

Research Article

A Novel Hybrid Deep Learning Model for Complex Systems: A Case of Train Delay Prediction

Dawei Wang ¹, Jingwei Guo ², and Chunyang Zhang ³

¹School of Electronics and Control Engineering, Chang'an University, Xi'an 710064, China

²Faculty of Business, City University of Macau, Macau S.A.R. 999078, China

³Luoyang Vocational College of Culture and Tourism, Luoyang 471000, China

Correspondence should be addressed to Jingwei Guo; jwguo@cityu.edu.mo

Received 2 October 2023; Revised 26 March 2024; Accepted 26 April 2024; Published 7 May 2024

Academic Editor: Mosbeh Kaloop

Copyright © 2024 Dawei Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Predicting the status of train delays, a complex and dynamic problem, is crucial for railway enterprises and passengers. This paper proposes a novel hybrid deep learning model composed of convolutional neural networks (CNN) and temporal convolutional networks (TCN), named the CNN + TCN model, for predicting train delays in railway systems. First, we construct 3D data containing the spatiotemporal characteristics of real-world train data. Then, the CNN + TCN model employs a 3D CNN component, which is fed into the constructed 3D data to mine the spatiotemporal characteristics, and a TCN component that captures the temporal characteristics in railway operation data. Furthermore, the characteristic variables corresponding to the two components are selected. Finally, the model is evaluated by leveraging data from two railway lines in the United Kingdom. Numerical results show that the CNN + TCN model has greater accuracy and convergence performance in train delay prediction.

1. Introduction

Complex systems are distinguished by their complexity, uncertainty, relevance, and openness [1, 2]. These characteristics make it difficult to predict the state of complex systems, such as weather forecasts and energy consumption predictions. With the development of artificial intelligence, the study of complex systems has entered the data-driven era [3]. Recently, the application of deep learning for the state prediction of complex systems had some achievements. For instance, 1D convolutional neural networks (CNN) have been leveraged to predict the consumption of building energy [4]. A predictive maintenance model of complex systems based on recurrent neural networks (RNN) has been designed [5]. A new neural network has been designed to predict train delays on Iranian railways [6]. However, the performance of these state prediction models for complex systems, especially in the application of train delay prediction, raises the following questions that need to be discussed:

(1) Component complexity caused by a complex, hierarchical architecture and operational process makes the

characteristic factors for complex systems self-related and cross-related [7]. As a result, the influencing factors of train delay in railway systems should be distinguished by their inherent attributes and the degree of closeness among them.

- (2) Influencing factors from different categories in railway systems have different characteristics of self-organization and self-evolution. To improve the prediction accuracy of train delay, the model needs to structure the corresponding deep learning components for different categories of influencing factors.
- (3) Some fluctuations and inputs from external environments may greatly change the interactions that connect the individual agents in railway systems. Therefore, the previous state of each characteristic factor will have an impact on the later state of the train delay. We need a powerful model to mine the intrinsic temporal characteristics of train delay. However, the existing models may have shortcomings in processing time-series data.

Predicting train delays accurately is crucial for railway enterprises to promptly reschedule the timetable, despite the

TABLE 1: Disadvantages of state prediction in some areas of real-world complex systems.

Prediction target	Model	Disadvantage	Reference number
Building thermal load	LSTM	Weather data need to be processed separately	[22]
Heating demand	RNN	Difficult to process long-term thermal load	[24]
Thermal load of district heating networks	DNN	The attribute of characteristic should be considered	[25]
Building energy consumption	GRU	Only classify the building energy data into time categories	[26]
Numerical weather prediction	DNN	The spatial characteristic of the grid point should be mined by the model	[27]
Storm surges	CNN_LSTM	No model is built for spatial characteristics	[28]
PM2.5 concentration	CNN_LSTM	PM _{2.5} influencing factors need to be distinguished	[29]
Temperature and wind speed	3D CNN_FNN	The time characteristics of dynamic systems should be considered separately	[30]
Useful life of the complex system	LSTM	The efficiency of the model could be improved	[31]
Thermal and cooling consumption of building	1D CNN	The model's ability to process time series data should be improved	[4]
Train delay	DNN	External influence should be further analyzed	[32]
Train delay	LSTM	Spatial relationships among different stations should be considered	[33]
Train delay	LSTM	Time sequence characteristics mining of train influencing factors should be improved	[34]

complexity of the issue within railway systems [8]. Railway dispatchers can use precise delay prediction to evaluate train statuses, make informed decisions, and optimize transportation networks for enhanced efficiency. Moreover, accurate delay prediction data can help passengers estimate travel times and revise plans, improving the service quality of railway systems [9]. Additionally, railway systems are complex systems that can be influenced by the environment, anthropogenic factors, and train events, which may interact with each other [10, 11]. Given the fact that events may be dependent on two stations and trains, it is possible for the operations of railway systems to be both self-correlated and cross-correlated [12, 13]. Self-correlated factors in railway systems can contribute to the temporal characteristic of train delay. Cross-correlated events in railway systems can result in train delays exhibiting spatiotemporal coupling characteristics. Extracting the features of train delays from railway operation data is difficult, especially due to the interplay between temporal and spatial factors. Therefore, predicting train delays remains a challenging task. Most prior studies have categorized characteristic factors into a single category, such as time. However, these models fail to accurately capture the interaction between internal and external factors in railway systems. Moreover, these studies did not take into account the correspondence between the model and the characteristic factors, especially the correspondence model for spatial characteristics.

Currently, some applications of hybrid deep learning models [14] have provided us with a new idea to address the problem of correspondence between the model and the characteristic factors in the prediction of train delay. To capture more comprehensive information, deep learning models with different features are integrated to create hybrid models [15]. With hybrid deep learning models, different characteristic factors can be fed into correspondence models

to improve performance. For instance, in [16, 17], temporal features were first extracted from the data by feeding it into the long short-term memory (LSTM) model, and the spatial features were then further retrieved by feeding the features into the CNN model. An alternate serial link between LSTMs and CNNs in [18] was proposed, wherein CNN was initially utilized. In [19], the serial CNN-LSTM structure of a 1D and a 2D CNN was compared. Following this research direction, the study designs a novel hybrid deep learning model, named the CNN + TCN model, to predict the delay of trains. In the CNN + TCN model, the characteristic factors are divided into the 3D spatiotemporal category and the 1D time series category. Then, the CNN + TCN model employs a 3D CNN component that is fed into the 3D data to mine the spatiotemporal characteristics, as well as a temporal convolutional networks (TCN) component that captures the temporal characteristics of railway operation data.

