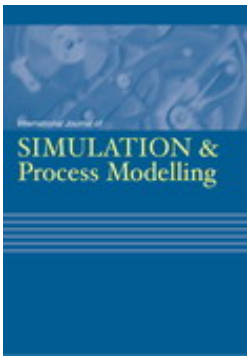


International Journal of Simulation and Process Modelling

ISSN online: 1740-2131 - ISSN print: 1740-2123
https://www.inderscience.com/ijspm

Simulation-driven fault diagnosis for track circuits using multi-scale convolution and transformers under imbalanced data conditions

Jundong Fu, Xiaojun Yuan

DOI: 10.1504/IJSPM.2025.10073120

Article History:

Received:	16 January 2025
Last revised:	26 March 2025
Accepted:	02 April 2025
Published online:	01 September 2025

Simulation-driven fault diagnosis for track circuits using multi-scale convolution and transformers under imbalanced data conditions

Jundong Fu* and Xiaojun Yuan

School of Electrical and Automation Engineering,
East China Jiaotong University,
Nanchang, China
Email: fujundongzjb@163.com
Email: xiaojun18320457053@163.com
*Corresponding author

Abstract: With the increasing complexity of railway systems, diagnosing faults in ZPW-2000A track circuits poses significant challenges, especially under imbalanced data conditions. This paper introduces a novel simulation-driven fault diagnosis framework combining multi-scale convolution, transformer encoders, and LDAM loss to enhance classification accuracy. The method extracts short- and long-term features from time-series data, effectively addressing data imbalance through gated convolution and optimised classification boundaries. Simulation experiments on 1,600 samples demonstrated a 98.75% accuracy, significantly outperforming existing approaches. Ablation studies validated the contribution of each module. The findings show potential for real-time fault detection and improved railway safety. Future work includes validating the model on real-world data and optimising its computational efficiency. This research contributes to simulation and process modelling by offering a robust, scalable solution for fault diagnosis, aligning with industrial needs and advancing the state-of-the-art in imbalanced data analysis.

Keywords: fault diagnosis; imbalanced data handling; simulation-based methods; railway safety systems; deep learning models; process modelling.

Reference to this paper should be made as follows: Fu, J. and Yuan, X. (2025) 'Simulation-driven fault diagnosis for track circuits using multi-scale convolution and transformers under imbalanced data conditions', *Int. J. Simulation and Process Modelling*, Vol. 22, Nos. 1/2, pp.29–46.

Biographical notes: Jundong Fu is an Associate Professor at the School of Electrical and Automation Engineering at East China Jiaotong University (ECJTU), Nanchang, China. His main research field is building electrical and intelligent and he has extensive experience in the teaching and research of building electrical. At present, his main research areas are intelligent building electrical control, building electrical fault diagnosis and building electrical automation design.

Xiaojun Yuan is currently pursuing a Master's degree at ECJTU. He focuses on fault diagnosis, deep learning, and simulation modelling, with a research emphasis on establishing a digital twin system for railway circuits.

1 Introduction

The global railway industry is rapidly developing, bringing significant benefits. However, this growth also raises critical safety concerns, making railway safety a fundamental requirement for sustainable industry progress. Track circuits are an essential component of railway equipment, as they can detect whether a section of track is occupied by a train and can also be used to transmit information. The ZPW-2000A track circuit plays a crucial role in the operation of railways. However, diagnosing faults in the ZPW-2000A track circuit is challenging due to its complex structure, which includes both indoor and outdoor components, multiple devices, and extensive circuitry with numerous potential failure points. In the past, faults in track circuits were mainly identified through manual inspection and step-by-step troubleshooting, such as routine line inspections. Such manual diagnostics are time-consuming and labour-intensive, failing to meet the requirements for efficient fault diagnosis in track circuits. When a fault remains unresolved for an extended period, it disrupts the normal train scheduling and operation plans, leading to delays. This may even result in risks such as train collisions or derailments, which significantly affect the efficiency and safety of train operations. There is an urgent need for a rational, effective method to quickly, accurately, and efficiently diagnose faults in track circuits, presenting opportunities and challenges for track circuit fault diagnosis.

Early fault diagnosis algorithms were mainly based on rule-based, model-based, and statistical methods. However, these approaches faced limitations, including challenges in knowledge acquisition, heavy reliance on system models, and poor performance in addressing complex and nonlinear problems. With the increase in data volume and computational power, traditional fault diagnosis methods have gradually exposed their limitations, particularly when dealing with complex systems and nonlinear issues, where their performance is often suboptimal. To overcome these challenges, in recent years, data-driven deep learning methods have gradually become an important development direction in the field of fault diagnosis. Unlike traditional methods, deep learning extracts features from large datasets, captures complex nonlinear relationships, and eliminates the need for manual rules or precise models. Through an end-to-end learning process, deep learning not only improves diagnostic accuracy but also enables the handling of more complex and dynamic systems. The BP neural network was formally introduced and popularised by Rumelhart et al. (1986), laying the foundation for current deep learning methods. Convolutional neural networks (CNNs), proposed by LeCun et al. (1998), have been widely used in computer vision (CV), fault diagnosis, and other fields. The long short-term memory (LSTM), proposed by Hochreiter and Schmidhuber (1997), effectively captures long-term dependencies and has been widely applied in natural language processing, fault diagnosis, and other fields. In fault diagnosis with time-series data, BP neural networks are often used for final feature classification, CNNs use 1D convolutions to extract short-term temporal features and perform dimensionality reduction, and LSTM is suitable for extracting long-term features. Today, researchers continue to improve CNNs or propose new algorithms to extract data features. Due to the limitations of single-scale convolutional kernels in extracting fault features comprehensively, and their performance degradation under strong noise conditions, multi-scale convolution has garnered increasing attention. Shao et al. (2024) proposed an intelligent fault diagnosis method based on a multi-scale deep convolutional neural network (MSD-CNN) model and strong noise data augmentation. This method extracted multi-scale features from raw fault signals using multi-scale cascaded convolutional kernels. Additionally, an enhancement strategy for strong noise was employed to increase the quantity and diversity of training samples, enabling the model to learn richer deep features during training. A new anti-noise multi-scale convolutional neural network (AM-CNN) was proposed by Jin et al. (2022) to address the problem of compound fault diagnosis under varying noise levels. They designed a multi-scale convolution block to extract multi-scale features and address the strong coupling of input information, achieving high accuracy and F1 macro results. However, multi-scale convolution is limited to short-term feature extraction and struggles to capture long-term dependencies in complex fault patterns, reducing its effectiveness in dynamic and complex systems. Bahdanau et al. (2014) first applied the attention mechanism to address challenges in machine translation. Since then, attention mechanisms have continued to evolve, proving to be effective in extracting long-term features from data, making them particularly suitable for tasks with long-term dependencies. The transformer model was introduced by Vaswani et al. (2017) to solve sequence-to-sequence problems in natural language processing. Following this, the transformer model was improved and applied to fields such as CV and fault diagnosis. The transformer's use of multi-head self-attention layers led to significant success in applications involving time-series data. Hu et al. (2022) introduced an attention mechanism into the basic model, accurately establishing the relationship between fault features and fault patterns, while also improving diagnostic accuracy in complex noise environments. Finally, through comparison with other methods, the effectiveness and stability of this approach were validated. Only the encoder part of the transformer was utilised by Gorishniy et al. (2021) to construct their model, as data classification does not require a decoder and the encoder alone is sufficient for feature extraction. However, attention mechanisms and their improvements are particularly suitable for extracting long-term features from time-series data, while their effectiveness in short-term feature extraction is not as pronounced. Subsequently, researchers began combining convolutional layers with attention mechanisms to leverage convolution's ability to extract local features and utilise attention mechanisms to capture global dependencies, thereby improving the overall performance of models in time-series data processing. Ke et al. (2024) proposed a convolutional neural network with multi-head self-attention (CNN-MHSA) for track circuit fault diagnosis. This model used convolutional layers to locally perceive and extract features from sequential data, while the multi-head self-attention layer established long-range dependencies and captured global features from the data. In comparison with other algorithms, the model significantly improved the overall performance of ZPW-2000A track circuit fault diagnosis. To address the issue of fault category overlap in simulated circuits, Yang et al. (2023) used multi-scale convolution to aggregate features from multiple receptive fields and introduced an attention mechanism to enhance useful features while suppressing irrelevant ones. The results showed that, compared to other recent state-of-the-art models, this model outperformed in early-stage multi-fault classification performance. Most of the aforementioned studies primarily focus on better extracting long-term or short-term features from data and do not take into account the potential impact of imbalanced data, which may lead to suboptimal performance when dealing with imbalanced datasets.

