lowest false prediction of break of 5.1%, it has a relatively high false prediction of nonbreak of 22.1%. Besides, 1DCNN achieves the lowest false prediction of nonbreak of 9.7%, while the proposed model has a false prediction of nonbreak of 14.7%. However, 1DCNN missed more breaks with false prediction of break of 18.7% compared to the proposed model with false prediction of break of 9.3%. Since the false prediction of break and false prediction of nonbreak are both related to the prediction performance of a single class, they can be misleading in the case of an imbalanced testing dataset. Thus, the balanced accuracy is introduced to measure the overall performance. The proposed model achieves the highest balanced accuracy of 88.0%. On a closer examination of the results, it is found that the good results are largely attributed to a very low false prediction of break of 9.3%, with a comparable false prediction of nonbreak of 14.7%. This indicates that most of the rail break samples can be correctly classified, which is very important for preventing rail breaks.

The comparative results above demonstrate the superiority of the proposed approach. To evaluate the importance of each module in the proposed approach, three competing approaches are constructed, i.e., "proposed approach without TimeGAN," "proposed approach without BiLSTM," and "proposed approach without attention." For each of the three approaches, one of the three modules is removed from the proposed approach, where the removed BiLSTM module is replaced with a standard LSTM layer and the removed attention is replaced with a fully connected layer with the same number of neurons for a fair comparison. The influence of the three modules on the rail break prediction is compared with the same metrics, including false prediction of break, false prediction of nonbreak, and balanced accuracy.

The performance comparisons are shown in Table V. It can be observed that the proposed approach without TimeGAN tends to be overwhelmed by the majority class (nonbreak) and results in poor performance for the minority class (break). After applying TimeGAN, the proposed approach performs well. This indicates that creating synthetic samples for the minority class can relieve the data imbalance problem and improve the overall prediction performance. Also, it can be seen from Table V that the proposed approach without BiLSTM achieves a balanced accuracy of 86.4%, while the proposed approach can acquire a balanced accuracy of 88.0% under the same circumstance. This confirms that the BiLSTM unit can capture more temporal information in sequential data than the standard one. Moreover, the proposed approach without attention achieves a balanced accuracy of 86.9%, which is worse than the proposed approach of 88.0%. It is also observed that the false prediction of break is significantly reduced from 16.1% to 9.3% when attention is used. For railway operators, reducing false prediction of break is important to prevent rail breaks. Thus, it is believed that the proposed model performs better in rail break prediction.

C. Complexity Analysis

The collected measurements in the field are updated on a daily basis and the data volumes are enormous. When making a prediction for the whole track, more than 35 000 sets of samples need to be processed by the prediction model. Thus, the complexity of the prediction model needs to be assessed to check its potential for deployment in railway systems, which are usually equipped with limited device memories and computation capability. Two indexes, including trainable parameters and prediction speed, are used to evaluate the complexity of the model in space and time [57]. For a fair comparison, the experiments are conducted on the same 1000 new samples and under the same hardware and software configuration, as listed in Table VI.

The results of the complexity analysis are shown in Table VII. It is observed that Transformer, CNN-LSTM, and MLP involve more trainable parameters and as such require more computer memories than the proposed model. Although both the 1DCNN and the proposed model have fewer trainable parameters, the prediction performance of the 1DCNN is worse than that of the proposed model. The prediction speed of different models is also shown in Table VII. It can be observed that the proposed model is faster than the Transformer model. The Transformer has to invest a large amount of time in predicting 1000 new samples of 410 s. This dramatically limits its potential usage in railway systems, although it shows competitive performance to predict rail breaks. The proposed model, although not the fastest one, has a competitive prediction speed of 80 s and the best prediction performance, showing its deployment opportunity in field systems for real-time prediction. Therefore, the superior results achieved by the proposed model, along with its simplicity and competitive computational cost, confirm that the proposed model is able to perform real-time prediction.