The proposed model can provide a novel approach to improving accuracy and convergence performance. The main contributions of this work are summarized as follows:

- (1) Considering the complexity of characteristic factors in railway systems, we sort the influencing factors into spatiotemporal categories and time series categories. Then, we build the real-world train data into 3D data as the spatiotemporal category.
- (2) A TCN component is structured to reflect the complex dynamic characteristics of railway system operation and is applicable to time series data in railway systems. In addition, this TCN component can improve the convergence of the hybrid model.
- (3) The proposed deep learning framework includes one TCN component and one 3D CNN component to address the problem of correspondence between the

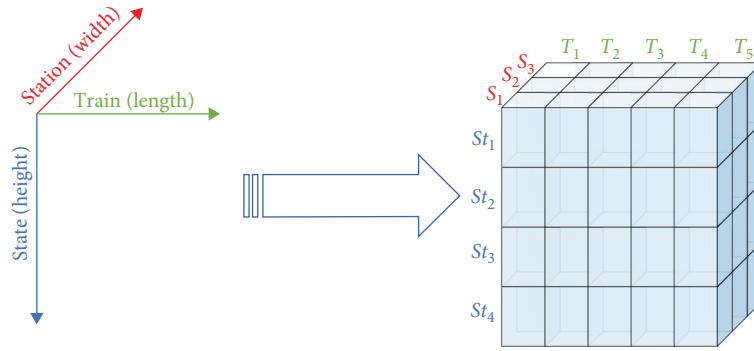


FIGURE 1: Schematic diagram of input data of 3D CNN (width = 5, height = 3, and length = 4).

model and the characteristic factors in railway systems, which can promote the predictive performance of the model.

The article's remainder is organized as follows: In Section 2, the previous studies are reviewed. In Section 3, the research first introduces 3D CNN and builds 3D data containing the spatio-temporal characteristics of real-world train data. Then, the research structures a TCN component that is applicable to railway systems. Furthermore, the architecture of the proposed model, named the CNN + TCN model, is introduced. In Section 4, we describe the railway data for two lines in the United Kingdom and select the characteristic variables. In addition, a numerical investigation of train delay prediction is carried out to validate the effectiveness of the proposed method. Finally, our work in Section 5 is concluded.

2. Related Work

Deep learning is an effective technology for dealing with large amounts of data, extracting inherent features, and modeling nonlinear phenomena [20]. Nowadays, some deep learning approaches have been applied in the state prediction area of complex systems, such as the deep neural network (DNN) [21], the LSTM [22], the gated recurrent unit (GRU) [23], the RNN [5], and the CNN [4]. However, the internal and external features of complex systems cannot be ignored. Table 1 shows the disadvantages of state prediction in some areas of real-world complex systems. In some fields, predicting the state of complex systems may be regarded as merely ordinary deep learning prediction problems that do not consider the inherent features and correlations of influencing factors. More importantly, it can be seen that some research has considered the dynamic characteristics of complex systems and structured models for mining temporal characteristics. However, all influencing factors of the complex system are only classified in the time series category without considering its other intrinsic properties. For example, the prediction for energy consumption of buildings did not consider the impact of visitor flowrate (as shown in Table 1). These temporal-characteristic mining models also have low accuracy and convergence performance. Some research has used hybrid deep learning methods to mine multiple intrinsic attributes of influencing factors. The prediction

performance of these hybrid models has improved compared with individual deep learning models. However, the corresponding models for the influencing factors of different attributes that affect the further improvement of prediction accuracy have seldom been considered.

As shown in Table 1, the disadvantages of some research in the prediction field of train delay have been found due to ignoring the spatial characteristics of railway systems. In addition, Sun et al. [35] utilized a radial basis function network to predict the delay time at technical stations. This method is only applicable to large-scale technical stations with longer historical data delays. Oneto et al. [36] designed a train delay prediction system (TDPS), which used big data technology and the deep learning machine algorithm. However, TDPS cannot effectively mine the temporal features of real-world train data. Zhang et al. [37] proposed a method to predict train-associated delay that used the Morlet mother wavelet basis function as an excitation function. Through the analysis of previous research, we find that the existing methods ignore the relationships between stations and the dependencies of railway events between trains and stations. In addition, existing research has the disadvantage of mining the temporal relationship between adjacent trains. As a result, this study decides to use a 3D CNN component to capture the operation interactions in railway systems. The superiorities of this study are as follows:

- (1) The proposed method focuses on the internal characteristics of influencing factors in train delay with different categories, which can improve the perception of comprehensiveness.
- (2) The study employs a more advanced time-series data processing architecture, TCN, which can better minimize the time relationship between adjacent trains and reduce the data processing time.
- (3) The proposed method can process time data and spatiotemporal data from the railway system simultaneously, which enhances predictability.

2.1. The Delay Prediction Model. According to [38–41], the operation's interactions in the railway system can be represented by the spatiotemporal relationship between stations and the dependencies of events between trains and stations.