Figure 1 System structure and voltage monitoring points of the ZPW-2000A

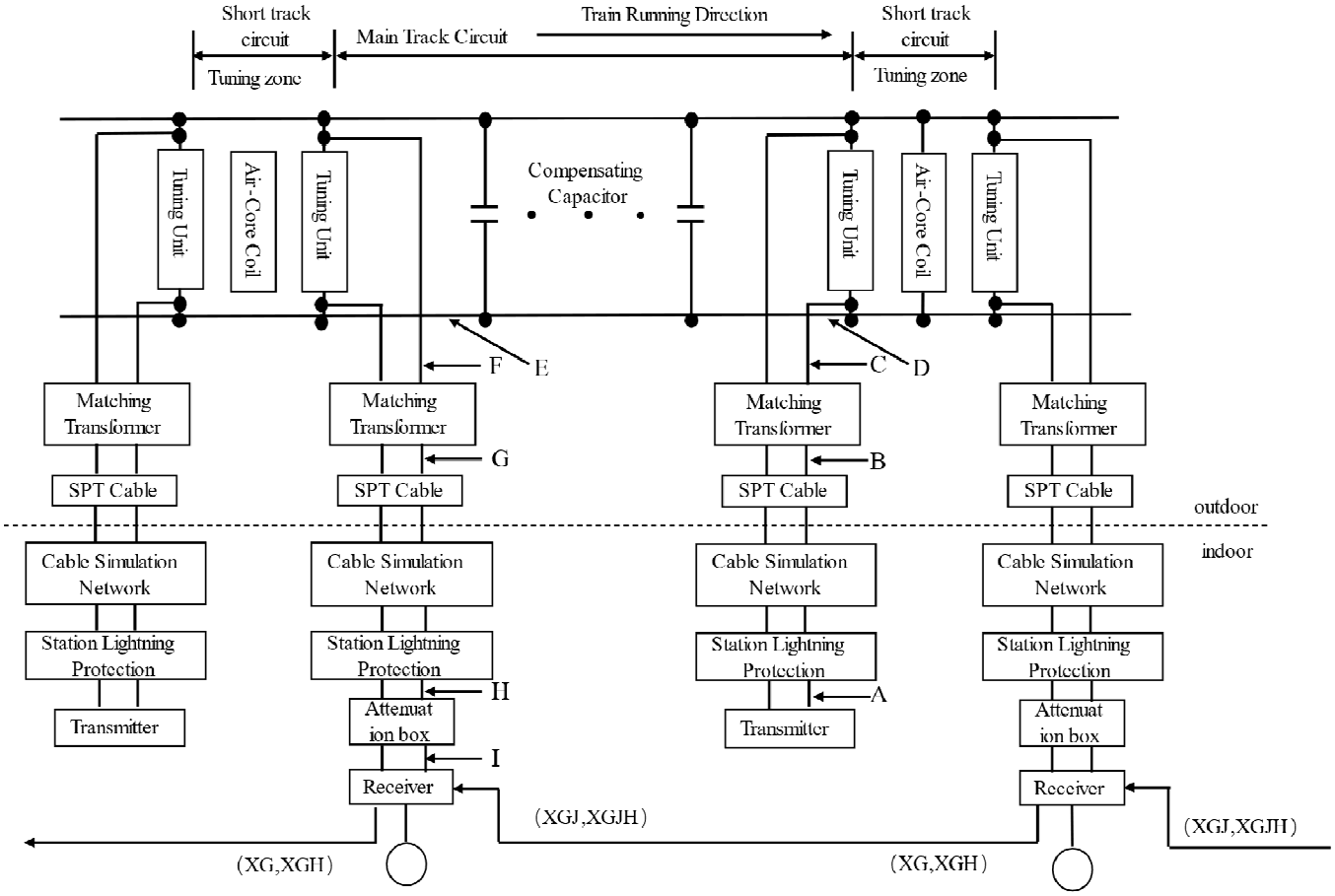
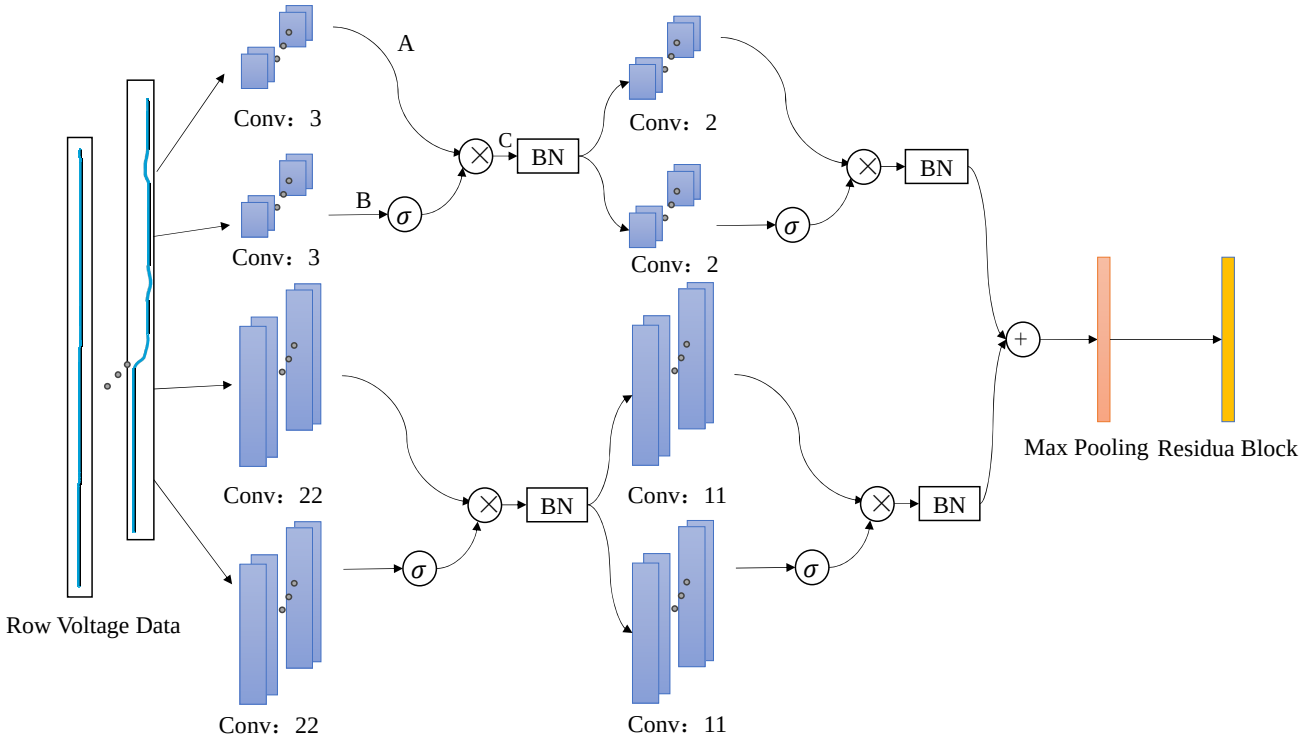


Figure 2 The multi-scale convolutional structure used for extracting short-term features at different scales (see online version for colours)



In practical applications, track circuits typically operate under normal conditions, with faults being rare. This leads to a significant data imbalance, where normal samples vastly outnumber fault samples. Such imbalance can bias the model towards the majority class, causing it to overlook minority fault samples and reducing fault detection accuracy. During training, the model might over-optimise predictions for the normal state, leading to insufficient fault detection capabilities and potentially resulting in missed or false alarms. Therefore, when performing feature extraction and classification, the impact of data imbalance must be considered, and appropriate handling methods should be employed. Zhang et al. (2022) pointed out that methods for addressing imbalanced data can generally be categorised into three approaches: data augmentation-based, feature learning-based, and classifier design-based. From the perspective of data augmentation, Arafa et al. (2022) employed SMOTE to oversample the training data, followed by the application of DBSCAN to detect and remove noise. Subsequently, SMOTE was applied again to rebalance the dataset before feeding it into the underlying classifier. The results demonstrated that this approach improved the classifier's performance and outperformed both the original dataset and the performance of SMOTE in terms of various metrics. A new generator and discriminator for generative adversarial networks (GANs) were designed by Zhou et al. (2020), using a global optimisation scheme to generate more discriminative fault samples, which were validated through rolling bearing experiments. From the feature learning perspective, Heinrich et al. (2021) discussed how deep learning models, such as gated convolutional neural networks (GCNNs), can be used to handle time-series data in next-event prediction tasks. The study concluded that due to its dynamic feature selection capability, GCNN can adapt to different data distributions, especially when facing imbalanced data, by adaptively adjusting the weights of convolutional kernels, making it more effective at learning features from the minority class. A normalised CNN was proposed by Zhao et al. (2020) for diagnosing rolling bearings under imbalanced data and variable operating conditions, and the method's effectiveness was validated based on two popular experimental datasets, demonstrating its robust diagnostic performance across various scenarios. From the classifier design perspective, Sun et al. (2023) addressed the common issue of data imbalance in wind turbine fault diagnosis by establishing a coarse learner and multiple fine learners, integrating multiple models, and validating the method through experiments based on simulated data and real SCADA measurements from wind farms. Focal loss, a new loss function, was used by Hammad et al. (2022) to optimise deep learning models, ultimately improved detection accuracy by 9%. Liang et al. (2020) proposed the LR-SMOTE algorithm to generate new samples near the sample centre, avoiding the creation of anomalous samples or altering the data distribution.

Ke et al. (2024) combined convolutional layers with multi-head self-attention mechanisms, while Yang et al. (2023) integrated multi-scale convolution and self-attention mechanisms. However, neither of these approaches uses multi-scale convolution combined with transformer encoders to extract features from time-series data. In the research on handling imbalanced data, most studies focus on one of three methods: data augmentation, feature learning, or classifier design, without comprehensively considering the integration and optimisation of these approaches. As a result, existing methods often fail to leverage the advantages of these techniques fully. Addressing these gaps in the literature, the model proposed in this paper utilises multi-scale convolution combined with transformer encoders to extract multi-level features from time-series data. From the perspective of integrating feature learning and classifier design methods, gated convolutions and the LDAM loss function are employed to tackle the data imbalance issue effectively. This approach not only enhances the model's adaptability to complex dynamic systems but also improves its ability to recognise minority class faults, thereby demonstrating strong performance in track circuit fault diagnosis.

The ZPW-2000A track circuit is a widely used type of track circuit in modern railway signalling systems, capable of detecting train occupancy and transmitting information. Due to its unique operation in complex environments, the ZPW-2000A track circuit faces challenges from various fault types, including circuit interruptions and electrical component failures. By applying advanced fault diagnosis models, not only can the accuracy of fault detection in the track circuit be improved, but the overall safety and operational efficiency of the railway system can also be enhanced. Therefore, the ZPW-2000A track circuit is an ideal target for research. For track circuit fault diagnosis under imbalanced data conditions, based on the above research, we combine several advanced techniques to enhance the model's diagnostic performance. First, we use multi-scale convolution to extract short-term features from different scales in the time-series data, helping the model better capture local variations. At the same time, we use the encoder part of the transformer to extract global features from the data, capturing long-term dependencies, thereby enhancing the model's ability to identify complex fault patterns. Finally, we use a fully connected layer to classify the extracted features, further improving classification accuracy. When handling imbalanced data, we optimise from the feature learning and classifier design perspectives. By introducing gated convolution, we effectively scale the spatial features, allowing the model to flexibly capture different types of features in multi-dimensional space. Additionally, we employ the LDAM loss function, which adjusts the classification boundaries during training, shifting them towards the majority class, thereby reducing bias caused by imbalanced data and improving the classification of minority fault types.

The main tasks of this paper can be summarised as follows:

1 This paper proposes a network architecture that combines multi-scale convolution with gated convolution, followed by transformer encoder and fully connected layers, and uses the LDAM loss function. It can effectively extract both short-term and long-term features, and alleviate the impact of imbalanced data.

2 This paper conducted a systematic simulation of the practical ZPW-2000A track circuit and obtained data through simulated faults. By comparing the simulated data with real-world data, the rationality of the proposed system is validated.

3 Ablation experiments were performed using the simulated data to verify the functionality of each module. The results, compared with other algorithms, demonstrated the superiority of the proposed method.

The fault diagnosis model proposed in this paper can quickly and effectively identify the causes of track circuit faults, reducing the time required for fault detection and ensuring the efficiency and safety of railway operations. At the same time, the established ZPW-2000A track circuit simulation model also plays a role in data augmentation by generating fault data, providing support for subsequent research.

The remainder of this paper is organised as follows: Section 2 reviews the state-of-the-art fault diagnosis methods and their limitations. Section 3 presents the proposed hybrid framework. Section 4 validates the framework through simulation and experiments, followed by a discussion of its practical implications in Section 5. Finally, Section 6 concludes the paper and outlines future work.