TABLE IV
PERFORMANCES OF DIFFERENT PREDICTION MODELS

Prediction model	False prediction of break (The lower the better)	False prediction of non-break	Balanced accuracy (The higher the better)
MLP	19.5%	13.3%	83.6%
1DCNN	18.7%	9.7%	85.8%
CNN-LSTM	15.4%	11.5%	86.6%
Transformer	5.1%	22.1%	86.4%
AM-BRNN	9.3%	14.7%	88.0%

TABLE V
PERFORMANCES OF COMPARATIVE APPROACHES

Comparative approach	False prediction of break (The lower the better)	False prediction of non-break	Balanced accuracy (The higher the better)
Without TimeGAN	43.3%	4.3%	76.2%
Without BiLSTM	11.9%	15.3%	86.4%
Without Attention	16.1%	10.2%	86.9%
AM-BRNN with TimeGAN	9.3%	14.7%	88.0%

TABLE VI
HARDWARE AND SOFTWARE CONFIGURATIONS FOR THE PREDICTION SPEED TESTING

Hardware/software	Configuration
Processor	1.8 GHz 4-Core Intel Core i7
Memory	16 GB 2133 MHz DDR4
Drive	NVMe
Graphics	UHD Graphics 620
Software	Python 3.7

TABLE VII
RESULTS OF COMPLEXITY ANALYSIS BASED ON TRAINABLE PARAMETERS AND PREDICTION SPEED

Prediction model	Trainable parameters	Prediction speed (s)
MLP	202, 050	20
1DCNN	80, 386	30
CNN-LSTM	138, 242	60
Transformer	463, 738	410
AM-BRNN	77, 954	80

TABLE VIII
RAIL BREAKS RECORDED BETWEEN 01/02/2021 AND 15/05/2021

Rail break event	Date found
1	13/02/2021
2	22/02/2021
3	03/03/2021
4	17/03/2021
5	24/03/2021
6	28/03/2021
7	06/04/2021
8	16/04/2021
9	20/04/2021
10	**05/05/2021**
11	**15/05/2021**

TABLE IX
RESULTS OF THE REAL-LIFE VALIDATION

Description		
Details of track section	Length of lines	350 km
	No. of samples (20m segment)	17500
	Actual breaks	11
Prediction result	Predicted actual breaks	9
	Missed breaks	2
Metric	False prediction of break	18.2%
	False prediction of non-break	8.2%
	Balanced accuracy	86.8%

D. Real-Life Validation

In Section IV-B, the proposed prediction model is evaluated using randomly selected training and test datasets consisting of only historical rail breaks. In this section, the proposed prediction model is trained using historical rail breaks, but predictions are based on newly collected data, and prediction performance is tested using future rail breaks.

The whole track section is 350 km long. The historical data used for training the prediction model were between 01/01/2018 and 01/12/2020. After the model development, prediction was made on 01/02/2021 based on the newly collected data between 01/12/2020 and 01/02/2021. The newly collected data included 17 500 samples, where each sample represented an adjacent segment of 20 m. After the prediction, there were 11 rail breaks occurred in the study section between 15/02/2021 and 15/05/2021, as listed in Table VIII.

The prediction results were verified according to the future "unseen" 11 rail breaks and presented in Table IX. All the nine rail breaks before 01/05/2021 were correctly predicted. Only two rail breaks after 01/05/2021 were missed, as highlighted in bold in Table VIII. One possible reason was that only data before 01/02/2021 were used for prediction. Those two sites probably developed rail cracks after 01/02/2021. If much newer data were used, these two missed rail breaks could be predicted. In general, nine out of 11 rail breaks were successfully predicted with a low false prediction of nonbreak of 8.2% along the whole track section. This proves the effectiveness of the proposed prediction model. The results have crucial implications for infrastructure managers to carry out timely and valid maintenance activities against rail breaks.

E. Cause Analysis

Given the prediction results in Section IV-C, the SHAP method is employed to calculate the SHAP values for predicted rail breaks. Since the SHAP values are the measure of the