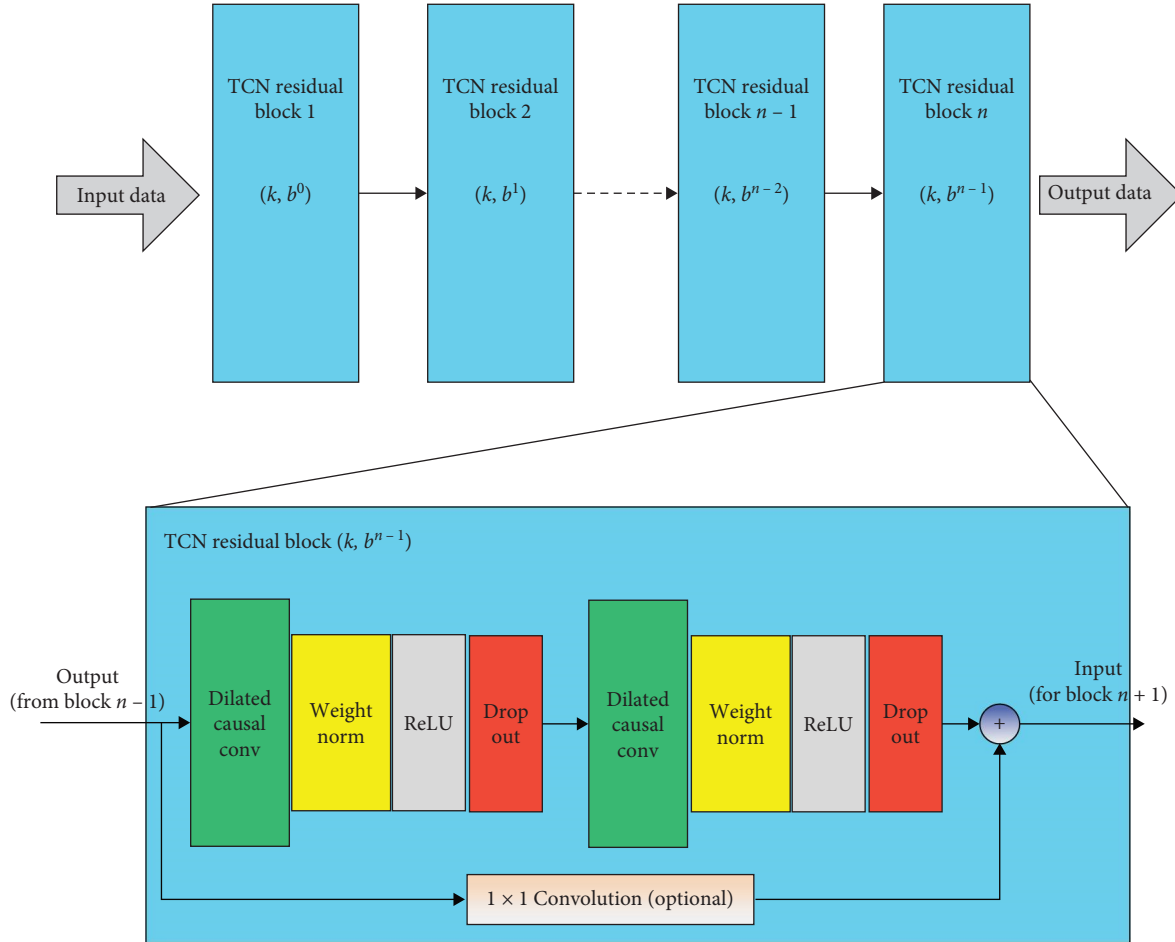


FIGURE 2: Schematic diagram of TCN architecture.

A model capable of capturing spatiotemporal relationships is required to extract the interdependencies between time and space events in the railway system. Furthermore, previous research indicates that the influencing factors for train delays exhibit time-series properties. As a result, this study sorts the influencing factors into spatiotemporal categories and time series categories. In this section, we first introduce 3D CNN, which has the ability to process spatiotemporal data. Next, we construct 3D data incorporating the spatiotemporal attributes of the railway system. Then, we structure a TCN component to mine temporal features from railway data. Finally, the CNN + TCN model is introduced.

2.2. CNN. 2D CNN can better extract spatial features from image data by processing multiple pixels simultaneously with convolution filters [42]. 2D image data in a 2D CNN is converted into 3D data. The real-world train data have obvious spatial-temporal characteristics [40], whereas 2D CNN could not mine spatial characteristics from the data with such characteristics. On the other hand, 3D CNN can effectively process image data with temporal sequence information, such as video data [43]. 3D CNN data, including video duration, has a time characteristic, which is the third dimension in its convolutional filters. In contrast to 2D CNN, the convolutional filters of 3D CNN are 3D. The third dimension in the convolutional filters of

3D CNN is the time depth [44, 45]. This allows for constant temporal characteristics, enabling 3D CNN to extract spatiotemporal coupling relationships from input data [46]. As a result, this study utilizes 3D CNN for time-space characteristic learning and video analysis in railway systems, focusing on the spatiotemporal coupling characteristic to extract real-world train data properties.

Moreover, pooling layers are added after 3D CNN filters, and they calculate the averaged or maximum value of output maps [46]. The number of parameters required for training the 3D CNN can be significantly decreased by using the pooling layers to obtain the crucial values that identify the characteristics. The pooling layer significantly reduces time investment for CNN models while maintaining model accuracy. The CNN can be configured with either global or local pooling layers. Smaller kernel sizes are employed in local pooling (e.g., $2 \times 2 \times 2$ for 3D CNN). Global pooling, on the other hand, has an impact on all feature maps acquired through the layers of convolution. In the paper, we choose local and max pooling as these approaches are effective in capturing vital characteristics.

This study puts real-world train data with spatiotemporal characteristics into a 3D CNN. To mine spatial characteristics utilizing 3D CNN, we convert real-world train data into 3D data. As illustrated in Figure 1, the width of the 3D data represents the stations on the railway line, and there is a

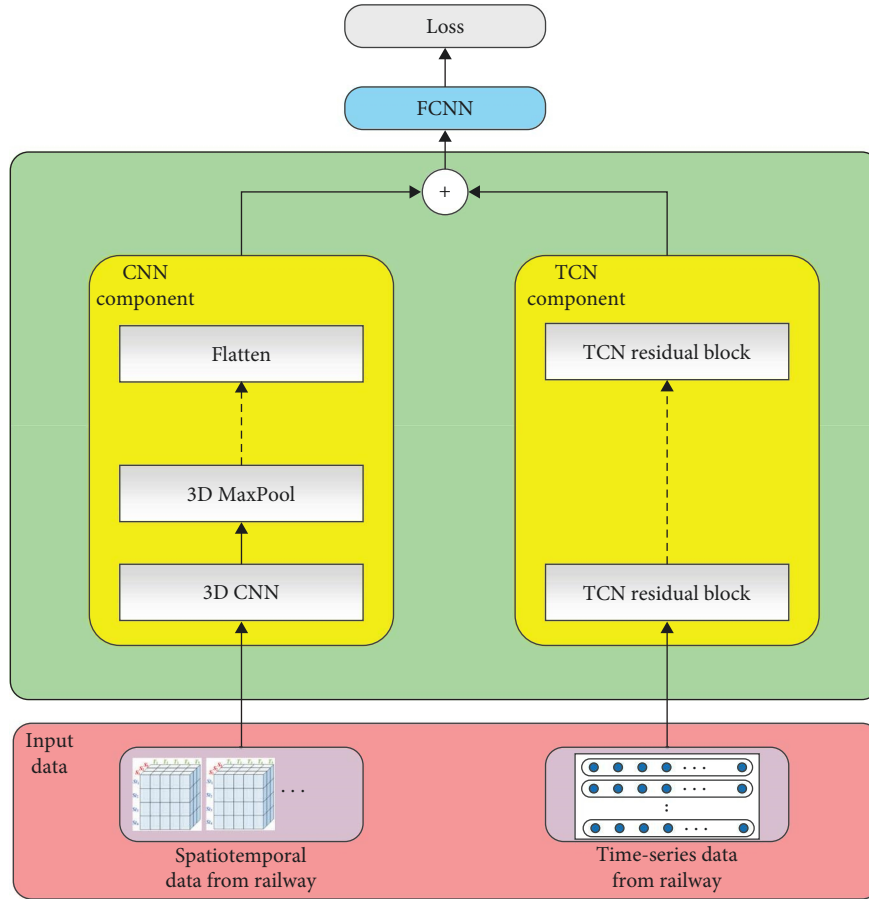


FIGURE 3: Structure diagram of the CNN + TCN model.