2 State-of-the-art in fault diagnosis and simulation

Traditional fault diagnosis methods mainly include rule-based, model-based, and statistical approaches. Rule-based methods rely on expert experience and predefined rules to identify faults and are suitable for simple and well-known fault patterns. Model-based methods involve creating mathematical models to simulate the normal behaviour of the system, thereby identifying abnormal states. Statistical methods, on the other hand, rely on data analysis and pattern recognition, using a large amount of historical data to predict and diagnose faults. In contrast, modern techniques such as deep learning methods, by automatically learning a vast number of complex patterns and features, can handle more complex and dynamic fault diagnosis tasks. They perform exceptionally well, especially when dealing with large datasets and intricate features, making them the dominant fault diagnosis approach in contemporary systems.

In deep learning, networks such as CNN and LSTM are widely used in fault diagnosis tasks. CNN, through its convolutional layers, is effective at extracting local features, making it suitable for processing input data with spatial structures, such as images or local feature extraction from time series. It can automatically learn multi-level feature representations, offering significant advantages, especially when dealing with complex fault patterns. LSTM, on the other hand, excels at handling sequential data by introducing memory cells and gating mechanisms, enabling it to capture long-term dependencies in time series, making it well-suited for fault detection and prediction in dynamic systems. Often, CNN and LSTM are combined to allow the model to better adapt to multi-dimensional and multi-time-scale data inputs, thereby improving the accuracy and robustness of fault diagnosis. However, both networks struggle with imbalanced data. Under such conditions, the model tends to prioritise the majority class, leading to poor performance on minority class samples. This often results in misclassifying minority faults as majority classes, significantly reducing fault diagnosis accuracy.

To address the impact of noise, Jin et al. (2022) used multi-scale convolution to resolve input data aliasing, while Shao et al. (2024) also applied multi-scale convolution but incorporates an enhancement strategy for strong noise, increasing the quantity and diversity of training samples. However, multi-scale convolution can only extract short-term features, and there are still limitations in extracting long-term features. Additionally, data augmentation is challenging and lacks stability. Ke et al. (2024) combined CNN with multi-head self-attention mechanisms, and Yang et al. (2023) integrated multi-scale convolution with attention mechanisms. While both approaches combine the ability to extract short-term and long-term features, they can suppress the impact of noise to some extent. However, combining multi-scale convolution with the transformer encoder offers greater advantages, as the transformer encoder can be designed with multiple layers. By extracting features layer by layer, it can learn more complex global features while better suppressing the effects of noise.

When handling imbalanced datasets, common methods can generally be categorised into three types: data augmentation-based, feature learning-based, and classifier design-based approaches, including data resampling, SMOTE, and LDAM loss. Data resampling mitigates imbalance by adjusting the ratio of minority to majority class samples, but it may lead to overfitting or information loss. SMOTE enhances data diversity by synthesising new minority class samples, but it may introduce noise. The class label distribution is addressed by LDAM loss to reduce the influence of the majority class and improve the recognition of minority classes. Each of these methods has its advantages and drawbacks, and the appropriate strategy should be selected based on the specific task at hand.

In the field of railway circuit fault diagnosis, Zheng et al. (2020) modelled the four-terminal network of the ZPW-2000A track circuit to obtain fault data, and this paper also used the four-terminal network theory to simulate and model the ZPW-2000A track circuit. An adaptive fault diagnosis system for track circuits was proposed by Liu et al. (2021), which used pattern recognition to extract fault

features from simulation data, and the experimental results show that the proposed method achieves high fault diagnosis accuracy. De Bruin et al. (2016) employed LSTM in recurrent neural networks, and their experiments concluded that LSTM outperforms convolutional neural networks in track circuit fault diagnosis. Multi-head self-attention was combined with convolutional neural networks (CNN-MHSA) by Ke et al. (2024) for track circuit diagnosis, and it was found that this model significantly improved the overall performance of ZPW-2000A track circuit fault diagnosis when compared with other algorithms. The aforementioned studies primarily focused on how to more effectively extract long-term or short-term features of data, while paid less attention to the potential impact of data imbalance on model performance. In terms of handling imbalanced data, most studies only adopted a single approach and did not consider the combined advantages of various methods for comprehensive processing. The model proposed in this paper uses gated convolutions and the LDAM loss function to handle imbalanced data in both feature extraction and classification. It also incorporates multi-scale convolution and transformer encoders to comprehensively extract data features, which can, to some extent, suppress the impact of noise. Since ZPW-2000A track circuits require rapid and accurate fault diagnosis when failures occur, the deep learning framework can effectively meet the demands of fault diagnosis for ZPW-2000A track circuits.

The main components of the ZPW-2000A track circuit include indoor equipment such as the transmitter, receiver, attenuation box, and cable simulation network, as well as outdoor components such as the SPT cable, matching transformer, air-core coil, and tuning unit. During operation, the transmitter of the main track circuit generates different low-frequency modulated signals. These signals are transmitted via cables to the matching transformer and tuning unit, and then distributed to both the main track circuit and adjacent small track circuits. The signals transmitted to the main track eventually reach the receiver of the current track, while the small track circuit signals, transmitted to the tuning area, are ultimately processed by the receiver of the adjacent track circuit ahead of the train. This forms the execution condition of the small track circuit relay, which is then transmitted to the track circuit receiver of the current segment. The receiver of this segment simultaneously receives both the main track frequency-shift signal and the execution conditions of the adjacent small track relay (XGJ, XGJH), and performs integrated analysis to control the switch of the track circuit relay (GJ) for that segment. This allows the determination of whether the track segment is free or occupied. The system structure of the ZPW-2000A is shown in Figure 1.

Among them, A~I represents the nine voltage monitoring points, which are used to diagnose fault types in the ZPW-2000A track circuit based on voltage changes. During normal operation of the track circuit, faults often manifest as sudden changes or anomalies within localised time steps. However, faults may not be confined to a specific moment in time; instead, they may span multiple time steps and exhibit global characteristics. At the same time, normal operating conditions are more frequent, while faults occur relatively infrequently. Therefore, we choose multi-scale convolution to extract short-term features, transformer to capture global features, and gated convolution along with the LDAM loss function to handle the imbalanced data.

3 Proposed hybrid fault diagnosis framework

3.1 Multi-scale convolution module

CNNs are widely used for feature extraction and dimensionality reduction, achieving excellent results. However, a single CNN with a fixed kernel size struggles to extract multi-scale features, resulting in poor performance on complex data. This has led researchers to explore and adopt multi-scale convolution for feature extraction. For example, in the case of different fault categories being mixed in a simulated circuit, Yang et al. (2023) used multi-scale convolution to aggregate features across multiple receptive fields. Shao et al. (2024) employed multi-scale convolution to better extract features under strong noise conditions. Similarly, in track circuit fault diagnosis, different fault categories can overlap, so this paper adopts multi-scale convolution to capture feature information at various scales, thus improving the network's ability to express multi-scale features. The structure of the multi-scale convolution is shown in the diagram. Since the data is a single time-series dataset with 60 time steps, and each time step has nine voltage features, only one-dimensional convolution is needed for feature extraction along the time axis. The data is first processed through two parallel convolutional layers. The convolutional layers with kernel sizes of 3 and 2 are used to extract short-term local features, while the layers with kernel sizes of 22 and 11 capture slightly longer-term local features. Additionally, batch normalisation (BN) is applied after each convolutional layer to maintain the stability of the data. After processing, the output of each convolution path has 16 channels and 29 time steps (the number of channels represents the number of features at each time step). The outputs from the two convolutional paths are then concatenated along the channel dimension, resulting in data with 29 time steps and 32 features. Finally, the data is passed through a max-pooling layer to reduce the time steps to 14, followed by residual blocks to alleviate the vanishing gradient problem and accelerate the model's convergence. By combining multiple convolutional layers, the network can extract various local features at different scales. Smaller kernels capture local temporal features with shorter intervals, while larger kernels extract features with longer intervals. This multi-scale feature extraction method effectively overcomes the limitations of a single convolutional layer's inability to fully capture features, making the network better suited for complex data and helping to alleviate the issue of overlapping fault categories in track circuits. Moreover, the

reduction in time steps from 60 to 14 achieves dimensionality reduction, which speeds up model training. As shown in Figure 2, the proposed multi-scale convolutional structure is capable of extracting short-term features at different scales, thereby improving classification accuracy.

3.2 Gated convolution and LDAM loss

Multi-scale convolution can effectively extract features of different sizes, but its effect on imbalanced data might be limited. Li et al. (2019) used multi-head self-attention and gated convolution to capture the patterns of normal requests in the network, ultimately achieving good results. To address the issue of more normal data and less fault data in real-world scenarios, we use a feature learning-based approach. We also introduce gated convolution on top of each convolution layer.

In Figure 2, A represents the output of a regular convolution, and B represents the output of the gated convolution before passing through the sigmoid function. The output C of this convolution layer is then expressed as:

$$C = A \otimes \sigma(B) \tag{1}$$

The kernel size and stride of the gated convolution are the same as those of the connected convolution. The data is processed in the same way through both the gated convolution and the regular convolution layer. The difference lies in the fact that the gated convolution is passed through a sigmoid function, which limits the output values between 0 and 1. Afterwards, the output of the gated convolution is element-wise multiplied by the result of the connected convolution layer. Thanks to the gating mechanism, the expression of features along the feature dimension becomes clearer. Specifically, the features that have a greater impact on the sample are amplified, which results in a more significant influence on the model's output. This demonstrates the information-filtering role of the gated convolution. The threshold value represents the influence of the current feature on the sample: the stronger the feature's influence, the larger the threshold.

Regarding classifier design, we also employed the large margin adaptive loss (LDAM) loss function to replace the conventional cross-entropy loss function. The LDAM loss function, proposed by Cao et al. (2019), is a method to address the issue of imbalanced data. The standard cross-entropy loss function for classification typically assigns higher weights to majority class samples. This causes the model to focus more on the majority class during optimisation, which may lead to a decision boundary biased toward the majority class. The core idea of the LDAM loss function is to adjust the class boundaries so that minority class samples receive higher weights in the loss function. It gives the minority class a larger classification boundary, shifting the actual boundary toward the majority class, thus improving the model's sensitivity to the minority class. The principle of LDAM is as follows: in model f the output for sample x is an N-dimensional vector, where N is the total number of classes, and y is the label corresponding to sample x. The distance $\delta(x, y)$ between the sample (x, y) and its decision boundary can be defined as:

$$\delta(x, y) = f(x)_y - \max_{k \neq y} f(x)_k \quad k \in \{1, \ldots, N\} \tag{2}$$

where $f(x)_y$ represents the output value of the y^{th} class in the model's output vector; $f(x)_k$ represents the output value of the k^{th} class in the output vector, where $k \in \{1, \ldots, N\}$. By introducing a class-dependent margin $\delta(x, y)$ to modify the standard cross-entropy loss, where the margin is inversely proportional to the class frequency, this adjustment helps to mitigate the imbalance between the majority and minority classes.