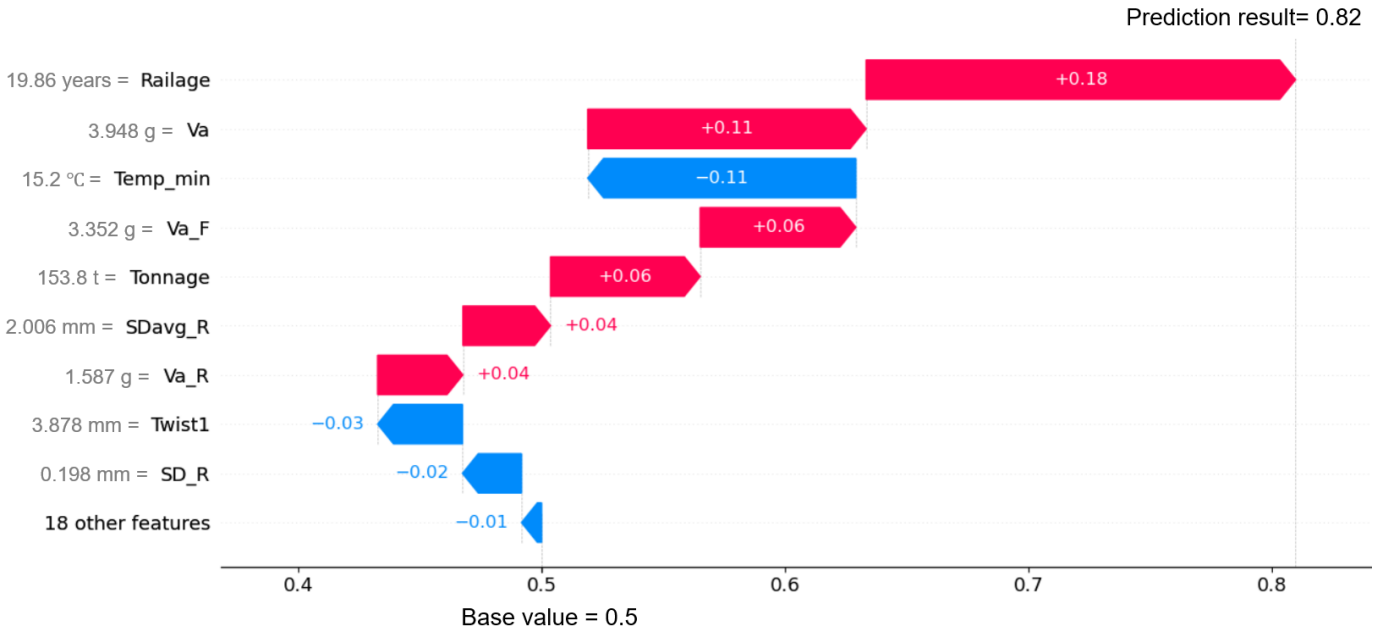


Fig. 12. SHAP values of parameters for a predicted rail break event.

parameter importance, parameters with large positive SHAP values are considered as the most important ones. A predicted rail break that occurred on 22/02/2021 is taken as an example. The prediction result obtained by the AM-BRNN model for this sample is 0.82. This indicates that this is a potential rail break because, in the prediction model, a sample is classified as a rail break if its prediction result is greater than 0.5. Fig. 12 shows the calculated SHAP values of each parameter for this predicted rail break. The left vertical axis indicates the parameters with their actual value. The corresponding SHAP values are shown on the red or blue arrows and are also represented by the length of arrows. Fig. 12 explains how to get from the base value (0.5) that would be predicted if we do not know any parameters to the current prediction result (0.82). The red arrows represent positive SHAP values, which push the prediction toward a higher value, while the blue arrows represent negative SHAP values, which push the prediction toward a lower value. Looking at Fig. 12, the rail age of 19.86 years, the vertical acceleration of 3.948 g, and the annual tonnage of 153.8 t are the most contributing parameters toward rail break prediction.

Based on the cause analysis, maintenance can be performed accordingly to prevent rail breaks. For example, high rail age can be reduced by rail joint replacement, rail defect removal, renewal, or rerailing. To reduce the effect of high vertical acceleration, grinding is usually recommended because previous studies have demonstrated that vertical acceleration was relevant to the surface defects [17].

V. Conclusion

This article adapts recent deep learning techniques and develops a new rail break prediction approach using daily monitoring data from in-service trains. To mitigate the problem of data imbalance and preserve the temporal dynamics in data, a TimeGAN is employed to learn the distribution of the real rail break samples and generate synthetic rail break samples. To make an accurate prediction, a feature-level AM-BRNN is proposed to enhance the feature extraction and capture two-direction dependencies from sequential data. The proposed approach is implemented on a three-year dataset collected from a section of railroads in Australia. Extensive comparative results demonstrate that compared with the current models, the proposed approach achieves superior performance for the rail break prediction using in-service train data. Furthermore, the complexity analysis of the proposed prediction model is conducted, confirming its potential to be deployed in standalone devices in field for real-time prediction. The results of a real-life validation demonstrate the effectiveness of the proposed approach, which capture nine out of 11 rail breaks three months ahead of time with a false prediction of nonbreak of 8.2%. Given the prediction results, a causes analysis is performed for a certain predicted rail break using the SHAP method. By implementing the proposed approach, the railroad can predict rail breaks three months ahead of time and plan maintenance accordingly to prevent rail break. Currently, parameters that vary with time (e.g., in-service train data) and parameters that do not vary with time (e.g., annual tonnage) are fed into the model together for rail break prediction. Future work will focus on providing predictions by addressing those two formats of data separately through a multichannel network to strengthen the utilization of information and further improve the prediction performance.