spatial relationship among them. The length indicates the quantity of trains on a single rail track. The train's departure has a sequence and a spatiotemporal relationship. What is more, the height of the 3D data represents the train's state at a station. As shown on the left side of Figure 1, the width is set to 3 in the constructed 3D data. Thus, each 3D piece of data contains information from three stations (S_1-S_3), which have adjacent spatial locations. The length is set to 5, which means that each 3D data set contains five trains (T_1-T_5), which are arranged in order of departure. The height is set to 4, which means that the train has four states (St_1-St_4) at each station. The states mentioned are derived from the recorded arrival and departure events at each station. Therefore, the data shape of the input 3D CNN is (3, 5, 4). After constructing the 3D data, the 3D CNN extracts the spatiotemporal features of 3D data through 3D convolution filters. The paper sets the depth of the input 3D CNN data to 1, indicating the output, which is the train delay.

2.3. TCN. In 2017, Lea [47] pioneered a TCN that integrates extended causal convolution and residual connection. TCN has a back-propagation path, which avoids the "gradient explosion" or "disappearance problem" that often occurs in the RNN [47–49]. TCN is produced by stacking numerous layers of TCN residual blocks, which results in the formation

of TCN. As shown in Figure 2, one TCN residual block contains two 1D extended causal convolutions with an identical expansion coefficient [50]. The extended causal convolution enables TCN to quickly capture causal relationships between longer temporal series. The expansion causal convolution layer of each TCN residual block has a different expansion coefficient b^{n-l} , where b denotes the expansion basis of the TCN residual block, and n denotes the layer's number in the TCN residual block. In addition, k is the size of the 1D convolution filter.

As shown on the lower side of Figure 2, a weight normalization layer is added to each layer to counteract the gradient explosion problem of the network. The above layer normalizes the input of the hidden layer. To make the module nonlinear, a rectified linear unit (ReLU) activation function is inserted into the 1D convolution layer. Additionally, the ReLU activation function also ensures that the output can have a negative value. Besides, the drop-out layer prevents overfitting. Finally, the introduction of 1×1 convolution residual connections directly maps the input to solve the degradation of the deep network [51]. The number n of TCN residual blocks is obtained as follows:

$$n \geq \log_b \left(\frac{(p-1)(b-1)}{2(k-1)} + 1 \right), \quad (1)$$

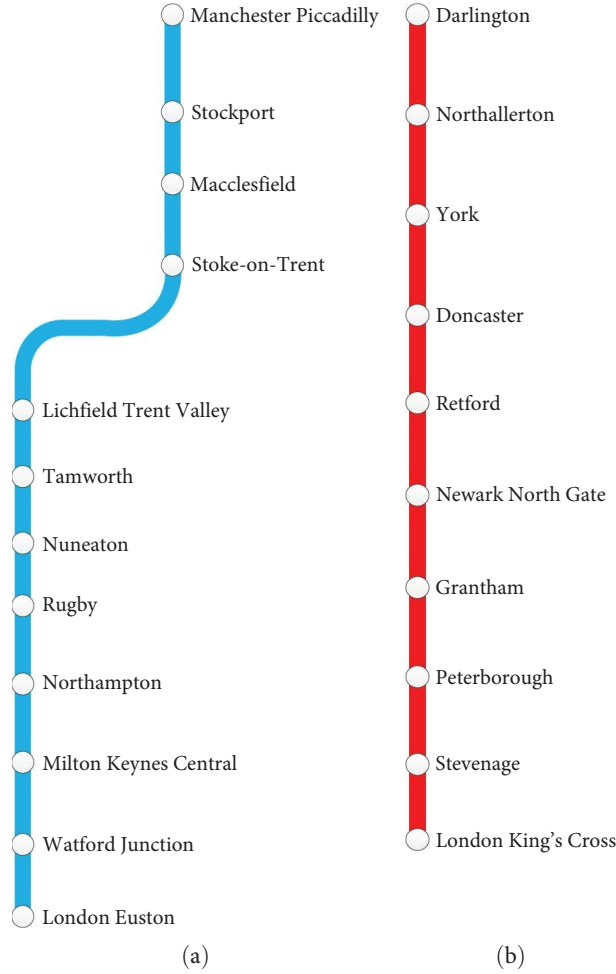


FIGURE 4: Station diagram of two railway lines: (a) from London Euston to Manchester Piccadilly; (b) from London King's Cross to Manchester Piccadilly.

where b is the expansion basis, k is the size of a 1D convolution filter, and p denotes the length of the temporal series. Combined with the characteristics of train delay prediction, the temporal series length p is set to 5, and the expansion base b is set to 2. Therefore, according to Equation (1), TCN is set as a two-layer residual block.

2.4. The CNN + TCN Model. In this research, a model for predicting train delays named CNN + TCN is proposed. In the proposed model, a CNN component provides features related to spatial correlation, and a TCN component provides features related to temporal series. The structure diagram of the CNN + TCN model is shown in Figure 3. Spatiotemporal data from train operations is transmitted to the CNN component, and temporal series data from train operations is transmitted to the TCN component. The CNN component first includes a 3D convolution layer. Behind the 3D convolution layer, a maximum pooling layer decreases the number of parameters while preserving the model's essential characteristics. To realize a fusion step, a flat layer is added. Meanwhile, a TCN component with two layers of remaining TCN blocks is set up on the right side of Figure 3.

Next, a fusion method is adopted to connect the output data from two components side by side. If the output sizes of the TCN component and CNN component are (Q, V) and (Q, U) , the combined data should be $(Q, V + U)$, where Q is the data amount of test or training. The merged data are subsequently run through a fully connected neural network (FCNN) layer to increase the weight of each unit. To determine the loss, a comparison is performed between the prediction value and the actual delay data, employing an objective function. The proposed model calculates the loss function by leveraging the mean square error (MSE). In addition, the activation functions of FCNN and 3D CNN use the parametric rectified linear unit (PReLU) function.