In the dataset S, the decision boundary δ_k for each class is defined as:

$$\delta_k = f(x)_y - \min_{i \in S} \delta(x_i, y_i) \tag{3}$$

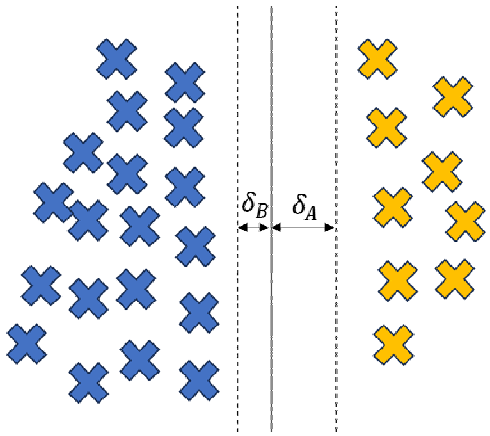
where $\delta(x_i, y_i)$ is the distance between a sample of a certain class and its decision boundary. $\min_{i \in S}\delta(x_i, y_i)$ represents the minimum distance between the decision boundary and all samples in the dataset S.

Through derivation, the optimal value of δ_k is given by:

$$\delta_k = \frac{C}{n_k^{1/4}} \quad k \in \{1, \ldots, N\} \tag{4}$$

where n_k is the number of samples in class k, and C is a hyperparameter. By introducing δ_k for each class, a global decision boundary is determined. The more samples there are for a class, the smaller δ_k becomes, thereby controlling the margin of the decision boundary.

Figure 3 The LDAM loss function biases the class boundary towards examples from the majority class (see online version for colours)



The result of the LDAM loss is converted into the output form of softmax, and the final expression is:

$$L_{LDAM}((x, y); f) = -\log \frac{e^{z_y - \delta_y}}{e^{z_y - \delta_y} + \sum_{k \neq y} e^{z_k}} \quad k \in \{1, \ldots, N\} \tag{5}$$

where δ_y represents the optimal classification margin for class y; z_y is the output value for class y in the model's output vector for sample x; and z_k is the output value for class k in the output vector for the same sample. The LDAM loss function adjusts the output $f(x)_y$ for class y by the optimal classification margin δ_y. The fewer the samples of class y, the larger δ_y becomes, and the final class boundary shifts towards the majority class, alleviating the impact of imbalanced data.

Figure 3 presents an imbalanced binary classification problem, clearly demonstrating how the LDAM loss function biases the class boundary towards the majority class. Class *B* represents the majority class and class *A* represents the minority class. The solid line represents the class boundary, which shifts towards the majority class *B*, resulting in $\delta_B < \delta_A$. This adjustment helps the model better differentiate the minority class *A*.

3.3 Transformer encoder

In 2017, Vaswani et al. proposed the transformer model to address sequence-to-sequence tasks in natural language processing. Gorishniy et al. (2021) used only the encoder part of the transformer to build their model, and similarly, we use the encoder part of the transformer to extract features. While the multi-scale convolution extracts short-term features, the transformer captures global features. Therefore, the two can complement each other to fully extract features from the data.

As shown in Figure 4, the encoder part of the transformer is capable of extracting global features from the data, compensating for the limitations of the multi-scale convolution.

The network we use consists of two stacked layers of the transformer encoder, which include multi-head self-attention layers, normalisation, residual connections, and fully connected networks. The key component is the multi-head self-attention layer, and the formula for multi-head self-attention is as follows:

$$\begin{cases} Q_i = EW_i^q & \text{(6a)} \\ K_i = EW_i^k & \text{(6b)} \quad (i = 1, 2, \ldots, I) \\ V_i = EW_i^v & \text{(6c)} \end{cases} \tag{6}$$

where W_i^q, W_i^k, W_i^v are the weight matrices for the query, key, and value, respectively. Q_i, K_i, and V_i are the corresponding generated matrices, and i represents the number of heads in the multi-head self-attention layer. E represents the input data.

$$head_i = softmax\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right)V_i \quad (i = 1, 2, \ldots, I) \tag{7}$$

where d_k is the dimension of the key vectors, and scaling by $\sqrt{d_k}$ is used to prevent gradient explosion. The softmax function is applied to normalise the dot product results, ensuring that the output is a probability distribution. $head_i$ represents the output of a single attention head, and its value indicates the importance of each time step.

$$H = Concat\left(head_1, head_2, \ldots, head_i\right)W_\Phi \quad (i = 1, 2, \ldots, I) \tag{8}$$

where W_Φ is the final weight matrix used for linear transformation, and *Concat* refers to concatenating the matrices. The outputs from multiple heads are concatenated and then fused through a linear transformation by W_Φ, ultimately forming a representation that contains global information. Through the multi-head mechanism, the model can parallelly capture dependencies from multiple perspectives in different subspaces, thereby comprehensively evaluating the importance of each time step and improving its ability to capture and express global information.

Each head has its own W^q, W^k, and W^v matrices, which are multiplied with the input matrix to generate the Q, K, and V matrices. After performing the attention computation, the output for each head is obtained. Finally, the outputs of all heads are concatenated and multiplied by the weight matrix to produce the final output.

The multi-head self-attention mechanism dynamically allocates weights based on the importance of different time steps in the input sequence, allowing the model to focus more on the key features at critical moments. After feature extraction by the multi-scale convolution layers, the data is passed through two layers of the transformer encoder, maintaining the same structure with 14 time steps and 32 features. First, the data is input in parallel to the transformer, overcoming the limitations of sequential processing in traditional RNN networks. Then, the data undergoes the multi-head self-attention mechanism, which extracts global features by enhancing the important time step features (i.e., increasing their values) while suppressing irrelevant time step features. After the multi-head self-attention layer, the data passes through residual connections and normalisation for processing, ensuring stability. Finally, the data is integrated through a fully connected network and further processed with residual connections and normalisation, providing support for subsequent classification or prediction tasks. The encoder structure of the transformer inherits the advantages of the multi-head self-attention mechanism while being more stable, allowing it to better capture the features of the data.

3.4 Integration into the overall framework

This paper proposes an effective track circuit fault diagnosis method for imbalanced data. As shown in Figure 5, the overall flow of the network illustrates the data movement and processing process. As depicted in Figure 6, the overall framework of the network provides a more intuitive display of the structural changes in the data and the roles of each component. In the flowchart, Conv1_3 and Conv2_22 represent two parallel convolutional layers, with kernel sizes of 3 and 22, respectively. The raw data consists of time-series data with 60 time steps and nine features.

Multi-scale convolution is used to extract local features of the data at different time scales, while gated convolution scales spatial features to mitigate the impact of imbalanced data. Then, transformer encoder is used to extract global features from the data. After global average pooling, the number of time steps is reduced to 1, and the features are finally input into a fully connected layer for classification.

4 Experimental validation and performance analysis

4.1 Dataset description and imbalance handling

To fully verify the effectiveness and feasibility of the algorithm, comprehensive data collection was conducted. Track circuit simulations were conducted, and as shown in Figure 7, the established simulation system can accurately generate various fault data. By artificially modifying the simulation system, experiments were performed on the normal and various fault states of the ZPW-2000A track circuit, and 1,600 data samples were collected. To verify the rationality of the developed insulation-free track circuit transmission model, a section of an actual track was selected for model validation. The actual parameters of the track circuit are shown in Table 1, and the model parameters were set according to the actual environment. The final results are compared in Table 2, demonstrating the accuracy of the ZPW-2000A track circuit simulation model. From Table 2, it can be seen that the maximum relative error between our simulated values and the actual values is 9.69%, indicating that the transmission model is reasonable. Therefore, the data obtained from this model can be used as input for the subsequent fault classification model.

Faults in track circuits occur rapidly, typically completing within one minute, after which the voltage stabilises within a specific range. Therefore, real-time fault diagnosis with minimal time intervals is essential. As a result, voltage data from nine points were collected at a frequency of one sample per second for one minute, as shown in Table 3, the various voltage monitoring points are presented. This way, each data sample consists of 60 time steps, with 9 voltage features at each time step. The small interval between samples ensures detailed feature capture and avoids the loss of information that could occur with larger intervals.

Additionally, to reflect the imbalanced nature of the data, 500 normal data samples and 100 samples of each fault type were collected, as shown in Table 4. This ensured that the network was exposed to more normal data during training, which provided the conditions for handling imbalanced data. Experimental data that is either too sparse or too excessive can have negative effects. Too few data samples may result in insufficient network learning, leading to inaccurate fault diagnosis, while too many samples could lead to excessive training time. Therefore, a balanced dataset of 1,600 samples was chosen for the experiment. Overall, the dataset consists of 1,600 time series samples, each with 60 time steps and nine voltage monitoring features per step. It includes 12 categories: one normal state (500 samples) and 11 fault states (100 samples per category). The normal data accounts for 31.25% of the total, while each fault category accounts for 6.25%, presenting a typical imbalanced distribution that reflects the sparsity of faults in real-world scenarios. Data was sampled once per second over one minute, ensuring the capture of short-term variations.

Table 1 Track circuit parameters in a real-world environment

Parameter name	*Parameter value*
Carrier frequency	1,700 Hz
Main track circuit length	1,201 m
Number of compensating capacitors	13
Capacitance value of compensating capacitor	55 μF
Output power level	Third level
Track bed resistance	1.0 Ω•km
Rail impedance	15.1 Ω•km
Receiver resistance	400 Ω

Table 2 The results of the comparison between the simulation value and the actual value

Feature parameters	*Actual value*	*Simulated value*	*Absolute error*	*Relative error*
A	135.2	135.2763	0.0763	0.06%
B	86.6	86.7956	0.1956	0.23%
C	1.95	2.054841	0.104841	5.38%
D	1.95	2.051766	0.101766	5.22%
E	1.12	1.109555	–0.01044	–0.93%
F	1.12	1.109515	–0.01048	–0.94%
G	10.08	9.9869	–0.0931	–0.92%
H	0.871	0.872358	0.001358	0.16%
I	0.483	0.436176	–0.04682	–9.69%

Table 3 Voltage monitoring points set up on the track circuit

Symbol	*Feature parameters*
A	Transmitting output voltage
B	Cable-side voltage of transmitting end cable simulation network
C	Output voltage of transmitting end matching transformer
D	Rail surface voltage at transmitting end
E	Rail surface voltage at receiving end
F	Input voltage of receiving end matching transformer
G	Cable-side voltage of receiving end cable simulation network
H	Track input voltage
I	Track output voltage

Figure 4 The transformer encoder structure used for extracting global features (see online version for colours)