References

[1] G. Chattopadhyay and S. Kumar, "Parameter estimation for rail degradation model," *Int. J. Performability Eng.*, vol. 5, no. 2, p. 119, 2009.
[2] S. Vesković, J. Tepić, M. Ivić, G. Stojić, and S. Milinković, "Model for predicting the frequency of broken rails," *Metalurgija*, vol. 51, no. 2, pp. 221–224, 2012.
[3] L. Bai, R. Liu, F. Wang, Q. Sun, and F. Wang, "Estimating railway rail service life: A rail-grid-based approach," *Transp. Res. A, Policy Pract.*, vol. 105, pp. 54–65, Nov. 2017.

[4] F. Ghofrani, N. K. Chava, and Q. He, "Forecasting risk of service failures between successive rail inspections: A data-driven approach," *J. Big Data Analytics Transp.*, vol. 2, no. 1, pp. 17–31, Apr. 2020.

[5] F. Ghofrani, S. Yousefianmoghadam, Q. He, and A. Stavridis, "Rail breaks arrival rate prediction: A physics-informed data-driven analysis for railway tracks," *Measurement*, vol. 172, Feb. 2021, Art. no. 108858.

[6] C. T. Dick, C. P. L. Barkan, E. R. Chapman, and M. P. Stehly, "Multivariate statistical model for predicting occurrence and location of broken rails," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1825, no. 1, pp. 48–55, Jan. 2003.

[7] D. Schafer and C. Barkan, "A hybrid logistic regression/neural network model for the prediction of broken rails," in *Proc. 8th World Congr. Railway Res.*, Seoul, South Korea, 2008, pp. 1–10.

[8] Z. Zhang, K. Zhou, and X. Liu, "Broken rail prediction with machine learning-based approach," in *Proc. Joint Rail Conf.*, Apr. 2020, Art. no. V001T08A014.

[9] F. Ghofrani, H. Sun, and Q. He, "Analyzing risk of service failures in heavy haul rail lines: A hybrid approach for imbalanced data," *Risk Anal.*, vol. 42, no. 8, pp. 1852–1871, Aug. 2022.

[10] P. F. Westeon, C. S. Ling, C. Roberts, C. J. Goodman, P. Li, and R. M. Goodall, "Monitoring vertical track irregularity from in-service railway vehicles," *Proc. Inst. Mech. Eng., F, J. Rail Rapid Transit Inst. Mech. Eng.*, vol. 221, no. 1, pp. 75–88, Jan. 2007.

[11] G. J. Yeo, "Monitoring railway track condition using inertial sensors on an in-service vehicle," Ph.D. dissertation, School Eng., Dept. Electron., Elect. Syst. Eng., Univ. Birmingham, Birmingham, U.K., 2017.

[12] J. Xie, J. Huang, C. Zeng, S.-H. Jiang, and N. Podlich, "Systematic literature review on data-driven models for predictive maintenance of railway track: Implications in geotechnical engineering," *Geosciences*, vol. 10, no. 11, p. 425, Oct. 2020.

[13] Z. Wei, A. Boogaard, A. Nunez, Z. Li, and R. Dollevoet, "An integrated approach for characterizing the dynamic behavior of the wheel–rail interaction at crossings," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 10, pp. 2332–2344, Oct. 2018.

[14] H. Ishii, Y. Fujino, Y. Mizuno, and K. Kaito, "The study of train intelligent monitoring system using acceleration of ordinary train," in *Proc. 1st Asia–Pacific Workshop Struct. Health Monit.*, Yokohama, Japan, Dec. 2006.

[15] M. Bocciolone, A. Caprioli, A. Cigada, and A. Collina, "A measurement system for quick rail inspection and effective track maintenance strategy," *Mech. Syst. Signal Process.*, vol. 21, no. 3, pp. 1242–1254, 2007.

[16] M. Molodova, Z. Li, A. Núñez, and R. Dollevoet, "Automatic detection of squats in railway infrastructure," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1980–1990, Oct. 2014.

[17] A. Jamshidi et al., "A decision support approach for condition-based maintenance of rails based on big data analysis," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 185–206, Oct. 2018.