3. Results and Discussion

3.1. Description of Real-World Train Data. In this article, train data from the United Kingdom were used to train and calibrate the model. This study chose two lines, namely, from London Euston station to Manchester Piccadilly station (EUS–MAN) and from London King's Cross to Darlington (KGX–DAR). The EUS–MAN line is a railway line operated

TABLE 2: Validation results of CNN + TCN for the different number of 3D CNN layers.

Depth	3D CNN	Loss	TCN	Loss
1st	3D CNN (32)	1.430	TCN (64)	1.012
2st	3D CNN (64)	1.461	TCN (64)	1.139
3st	3D CNN (64)	1.460	TCN (64)	1.139

by Avanti West Coast. As shown in Figure 4(a), the EUS–MAN line passes through 12 railway stations. The KGX–DAR line belongs to the east coast lines operated by the London North East Railway (LNER). As shown in Figure 4(b), the KGX–DAR line passes through ten railway stations.

Real-world train data were collected for 9 months, from October 1, 2021, to June 30, 2022. The KGX–DAR line collected 10,968 train operation records, and the EUS–MAN line collected 9,810 train operation records. The database collected includes planned/actual departure time, planned/actual arrival time, planned/actual operation time, train number, delay, dwell time, and date of each train at each station. To improve the data’s quality, abnormal raw data are processed as follows:

- (1) Deleting the abnormal delay (generally larger than 120 min) that may cancel the train due to a large delay.
- (2) Using the weighted average of adjacent records to process the missing data.

Next, the data are categorized into spatiotemporal feature data processed by 3D CNN and temporal series data processed by TCN. Train delays can generally be categorized into two types: self-delay and joint-delay [52]. Self-delay refers to the initial delay of a train caused by factors such as weather conditions, track equipment issues, or the railway power system. On the other hand, the joint-delay occurs when delays from preceding trains spread and affect subsequent trains [37]. Regardless of the type, the arrival delay time and departure delay time, which depend on the location and time of the train at each station, directly reflect the extent of the train delay. Therefore, the spatiotemporal characteristic parameters include (it is assumed that the delay of the k station is to be predicted) arrival delay time W_a (from the $n-k$ station to the $n-1$ station) and departure delay time W_d (from the $n-k$ station to the $n-1$ station).

Furthermore, the train delay is directly related to the state of its adjacent train. The tracking interval between two trains represents the minimum time required for two adjacent trains to remain undisturbed from each other [52]. When the preceding train experiences a delay, the actual tracking interval between the trains immediately reflects the impact of its delay on the following train delay. The actual dwell time at the station indicates the impact of the delay within the railway system on each train. The system delay may result in increased actual dwell time for each train. After the factors that cause the delay in the railway system are eliminated, efforts to mitigate the impact of delays may include shortening the actual stop dwell

time at certain stations. The train’s speed may decrease during delays, resulting in longer travel times between adjacent stations [53]. Thus, the actual travel time between adjacent stations can serve as an indicator of the degree of train delay. Additionally, the timetable has an impact on the train delay. Different scheduled timetables contain variations in travel times, dwell times, and tracking intervals, which affect the recovery capabilities of the railway system following delays [54]. Hence, based on the above analysis, the temporal characteristic parameters include the following (it is assumed that the delay of the k station is to be predicted):

- (1) Scheduled travel time T between two stations (from the $n-k$ station and the $n-k+1$ station to the $n-1$ station and the n station).
- (2) Scheduled dwell time P (from the $n-k$ station to the $n-1$ station).
- (3) Scheduled tracking interval I between two trains (from the $n-k$ station to the $n-1$ station).
- (4) Actual travel time T' between two stations (from the $n-k$ station and the $n-k+1$ station to the $n-1$ station and the n station).
- (5) Actual dwell time P' (from the $n-k$ station to the $n-1$ station).
- (6) Actual tracking train interval time I' between two trains (from the $n-k$ station to the $n-1$ station).

After determining the above characteristic parameters, the corresponding characteristic parameters are extracted as the input data for the 3D CNN and TCN. Then, this study randomly selects 75% of the real-world train data as the training set. The residual 25% train data is selected as the test set for subsequent verification.

3.2. Model Parameter Selection. We conducted a trial on model parameters using KGX–DAR line data to enhance prediction performance, highlighting the significant impact of the depth and quantity of neurons, units, and filters in each layer. We employ the parameter optimization strategy that is often used in the literature (e.g., [55, 56]) as the two of these parameters are interrelated. In this strategy, manually figuring out the depth and the number of neurons, units, or filters in each layer is used to create a suitable model. Previous research has demonstrated that each layer of a neural network typically contains 32, 64, or 128 neurons/units/filters [46, 55, 56]. The model is initially trained using one TCN and one 3D CNN layer. As validation loss decreases, the layers are added. The first CNN layer has 64 neurons, while each TCN layer has 64 units, considering the total number of influencing factors.