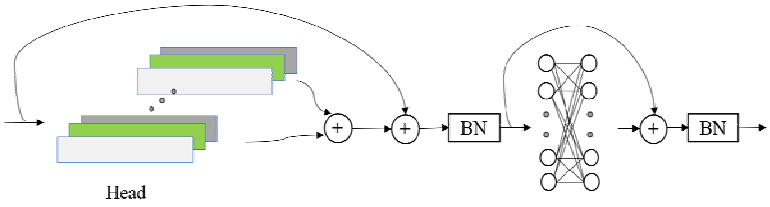


Figure 5 The overall network flow reflecting data movement and processing

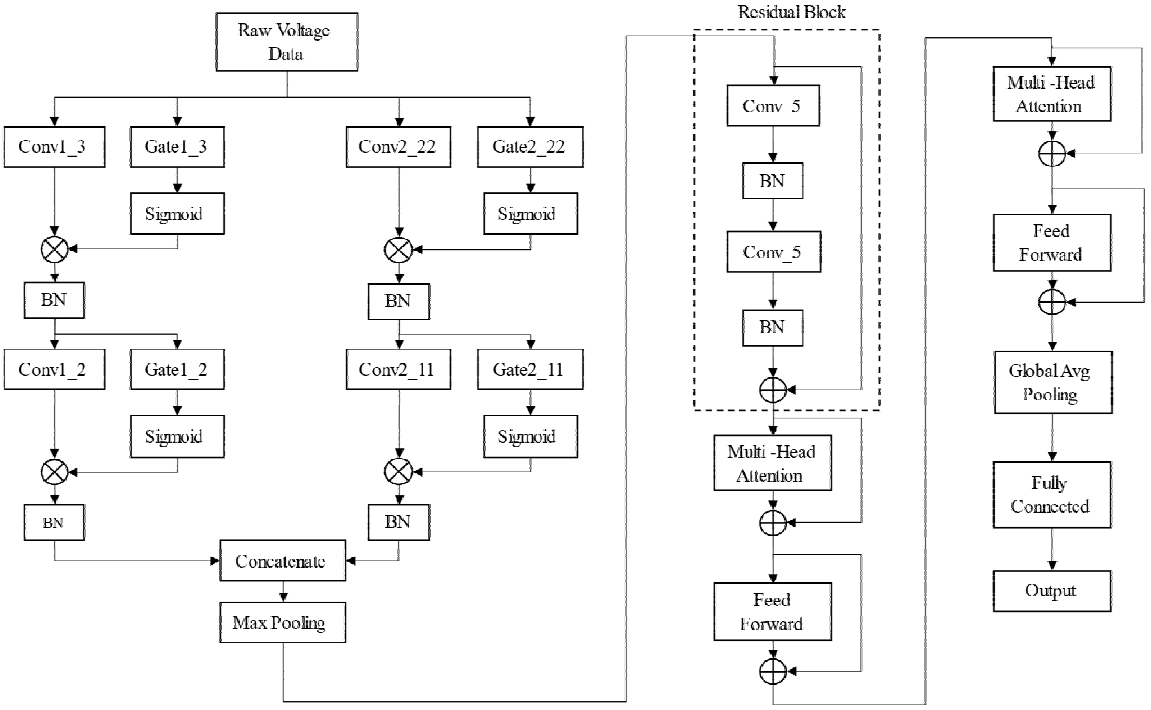


Figure 6 The overall network framework demonstrating data structure and component roles (see online version for colours)

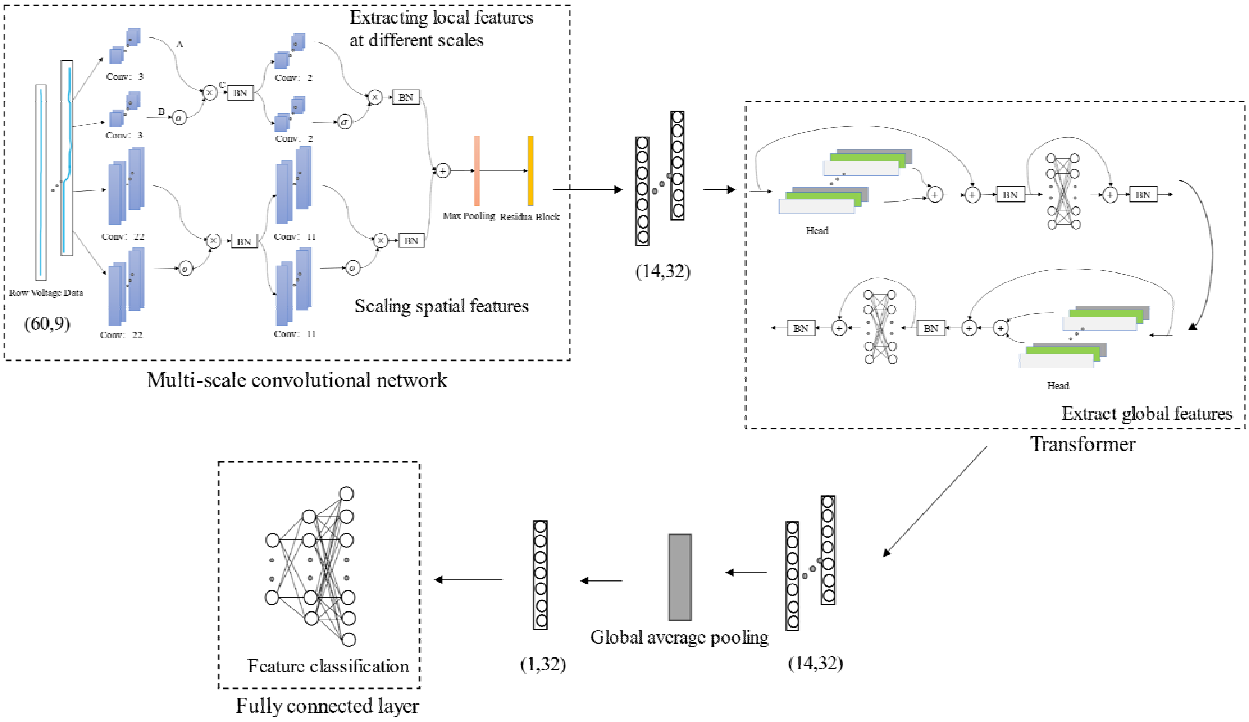
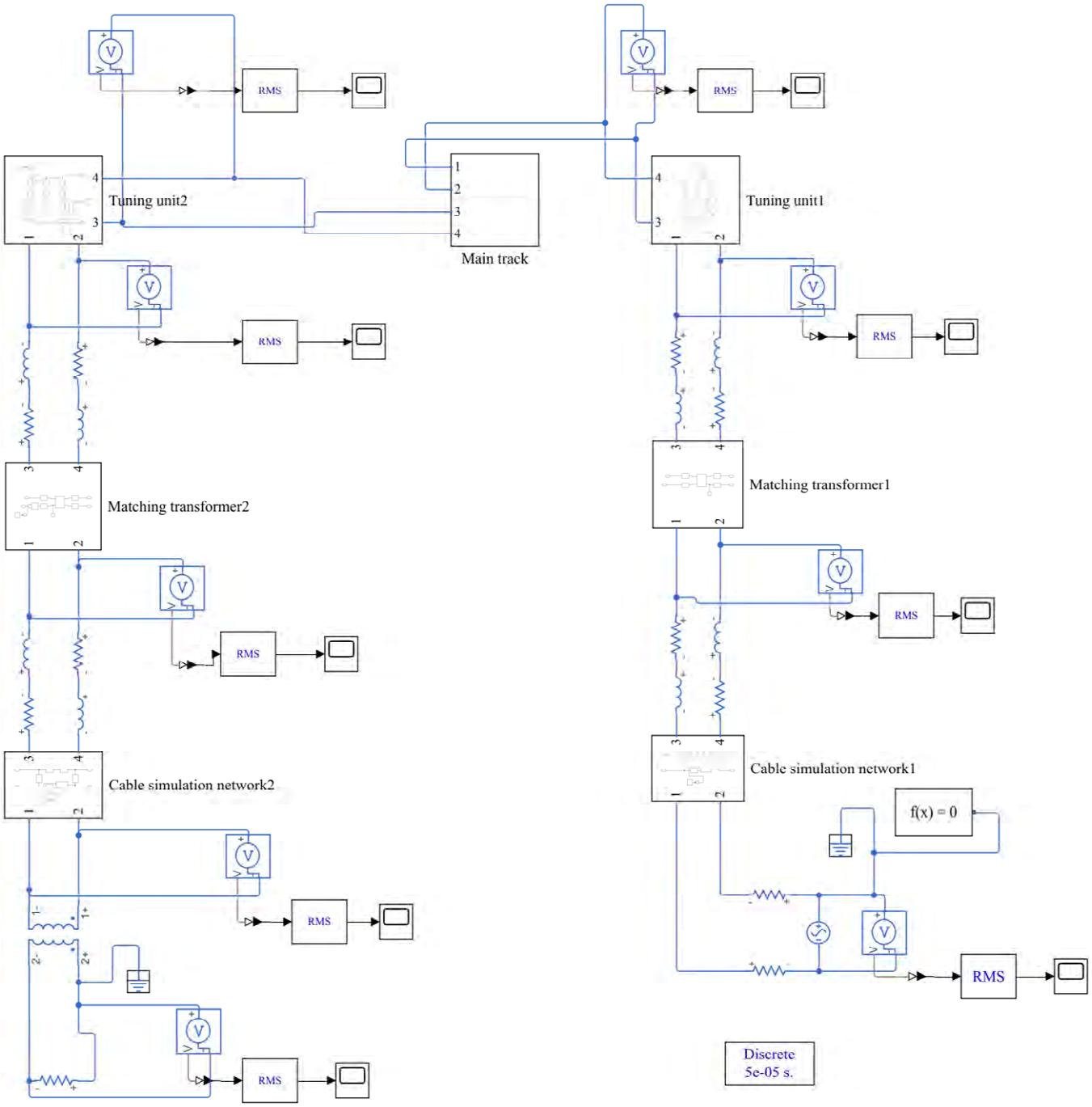


Figure 7 Simulated track circuit system used for data collection (see online version for colours)



4.2 Experimental setup and metrics

The experiment begins with data pre-processing. Since the data is collected through simulation, it contains almost no outliers, so pre-processing only involves a simple check for missing values, while data normalisation is handled within the model. In practical rail circuit fault diagnosis, missing and outlier data must be addressed – missing values can be handled by deleting or imputing samples, and outliers can be removed or corrected. The dataset is then split into training and test sets in an 8:2 ratio, while maintaining the same 8:2 proportion for each class. This approach reduces the risk of overfitting on specific categories, enhances overall prediction accuracy, and ensures the model learns the characteristics of different classes, thereby improving its generalisation ability.