[18] P. Tešić, S. Jovanović, and M. Dick, "Analysis of vehicle/track interaction measurement data using the V/TI Monitor system," *Gradevinar*, vol. 70, no. 2, pp. 105–119, 2018.

[19] C. Zeng, J. Huang, J. Xie, B. Zhang, and B. Indraratna, "Prediction of mud pumping in railway track using in-service train data," *Transp. Geotechnics*, vol. 31, Nov. 2021, Art. no. 100651.

[20] Z. Li, T. Zheng, Y. Wang, Z. Cao, Z. Guo, and H. Fu, "A novel method for imbalanced fault diagnosis of rotating machinery based on generative adversarial networks," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–17, 2020.

[21] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–47.

[22] G. Yang, Y. Zhong, L. Yang, H. Tao, J. Li, and R. Du, "Fault diagnosis of harmonic drive with imbalanced data using generative adversarial network," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[23] W. Zhang, X. Li, X.-D. Jia, H. Ma, Z. Luo, and X. Li, "Machinery fault diagnosis with imbalanced data using deep generative adversarial networks," *Measurement*, vol. 152, Feb. 2020, Art. no. 107377.

[24] R. Wang, S. Zhang, Z. Chen, and W. Li, "Enhanced generative adversarial network for extremely imbalanced fault diagnosis of rotating machine," *Measurement*, vol. 180, Aug. 2021, Art. no. 109467.

[25] J. Zhong, Z. Liu, C. Yang, H. Wang, S. Gao, and A. Nunez, "Adversarial reconstruction based on tighter oriented localization for catenary insulator defect detection in high-speed railways," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1109–1120, Feb. 2022.

[26] Y. Lyu, Z. Han, J. Zhong, C. Li, and Z. Liu, "A generic anomaly detection of catenary support components based on generative adversarial networks," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 5, pp. 2439–2448, May 2020.

[27] H. Liu, H. Zhao, J. Wang, S. Yuan, and W. Feng, "LSTM-GAN-AE: A promising approach for fault diagnosis in machine health monitoring," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–13, 2021.

[28] Y. Zhang, M. Liu, Y. Chen, H. Zhang, and Y. Guo, "Real-time vision-based system of fault detection for freight trains," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 5274–5284, Jul. 2019.

[29] S. Meng, S. Kuang, Z. Ma, and Y. Wu, "MtlrNet: An effective deep multitask learning architecture for rail crack detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.

[30] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[31] B. Zhao, C. Cheng, Z. Peng, X. Dong, and G. Meng, "Detecting the early damages in structures with nonlinear output frequency response functions and the CNN-LSTM model," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9557–9567, Dec. 2020.

[32] Q. Xie, G. Tao, B. He, and Z. Wen, "Rail corrugation detection using one-dimensional convolution neural network and data-driven method," *Measurement*, vol. 200, Aug. 2022, Art. no. 111624.

[33] D. Huang, S. Li, N. Qin, and Y. Zhang, "Fault diagnosis of high-speed train bogie based on the improved-CEEMDAN and 1-D CNN algorithms," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[34] J. Zhong, Z. Liu, Z. Han, Y. Han, and W. Zhang, "A CNN-based defect inspection method for catenary split pins in high-speed railway," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 8, pp. 2849–2860, Aug. 2018.

[35] Q. Chen, G. Nicholson, C. Roberts, J. Ye, and Y. Zhao, "Improved fault diagnosis of railway switch system using energy-based thresholding wavelets (EBTW) and neural networks," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2020.

[36] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, no. 5, pp. 602–610, 2005.

[37] T. de Bruin, K. Verbert, and R. Babuska, "Railway track circuit fault diagnosis using recurrent neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 523–533, Mar. 2016.

[38] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[39] H. Wang, T. Men, and Y.-F. Li, "Transformer for high-speed train wheel wear prediction with multiplex local–global temporal fusion," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.

[40] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4768–4777.

[41] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1135–1144.

[42] S. M. Lundberg et al., "Explainable machine-learning predictions for the prevention of hypoxaemia during surgery," *Nature Biomed. Eng.*, vol. 2, no. 10, pp. 749–760, 2018.

[43] S. Mangalathu, S.-H. Hwang, and J.-S. Jeon, "Failure mode and effects analysis of RC members based on machine-learning-based Shapley additive exPlanations (SHAP) approach," *Eng. Struct.*, vol. 219, Sep. 2020, Art. no. 110927.