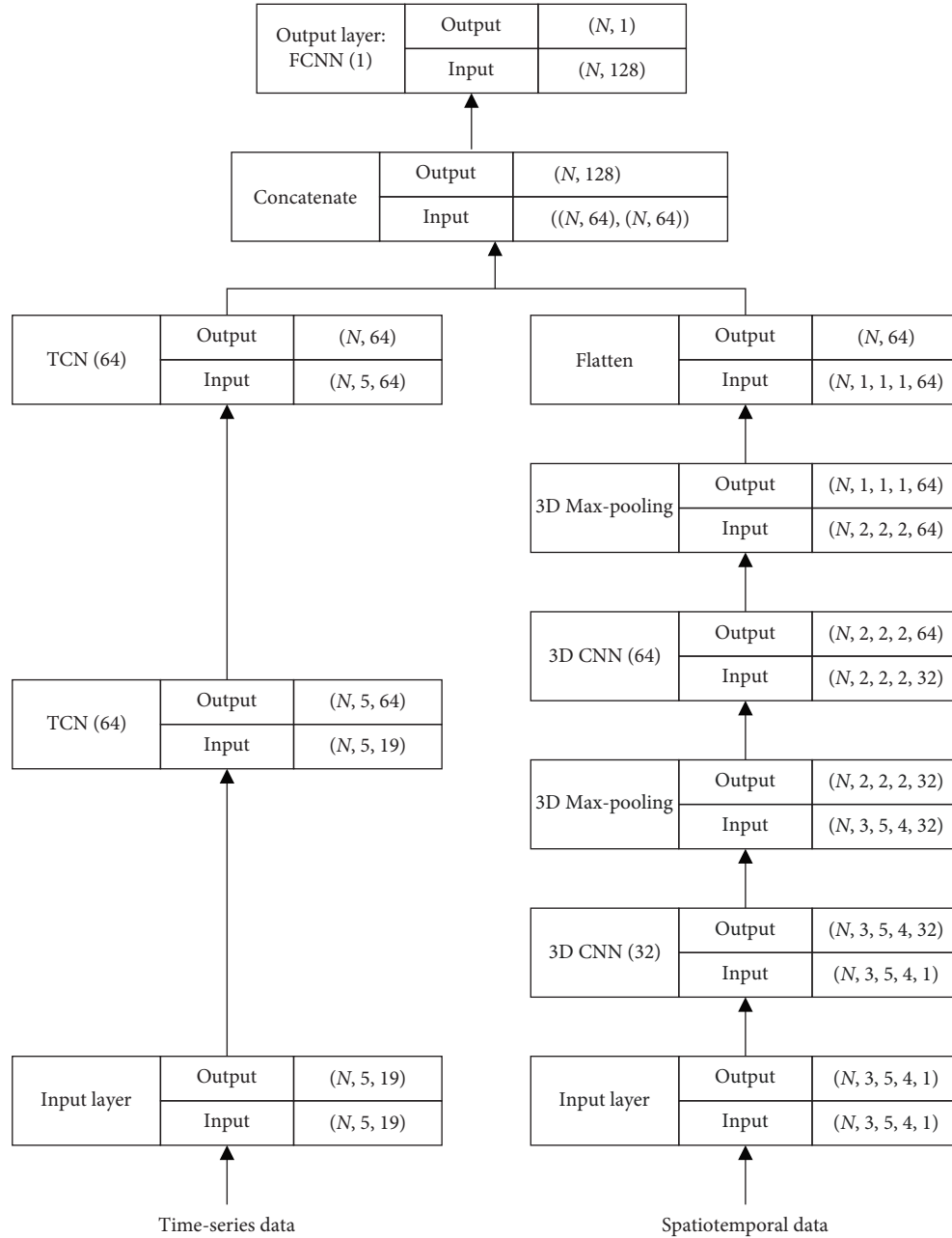


FIGURE 5: Data flow and data form of CNN + TCN model.

The trial results are shown in Table 2. The findings showed that the validation loss initially decreases as the depth/number of layers of 3D CNNs and TCNs increases, but it does not continue to decrease as the number of layers increases to a certain number. Meanwhile, the results suggest that data under-fits with fewer layers, well-fits with enough layers, and over-fits with more complex model structures as the structure becomes more complex. Figure 5 outlines the regular architecture consisting of two 3D CNN layers with 32 filters and 64 filters and two TCN layers with 64 TCN units to ensure a well-fitted model and prevent overfitting. A max-pooling layer is added after each 3D CNN in order to capture

crucial characteristics. Furthermore, we manually adjust the settings for those that do not significantly affect the accuracy of the model based on guidelines gleaned from fruitful research. For instance, in order to distinguish between tiny and local characteristics, the filter size and pooling size have been set to $2 \times 2 \times 2$ [46]. To ensure that the output data of the convolutional layers is the same size as the input data, the padding parameter in the 3D CNN portion is set to the same padding. The pooling kernel and convolutional filters have stride values of $2 \times 2 \times 2$ and $1 \times 1 \times 1$, respectively. The Adam optimizer was utilized, which is used to adjust the local learning rate during training [57].

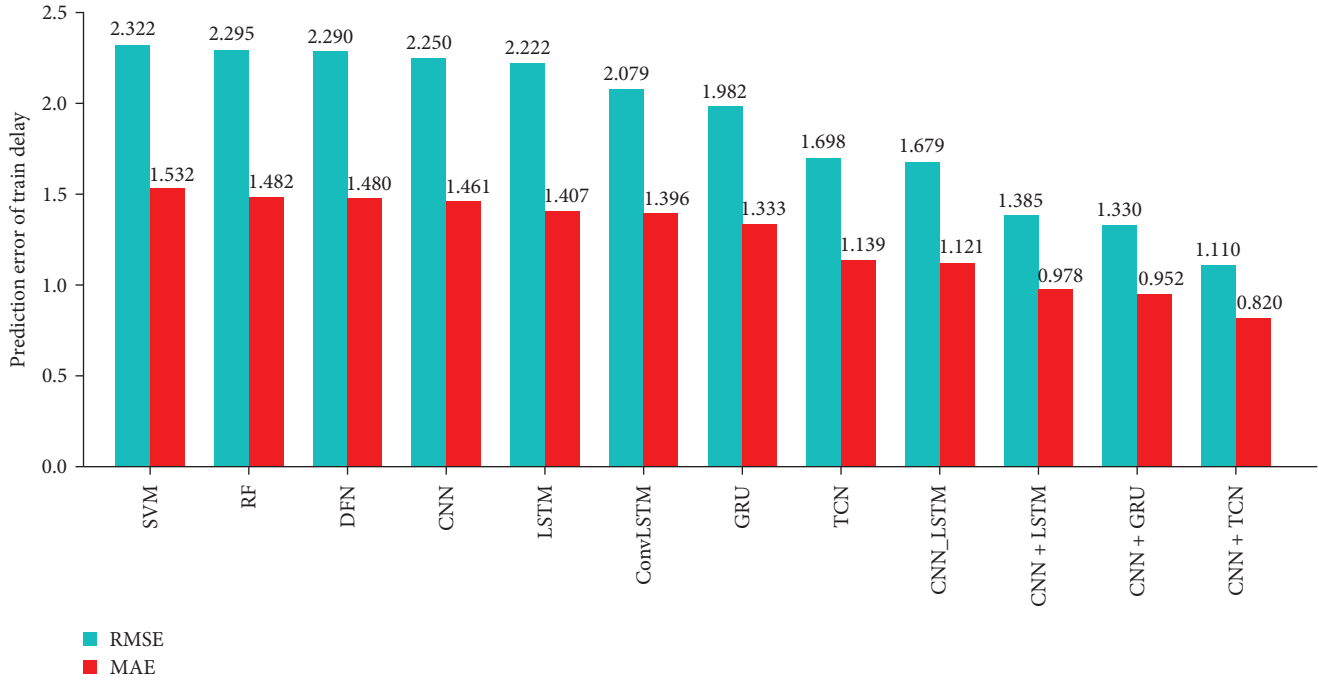


FIGURE 6: Prediction error of train delay of each model on KGX-DAR line.