During training, the StepLR strategy is used to gradually decrease the learning rate. A larger learning rate at the beginning prevents the model from converging too quickly to a local optimum in the early stages of training, allowing the exploration of more parameter space. In the later stages of training, when the model is close to the optimal solution, a smaller learning rate helps reduce oscillation and allows for finer updates. StepLR reduces the learning rate by a factor of gamma (lr = lr * gamma) when the training reaches the step_size. In this experiment, the learning rate (lr) is set

to 0.001, the step_size is 15, gamma is 0.7, and the total number of epochs is 120.

Table 4 The collected data provided under imbalanced conditions

Fault type	*Label*	*Number of samples*
Normal	0	500
Transmitter fault	1	100
Transmitting end cable simulation network fault	2	100
Transmitting end SPT cable fault	3	100
Transmitting end matching transformer fault	4	100
Transmitting end tuning unit fault	5	100
Main track break	6	100
Receiving end tuning unit fault	7	100
Receiving end matching transformer fault	8	100
Receiving end SPT cable fault	9	100
Receiving end cable simulation network fault	10	100
Receiver fault	11	100

The Adam optimiser is used for training, with network parameters determined through empirical methods and multiple experiments. All experiments are conducted using the PyTorch framework with GPU support. Hyperparameter tuning is essential in the process of training the model. The initial number of epochs was set to 100, but as the loss continued to decrease at the end of training, the epochs were increased to 120 to ensure the loss reached its minimum. The learning rate was generally set to 0.01, 0.001, or 0.0001, and experiments showed that a learning rate of 0.001 led to a fast initial decline in loss, followed by stable performance with minimal oscillation. The StepLR strategy was employed, with the step size typically chosen between 10 and 30 and gamma between 0.5 and 0.9. After multiple experiments, the optimal step size and gamma were finally determined. The network parameters are shown in Table 5, presenting the specific parameter settings for each structure in the network. The convolutional layers consist of two parallel convolutions: one with a smaller kernel and the other with a larger kernel. Before passing through the pooling layer, the two convolutions are merged along the channel dimension. Finally, the fully connected layer uses Dropout, with a 20% probability that each neuron will be turned off during each training iteration. This prevents the model from overly relying on any single neuron, reducing overfitting and enhancing the model's generalisation ability.

Regarding the selection of convolution kernel sizes, for the first layer with small kernels, we experimented with sizes of 3, 5, and 7 based on empirical knowledge, while keeping the large kernel unchanged. The results showed that kernel sizes of 3 and 5 produced nearly identical performance, whereas a kernel size of 7 led to poorer and more unstable results. Given that the collected data exhibits rapid short-term changes, we chose the smaller kernel size of 3 to better capture these short-term variations. Even when the data changes more rapidly, the three-sized kernel can more effectively extract short-term features. Additionally, the three-sized kernel involves the least number of parameters, conserving computational resources.

Table 5 The specific parameters of each structure in the network

Layer type	*Key parameters*	*Output*
Input layer	——	None (60, 9)
Convolutional layer	Kernel size: 3, input channels: 9, output channels: 12, stride: 1	None (58, 12)
	Kernel size: 22, input channels: 9, output channels: 12, stride: 1	None (39, 12)
BN layer	——	None (39, 12)
Convolutional layer	Kernel size: 2, input channels: 12, output channels: 16, stride: 2	None (29, 16)
	Kernel size: 11, input channels: 12, output channels: 16, stride: 1	None (29, 16)
BN layer	——	None (29, 16)
Max pooling layer	Kernel size: 2, stride: 2	None (14, 32)
Residual block	Kernel size: 5, stride: 1, padding: 2	None (14, 32)
Transformer encoder	Number of features: 32, number of heads: 4, number of encoder layers: 2	None (14, 32)
Global average pooling	——	None (32)
Fully connected layer	Neurons: 32	None (32)
Fully connected layer	Neurons: 22, dropout: 0.2	None (22)
Output layer	Neurons: 12	None (12)

For the first layer's large convolution kernel, we experimented with sizes of 11, 22, and 33. The results indicated that a kernel size of 22 yielded the best model performance. The 33-sized kernel had a larger number of parameters and poorer training outcomes, while the 11-sized kernel failed to capture longer local features effectively. Therefore, we selected the 22-sized kernel for better feature extraction. The second layer further reduces the dimension of the data while keeping the time steps of the data on both sides the same to prevent feature loss caused by concatenation. The size can be appropriately selected. When choosing the number of transformer encoder layers, we conducted experiments with 2, 4, and 6 layers while keeping the learning rate constant. Results showed that increasing the number of transformer layers led to lower test accuracy. Even after reducing the learning rate appropriately with

deeper layers, test accuracy continued to decline. This suggests that, given the current dataset of 1,600 samples, a two-layer transformer encoder is more suitable. Increasing the number of layers introduces a greater risk of gradient vanishing and overfitting while extending training and inference time. The number of layers should align with task complexity rather than being increased indiscriminately. In practical rail circuit fault diagnosis with larger datasets, the number of transformer layers can be adjusted in proportion to the data volume to fully capture global features.

We use accuracy, precision, recall, and F1-score to evaluate the performance of the model.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{9}$$

True positive (TP) refers to the number of correctly predicted positive samples, true negative (TN) refers to the number of correctly predicted negative samples, false positive (FP) refers to the number of samples incorrectly predicted as positive, and false negative (FN) refers to the number of samples incorrectly predicted as negative. Accuracy represents the proportion of correctly predicted samples out of the total samples.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

Precision represents the proportion of true positive samples among those predicted as positive.

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

The recall represents the proportion of true positive samples that are correctly predicted.

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{12}$$

The F1-score is the harmonic mean of precision and recall, which takes both into account. Accuracy reflects the overall proportion of correct predictions made by the model. Precision measures the reliability of the model's fault predictions – for example, if the model identifies ten instances as faults and 8 of them are actual faults, the precision is 80%, emphasising the reduction of false alarms. Recall, on the other hand, focuses on the model's ability to detect actual faults – if there are 10 true faults and the model identifies 7, the recall is 70%, prioritising the avoidance of missed detections. There is often a trade-off between precision and recall (higher precision may lead to lower recall and vice versa). Therefore, the F1-score is introduced as the harmonic mean of precision and recall, providing a balanced evaluation of the model's overall performance.

For the imbalanced dataset we are using, relying solely on accuracy is insufficient because if a category is rare, but the model incorrectly predicts all samples, the accuracy can still be high. Therefore, we use multiple metrics for evaluation.

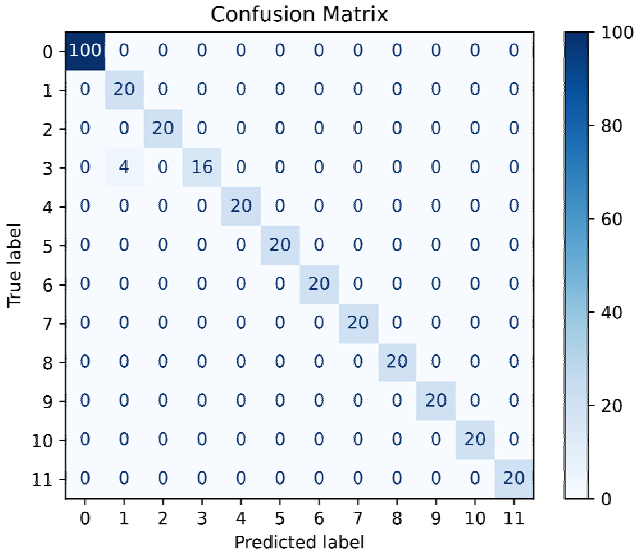
The final test results are shown in Table 6, demonstrating the effectiveness of the model in handling imbalanced data. The accuracy of the test set reached 98.75%, and the F1-score is very close to the accuracy value. This indicates that the model has a good classification performance, and both the minority and majority classes are mostly classified correctly, with no cases where only the majority class is correctly classified.

Table 6 Test results of the test set

Method	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1-score (%)*
Ours	98.75	98.96	98.75	98.74

The confusion matrix of the test set is shown in Figure 8, providing an intuitive display of the classification results, which demonstrate excellent performance. The change in loss during training is shown in Figure 9, reflecting the effectiveness of the learning rate decay. In the confusion matrix, it can be seen that four samples of class 3 were incorrectly classified as class 1 by the model. This could be due to the similarity between the fault data of the transmitter and the SPT cable at the transmitter end, leading to the model's misclassification. However, for other classes, the model performs well, which demonstrates its effectiveness on imbalanced data.

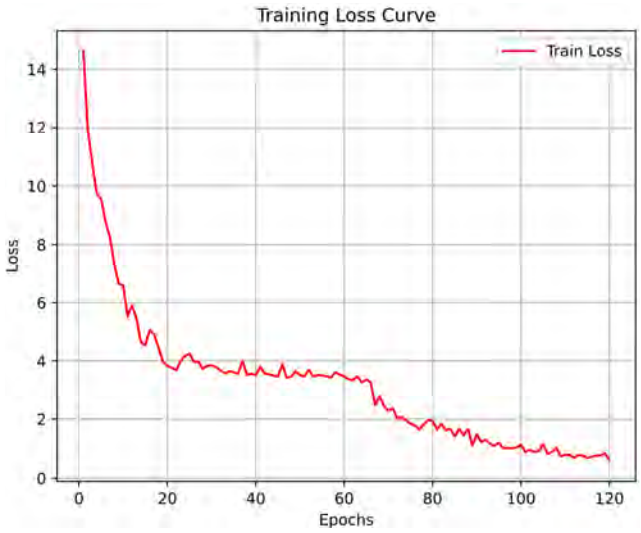
Figure 8 The confusion matrix of the test set (see online version for colours)



During training, the loss decreases rapidly in the first 20 epochs, stabilises until epoch 60, and then continues to decline before stabilising again at epoch 100. This behaviour is due to the larger learning rate used initially, causing more significant fluctuations in the loss during the early stages of training. From epoch 20 to epoch 60, the learning rate might not have decreased enough, leading to oscillations at the point of minimum loss, where the loss value remains nearly unchanged. After epoch 60, when the learning rate was reduced again, it alleviated the oscillations, and the loss gradually approached its minimum

value. The StepLR learning rate decay strategy helped to reduce the oscillations in the later stages of training, while also keeping the loss stable, thus confirming the effectiveness of the learning rate decay.

Figure 9 The loss values during training (see online version for colours)



Even after data pre-processing, real-world noise interference remains inevitable. To evaluate the model's robustness against noise, we introduced 5%, 10%, 20%, and 40% noise disturbances into the training set. The added noise included Gaussian white noise, sensor failure noise, and label noise. Gaussian white noise is a type of random noise with a normal distribution and uniform power spectral density, commonly used to simulate sensor measurement errors or environmental interference. The intensity of the Gaussian white noise was set to 10% of the standard deviation of the original data, fluctuating around a zero mean. Sensor failure noise refers to the complete malfunction of a monitoring channel, leading to continuous abnormal readings – in this case, setting one voltage monitoring point to zero throughout a sample. Label noise represents the incorrect assignment of sample labels. The experimental results, shown in Table 7, demonstrate the strong noise resistance of our proposed model.