[44] A. Núñez et al., "A condition-based maintenance methodology for rails in regional railway networks using evolutionary multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–7.

[45] R. Mohammadi, Q. He, F. Ghofrani, A. Pathak, and A. Aref, "Exploring the impact of foot-by-foot track geometry on the occurrence of rail defects," *Transp. Res. C, Emerg. Technol.*, vol. 102, pp. 153–172, May 2019.

[46] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[47] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, *arXiv:1511.05644*.

[48] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 3285–3292.

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

[50] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 3–19.

[51] J. Fu et al., "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3146–3154.

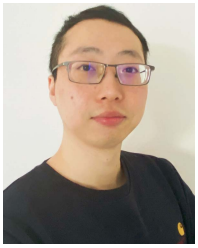
[52] Q. Zhou, Z. Qu, and C. Cao, "Mixed pooling and richer attention feature fusion for crack detection," *Pattern Recognit. Lett.*, vol. 145, pp. 96–102, May 2021.
[53] J. Cheng, R. Liang, and L. Zhao, "DNN-based speech enhancement with self-attention on feature dimension," *Multimedia Tools Appl.*, vol. 79, nos. 43–44, pp. 32449–32470, Nov. 2020.
[54] Z. Wang, Q. She, and J. Zhang, "MaskNet: Introducing feature-wise multiplication to CTR ranking models by instance-guided mask," 2021, *arXiv:2102.07619*.
[55] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.
[56] G. Forestier, F. Petitjean, H. A. Dau, G. I. Webb, and E. Keogh, "Generating synthetic time series to augment sparse datasets," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2017, pp. 865–870.
[57] Y. Zhang, M. Liu, Y. Yang, Y. Guo, and H. Zhang, "A unified light framework for real-time fault detection of freight train images," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7423–7432, Nov. 2021.

Cheng Zeng received the M.S. degree in civil engineering from Southwest Jiaotong University, Chengdu, China, in 2018. She is currently pursuing the Ph.D. degree with the Discipline of Civil, Surveying and Environmental Engineering, The University of Newcastle, Callaghan, NSW, Australia.

Her current research interests include signal processing, deep learning, and their applications in railway fault prediction.

Jinsong Huang is currently a Professor at the Priority Research Centre for Geotechnical Science and Engineering, The University of Newcastle, Callaghan, NSW, Australia. He has published over 100 journal articles on the risk assessment of slope stability and landslides, the modeling of spatial variability, stress integration techniques for elastoplastic models, the contact dynamics of granular media, the analysis of hydraulic fracturing, and the predictive maintenance of railway tracks. His research interests include risk assessment in geotechnical engineering and computational geomechanics.

He received the Regional Contribution Award from the International Association of Computer Methods and Advances in Geomechanics at its international conference in Kyoto in 2014 and the GEOSNet Award from the Geotechnical Safety Network in 2017. He will serve as the Conference Chair for the 8th International Symposium on Geotechnical Safety and Risk to be held at The University of Newcastle in December 2022.



Hongrui Wang (Member, IEEE) received the B.S. degree in electrical engineering and automation from the School of Electrical Engineering, Southwest Jiaotong University, Chengdu, China in 2012, and the Ph.D. degree in railway engineering from the Department of Engineering Structures, Delft University of Technology, Delft, The Netherlands, in 2019.

He was a Post-Doctoral Researcher with the Department of Engineering Structures, Delft University of Technology, until 2020, where he is currently an Assistant Professor. His research interests include machine learning, digital twins, and their applications in the structural health monitoring and designs of railway infrastructures and vehicles.

Dr. Wang is an Associate Editor for the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.

Jiawei Xie received the M.S. degree in civil engineering from Central South University, Changsha, China, in 2019. He is currently pursuing the Ph.D. degree with the Discipline of Civil, Surveying and Environmental Engineering, The University of Newcastle, Callaghan, NSW, Australia.

His current research interests include probabilistic geotechnical site characterization, signal processing, deep learning, and their applications in railway fault prediction.

Shan Huang received the M.S. degree in civil engineering from Central South University, Changsha, China, in 2019. She is currently pursuing the Ph.D. degree with the Discipline of Civil, Surveying and Environmental Engineering, The University of Newcastle, Callaghan, NSW, Australia.

Her current research interests include computational geomechanics, risk assessment in geotechnical engineering, and Bayesian inference.