3.3. *Verification Results.* To verify the model's performance, CNN + TCN is compared with the following baseline models:

- (1) DNN: DNN is a neural network with a large number of hidden layers [21].
- (2) LSTM: LSTM is able to avoid gradient disappearance and remember the dependence of long term, which is an improvement kind of RNN.
- (3) GRU: GRU and LSTM are very similar; both add a gating mechanism in the RNN unit to selectively add or forget historical information. GRU omits the forgetting gate of LSTM and has a faster training speed and fewer parameters.
- (4) TCN: TCN has been introduced in Section 3.2.
- (5) CNN: CNN has been introduced in Section 3.1.
- (6) ConvLSTM: ConvLSTM is a peer-to-peer architecture, designed by applying the convolution calculations of LSTM to mine multiple types of features [56, 58].
- (7) CNN_LSTM: CNN_LSTM first uses the CNN to process data and then transmits the processed data to LSTM [29].
- (8) CNN + GRU: CNN + GRU is obtained by replacing the TCN in the CNN + TCN model using GRU.
- (9) CNN + LSTM: CNN + LSTM is obtained by replacing the TCN in the CNN + TCN model using LSTM.
- (10) Random forest (RF): An RF maps predictor and dependent variable vectors using an ensemble of decision trees [59]. The average of the trees is the outcome of a regression RF.

- (11) Support vector machine (SVM): SVM is a class of generalized linear classifiers that perform binary classification on data using supervised learning [60]. Its decision boundary is the maximum margin hyper-plane solved for the learning sample.

In addition, we choose mean average error (MAE) and root mean square error (RMSE) as the performance criterion.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (q_i - \hat{q}_i)^2} \quad (2)$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |q_i - \hat{q}_i|, \quad (3)$$

where \hat{q}_i denotes a predicted value, q_i denotes an observed value, and N is a sample size.

Figures 6 and 7 show the prediction errors of KGX-DAR line data and EUS-MAN line data in each model, respectively. The following results are obtained via a comparison of the prediction errors:

- (1) The RMSE and MAE of the DFN and CNN models are higher than those of TCN, ConvLSTM, LSTM, and GRU, which indicates that the train data have an obvious time characteristic.
- (2) TCN has the lowest overall prediction error compared to all of the other individual models. The findings suggest that the TCN model offers a number of clear

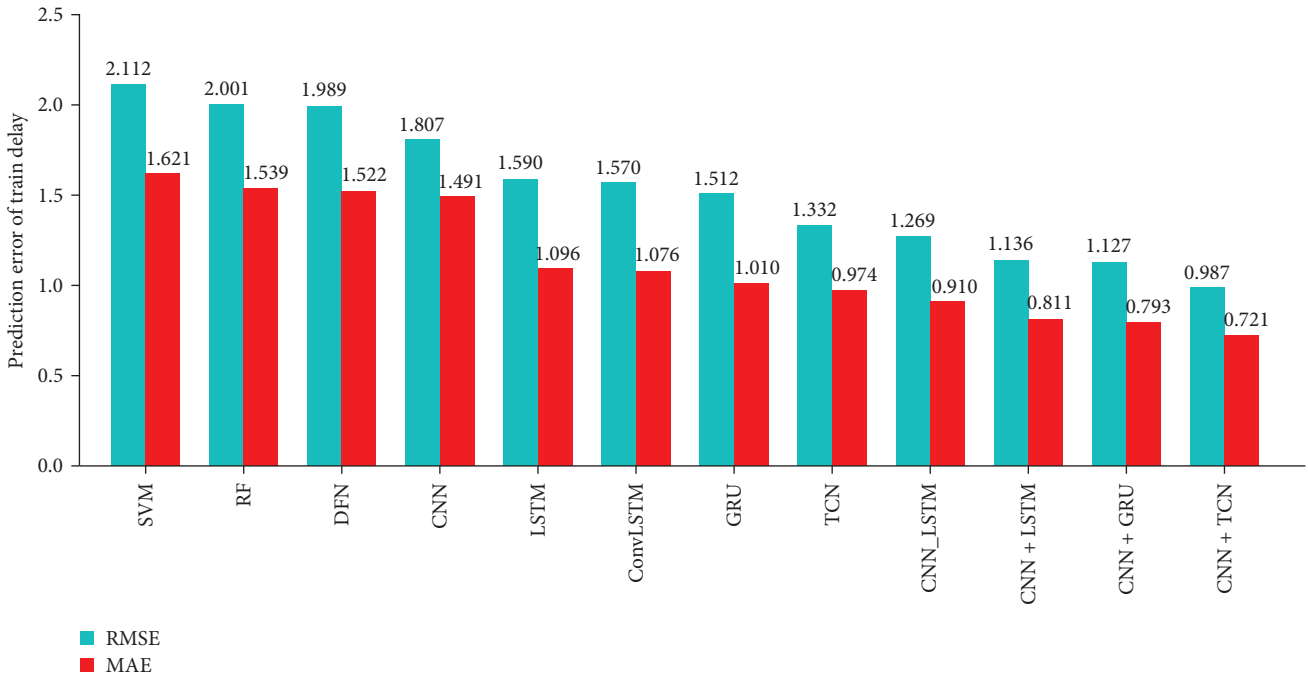


FIGURE 7: Prediction error of train delay of each model on EUS-MAN line.