Table 7 The experiment reflecting the model's noise resistance at different noise proportions in the training set

Noise (%)	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1-score (%)*
5	98.75	98.96	98.75	98.74
10	95.94	97.54	95.94	95.46
20	93.44	89.39	93.44	91.01
40	82.19	80.46	82.19	77.89

When noise accounts for 5% of the training set, the test metrics remain identical to those observed in the noise-free condition, indicating that the model's performance is nearly unaffected by a small proportion of noise. As the noise proportion increases, the test metrics begin to decline, but the decrease remains relatively small. Even under the extreme condition of 40% noise, the metrics still approach 80%. This demonstrates the strong noise resistance of our model, highlighting the robustness of the multi-scale convolution and transformer encoder in handling noisy data.

Overall, compared with models such as CNN-LSTM and CNN-MHSA, our model has a distinct advantage in handling imbalanced data and its inference speed fully meets the requirements of real-time track circuit fault diagnosis. These aspects demonstrate the feasibility of our model.

4.3 Ablation study

To verify the performance of each module in the network, we divided the network into three modules: A, B, and C. Module A is the multi-scale convolution module, module B is the transformer encoder module, and module C is the gated convolution and LDAM module. When module C is removed, the loss function is replaced by cross-entropy. Ablation experiments were conducted by removing each module from the network. The experimental results are shown in Table 8 and Figure 10, providing an intuitive display of the performance and classification results of each module.

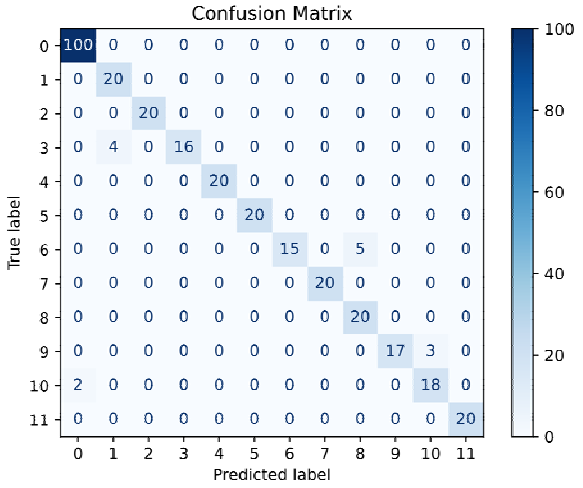
Table 8 Comparison of test set results reflecting the performance of the modules

Module	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1-score (%)*
A	95.62	95.81	95.63	95.52
A + B	96.88	96.91	96.88	96.87
A + C	97.81	98.14	97.81	97.79
A + B + C	98.75	98.96	98.75	98.74

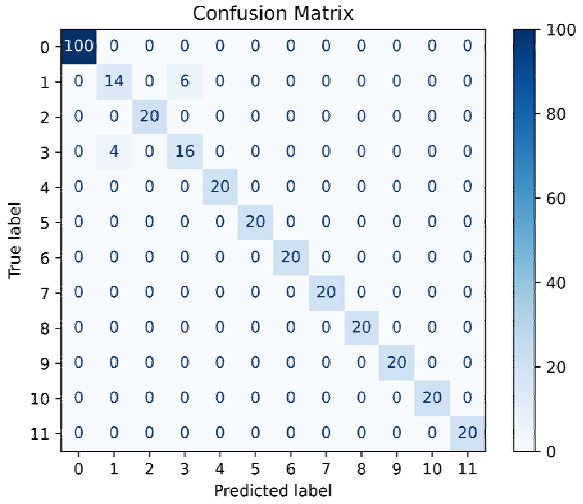
From Table 7, it can be seen that the accuracy of A + B + C is the highest, followed by A + C, then A + B, and A alone has the lowest accuracy. For A + B, the absence of the module designed to handle imbalanced data resulted in 10 samples from classes 1 and 3 being misclassified. This reflects the information filtering effect of the gated convolution and the boundary-shifting effect of the LDAM loss function, highlighting the effectiveness of these modules for imbalanced data. A+C, which lacks the transformer encoder module, resulted in 4 samples from class 3 being misclassified as class 1, and 3 samples from class 9 being misclassified as class 10. Compared to A + B + C, different types of samples were misclassified. Since the time variation for each type of fault is different, this reflects the ability of the transformer encoder to capture global features. A, due to the absence of both modules for handling imbalanced data and extracting long-term interval features, misclassified a total of 14 samples across four categories. All metrics were the lowest for this configuration.

Figure 10 Confusion matrix of each module on the test set (see online version for colours)

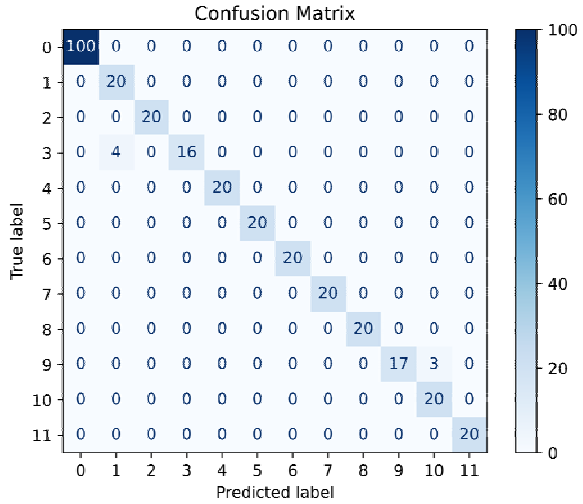
A: Multi-scale convolution



A + B: Multi-scale convolution + transformer encoder



A + C: multi-scale convolution + gated convolution and LDAM



4.4 Comparison with state-of-the-art techniques

To further validate the effectiveness of the method proposed in this paper, we compared it with other algorithms. The same dataset was used in the experiments to ensure fairness. The experimental results, as shown in Table 9, demonstrate the superiority of our proposed method.

Table 9 Test results of the test set in the original dataset

Method	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1-score (%)*
CNN	75	70	75	70.83
LSTM	81.25	68	81	74
CNN-LSTM	94.69	97.13	94.69	93.52
CNN-MHSA	95.94	97.54	95.94	95.46
Ours	98.75	98.96	98.75	98.74

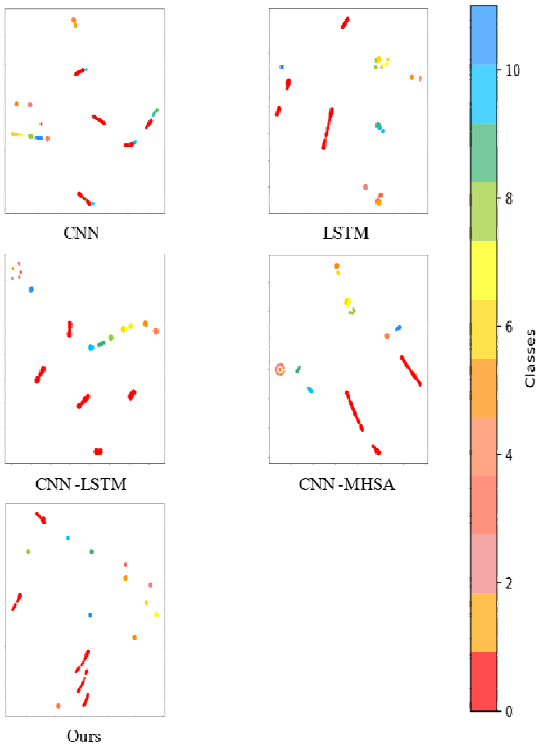
As can be seen from the table, our method outperforms all others in every metric. The lower F1-scores compared to accuracy for the other four algorithms highlight their difficulty in handling imbalanced data, as they tend to perform well on the majority class but poorly on the minority class. Among them, CNN has the lowest performance across all metrics. This is due to CNN's inferior feature extraction for sequential data compared to LSTM, which is capable of capturing long-range dependencies in time-series data, while CNN cannot. The combination of CNN and LSTM (CNN-LSTM) shows a significant improvement in all metrics compared to the individual methods. On the other hand, CNN-MHSA, which replaces LSTM with a multi-head self-attention mechanism, achieves even higher scores than CNN-LSTM, suggesting that multi-head self-attention is superior for long-term feature extraction.

To better demonstrate the advantage of our method under imbalanced data conditions, we increased the number of normal data samples from 500 to 1,000 and recalculated the metrics for each method. The results, as shown in Table 10, indicate that there is little change in the performance metrics of our method after increasing the majority class, demonstrating its effectiveness in handling imbalanced data. Additionally, as shown in Figure 11, the classification results are intuitively presented using a t-SNE plot.

Table 10 Test results of the test set in the dataset after adding normal data

Method	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1-score (%)*
CNN	0.71	66.4	70.71	65.44
LSTM	76.19	64	76	68
CNN-LSTM	86.43	82.19	86.43	83.03
CNN-MHSA	90.48	87.3	90.48	88.09
Ours	99.05	99.21	99.05	99.04

Figure 11 t-SNE plots of each algorithm reflecting the classification performance (see online version for colours)



As shown in Table 10, after increasing the number of normal data samples, the performance of the other four methods decreased, while our method only showed a slight change, which can be attributed to the randomness in training. From the t-SNE plots, it is evident that the other four methods perform well in classifying the majority class (class 0), but their performance on the minority classes is poor, with significant overlap and close distances between different classes. This is due to the network being exposed to a higher proportion of normal samples during training, leading to better learning of the majority class and poorer learning of the minority classes. In contrast, our method shows almost no overlap between the classes, with greater distances between them, highlighting the effectiveness of our approach in handling imbalanced data.

To compare the model's complexity and evaluate its practicality in real-time fault detection, we used three metrics: floating point operations (FLOPs), parameters (Params), and inference time. FLOPs represent the number of floating-point operations required for a single forward pass, used to measure the model's computational complexity; Params refers to the total number of trainable parameters in the model, including weights for fully connected layers and convolution kernels; inference time is the time it takes for the model to process a single input sample and generate a prediction. We compared three models, as shown in Table 11, which demonstrates the practicality of our model in real-time fault detection.

Table 11 Comparison of model complexity and inference time reflecting practicality

Method	*FLOPs*	*Params*	*Inference time*
CNN-LSTM	48.450 K	20.2 K	0.6405 ms
CNN-MHSA	26.760 K	4.966 K	0.7959 ms
Ours	8,431 K	297.058 K	3.7473 ms

Our model has significantly higher values for FLOPs and Params compared to CNN-LSTM and CNN-MHSA, indicating that both the training and testing times are the longest. This is due to the more complex structure of the entire model. However, in terms of accuracy, precision, and other metrics, our model performs the best. In practical rail circuit fault diagnosis, the improved performance reduces the probability of false positives and false negatives, enhancing safety, with a cost far lower than the losses caused by a single accident. Additionally, the inference time of our model is significantly less than the one-second data interval, ensuring the real-time capability and availability of fault diagnosis.

4.5 Discussion on misclassifications and error analysis

In the experiments, the misclassification results can be categorised into three types: the first type is the misclassification of minority class samples, the second type is the misclassification of different class samples, and the third type is the misclassification of similar samples. In the ablation experiments, the absence of gated convolution and LDAM loss function resulted in poor handling of imbalanced data, causing ten misclassified samples from two minority classes – significantly more than the complete model. For the model without the transformer encoder, the ability to extract long-term features was missing, resulting in a higher number of misclassified samples from different categories compared to the complete model. The results of the ablation experiments highlight the critical role of gated convolution, LDAM loss function, and transformer encoder in reducing misclassifications.

Regarding the experimental errors, they were mainly due to the similarity between the features of the two classes. The impact of data imbalance was minimal, as increasing the majority class samples resulted in only small differences in the metrics between the two tests. In contrast, the metrics of other algorithms showed significant differences, demonstrating the effectiveness of the proposed model in handling imbalanced data.

5 Practical implications

In practice, track circuit fault diagnosis often relies on manual inspection of voltage monitoring points, which is time-consuming and prone to errors. With our track circuit fault diagnosis model, the system can quickly and accurately locate the fault based on a large amount of data, significantly shortening the fault diagnosis time and

allowing people to quickly identify the fault location. This high-accuracy automated diagnosis can help avoid potential safety hazards, such as when a train continues to operate in a known fault area due to delayed emergency handling, increasing the risk of collision or derailment with the fault point. By reducing the fault detection time, train delays caused by faults can be minimised, thereby reducing losses and improving the overall operational efficiency of the railway system.

The method proposed in this paper can also be extended to other fields, such as industrial fault detection, medical diagnosis, network security, and intrusion detection. These fields involve imbalanced data and time dependencies, making them suitable for our model. The model structure can be flexibly adjusted according to the characteristics of specific data to adapt to different domains.

The model training phase requires significant computational resources, typically relying on high-performance GPUs, and takes a few minutes to train on our data. After the training is completed, the inference phase does not require substantial computational resources and can quickly produce results, making it easy to deploy the model for rapid implementation across various platforms.

6 Conclusions and future work

This paper proposes a network structure that combines multi-scale convolution with gated convolution, followed by two layers of transformer encoders and a fully connected network. The loss function used is LDAM. Multi-scale convolution extracts short-term features at varying time scales using different kernel sizes, while gated convolution selectively weights features for information filtering. The LDAM loss function adjusts the classification boundary to favor the majority class, effectively addressing class imbalance. The multi-head self-attention layer in the transformer encoder captures global data features, compensating for the inability of multi-scale convolution to extract long-term data features. The subsequent normalisation, residual blocks, and fully connected layers integrate the data and maintain stability, with the final fully connected layer serving as the classifier.

In the experimental section, a simulation system for the ZPW-2000A track circuit was established to collect raw voltage data. Ablation experiments were conducted to validate the function of each module, and comparisons with other algorithms demonstrated the superior ability of our method in diagnosing ZPW-2000A track circuit faults under imbalanced data conditions. At the same time, the model demonstrates good practicality, as it can enhance railway safety and efficiency, and is also applicable to fault diagnosis in other fields.

However, there are some limitations. The data used in this paper were simulated through a fault model, and compared to real-world fault data, the fault types are limited. The model is also relatively complex and requires high hardware specifications. In future work, we will attempt to combine real fault data with simulation data to make the model's predictions more realistic. We will also consider reducing the number of layers in the multi-scale convolution and transformer encoder and improving the structure of the residual blocks to reduce the model's complexity and enhance its operational efficiency.

Declarations

The source code part of this study was provided by artificial intelligence and modified manually for use, aiming to save time on writing simple code and improve research efficiency. Additionally, during the translation process into English, artificial intelligence was used for proofreading to enhance the fluency and academic quality of the language.

All authors declare that they have no conflict of interest.

References

Arafa, A., El-Fishawy, N., Badawy, M. and Radad, M. (2022) 'RN-SMOTE: reduced noise SMOTE based on DBSCAN for enhancing imbalanced data classification', *Journal of King Saud University-Computer and Information Sciences*, Vol. 34, No. 8, pp.5059–5074, https://doi.org/10.1016/j.jksuci.2022.06.005.

Bahdanau, D., Cho, K. and Bengio, Y. (2014) *Neural Machine Translation by Jointly Learning to Align and Translate*, arXiv, https://doi.org/10.48550/arXiv.1409.0473.

Cao, K., Wei, C., Gaidon, A., Arechiga, N. and Ma, T. (2019) 'Learning imbalanced datasets with label-distribution-aware margin loss', *Advances in Neural Information Processing Systems*, Vol. 32, https://doi.org/10.48550/arXiv.1906.07413.

De Bruin, T., Verbert, K. and Babuška, R. (2016) 'Railway track circuit fault diagnosis using recurrent neural networks', *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 28, No. 3, pp.523–533, https://doi.org/10.1109/TNNLS.2016.2551940.

Gorishniy, Y., Rubachev, I., Khrulkov, V. and Babenko, A. (2021) 'Revisiting deep learning models for tabular data', *Advances in Neural Information Processing Systems*, Vol. 34, pp.18932–18943, https://arxiv.org/abs/2106.11959.

Hammad, M., Alkinani, M.H., Gupta, B.B. and Abd El-Latif, A.A. (2022) 'Myocardial infarction detection based on deep neural network on imbalanced data', *Multimedia Systems*, pp.1–13, https://doi.org/10.1007/s00530-020-00728-8.

Heinrich, K., Zschech, P., Janiesch, C. and Bonin, M. (2021) 'Process data properties matter: introducing gated convolutional neural networks (GCNN) and key-value-predict attention networks (KVP) for next event prediction with deep learning', *Decision Support Systems*, Vol. 143, p.113494, https://doi.org/10.1016/j.dss.2021.113494.

Hochreiter, S. and Schmidhuber, J. (1997) 'Long short-term memory', *Neural Computation*, Vol. 9, No. 8, pp.1735–1780, MIT-Press, https://doi.org/10.1162/neco.1997.9.8.1735.

Hu, B., Tang, J., Wu, J. and Qing, J. (2022) 'An attention EfficientNet-based strategy for bearing fault diagnosis under strong noise', *Sensors*, Vol. 22, No. 17, p.6570, https://doi.org/10.3390/s22176570.

Jin, Y., Qin, C., Zhang, Z., Tao, J. and Liu, C. (2022) 'A multi-scale convolutional neural network for bearing compound fault diagnosis under various noise conditions', *Science China Technological Sciences*, Vol. 65, No. 11, pp.2551–2563, https://doi.org/10.1007/s11431-022-2109-4.

Ke, T., Zhang, Y., Hu, Q. and Zhang, C. (2024) 'An effective CNN-MHSA method for the fault diagnosis of ZPW-2000A track circuit', *Transactions of the Institute of Measurement and Control*, https://doi.org/10.1177/01423312241273855.

LeCun, Y., Bottou, L., Bengio, Y. and Haffner. P. (1998) 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE*, Vol. 86, No. 11, pp.2278–2324, https://doi.org/10.1109/5.726791.

Li, J., Fu, Y., Xu, J., Ren, C., Xiang, X. and Guo, J. (2019) 'Web application attack detection based on attention and gated convolution networks', *IEEE Access*, Vol. 8, pp.20717–20724, https://doi.org/10.1109/ACCESS.2019.2955674.

Liang, X., Jiang, A., Li, T., Xue, Y. and Wang, G. (2020) 'LR-SMOTE – an improved unbalanced data set oversampling based on K-means and SVM', *Knowledge-Based Systems*, Vol. 196, p.105845, https://doi.org/10.1016/j.knosys.2020.105845.

Liu, X., Wang, X. and Han, G. (2021) 'Adaptive fault diagnosis system for railway track circuits', *International Conference on Electrical and Information Technologies for Rail Transportation*, pp.491–498, https://doi.org/10.1007/978-981-16-9913-955.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) 'Learning representations by back-propagating errors', *Nature*, Vol. 323, No. 6088, pp.533–536, https://doi.org/10.1038/323533a0.

Shao, Z., Li, W., Xiang, H., Yang, S. and Weng, Z. (2024) 'Fault diagnosis method and application based on multi-scale neural network and data enhancement for strong noise', *Journal of Vibration Engineering & Technologies*, Vol. 12, No. 1, pp.295–308, https://doi.org/10.1007/s42417-022-00844-x.

Sun, S., Wang, T. and Chu, F. (2023) 'A multi-learner neural network approach to wind turbine fault diagnosis with imbalanced data', *Renewable Energy*, Vol. 208, pp.420–430, https://doi.org/10.1016/j.renene.2023.03.097.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. and Gomez, A.N. (2017) 'Attention is all you need', *Advances in Neural Information Processing Systems*, https://arxiv.org/abs/1706.03762.

Yang, J., Gao, T. and Jiang, S. (2023) 'A dual-input fault diagnosis model based on SE-MSCNN for analog circuits', *Applied Intelligence*, Vol. 53, No. 6, pp.7154–7168, https://doi.org/10.1007/s10489-022-03665-3.

Zhang, T., Chen, J., Li, F., Zhang, K., Lv, H., He, S. and Xu, E. (2022) 'Intelligent fault diagnosis of machines with small & imbalanced data: a state-of-the-art review and possible extensions', *ISA Transactions*, Vol. 119, pp.152–171, https://doi.org/10.1016/j.isatra.2021.02.042.

Zhao, B., Zhang, X., Li, H. and Yang, Z. (2020) 'Intelligent fault diagnosis of rolling bearings based on normalized CNN considering data imbalance and variable working conditions', *Knowledge-Based Systems*, Vol. 199, p.105971, https://doi.org/10.1016/j.knosys.2020.105971.

Zheng, Z., Dai, S. and Xie, X. (2020) 'Research on fault detection for ZPW-2000A jointless track circuit based on deep belief network optimized by improved particle swarm optimization algorithm', *IEEE Access*, Vol. 8, No. pp.175981–175997, https://doi.org/10.1109/ACCESS.2020.3025628.

Zhou, F., Yang, S., Fujita, H., Chen, D. and Wen, C. (2020) 'Deep learning fault diagnosis method based on global optimization GAN for unbalanced data', *Knowledge-Based Systems*, Vol. 187, p.104837, https://doi.org/10.1016/j.knosys.2019.07.008.