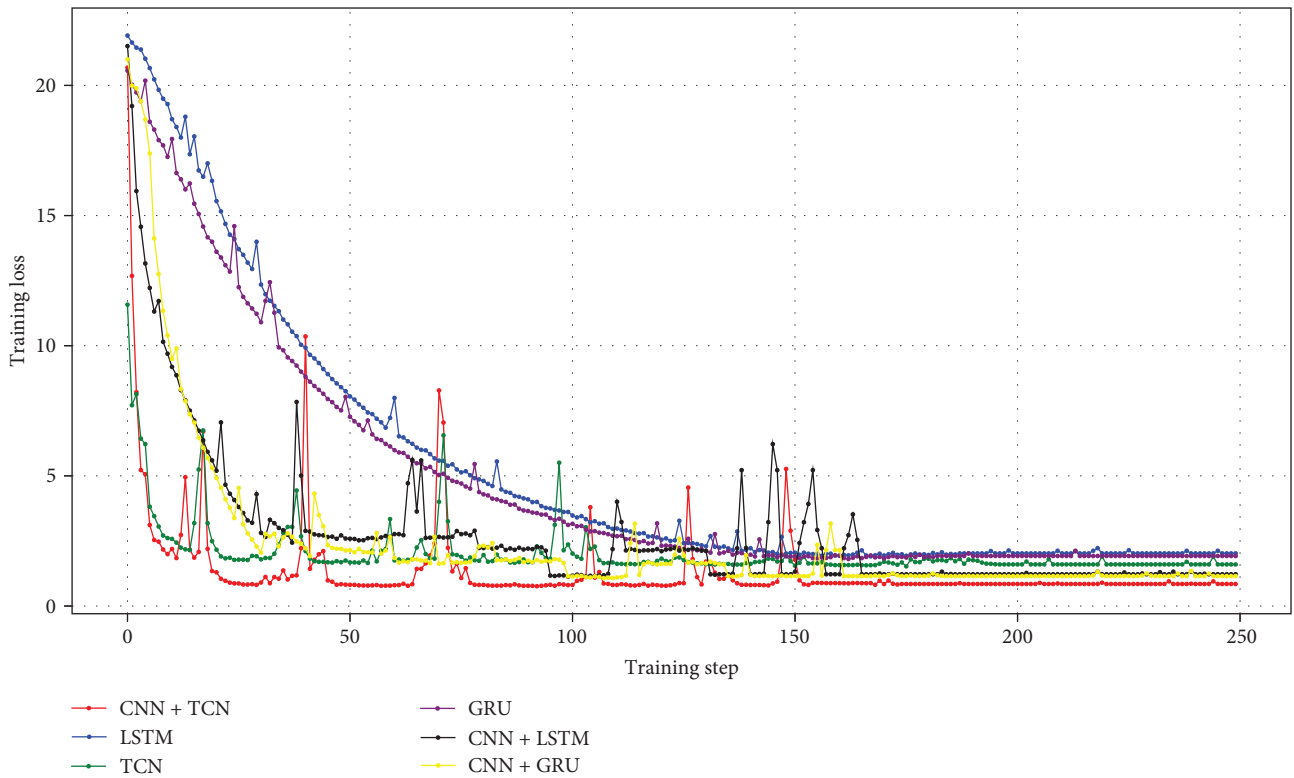


FIGURE 8: Training process of KGX-DAR line data in each model.

benefits when it comes to mining the time-temporal characteristics of data pertaining to real-world trains.

(3) The RMSE and MAE of the CNN_LSTM model are lower than those of all individual models,

which shows the advantages of hybrid models and the importance of spatial feature mining. However, the CNN_LSTM prediction error is higher than that of other hybrid models, which

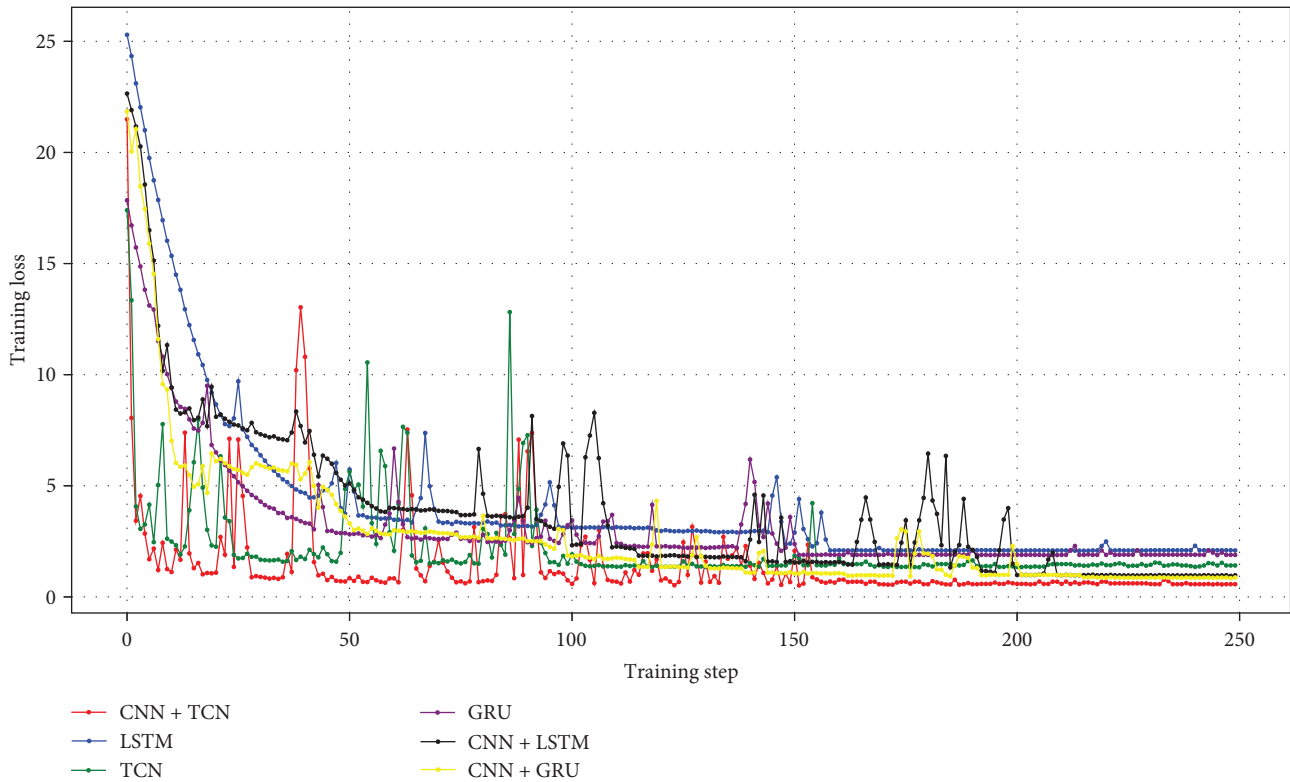


FIGURE 9: Training process of EUS-MAN line data in each model.

indicates the superiority of feature classification processing.

- (4) By comparing all models, the RMSE and MAE of the CNN + TCN model are the smallest. The results reveal that CNN + TCN has significant advantages in data classification and processing time series data.

This research also verifies the convergence of the CNN + TCN model. The process of training is depicted in Figures 8 and 9, respectively.

By observing the training error of each model, we can obtain the following results:

- (1) By comparing TCN, GRU, and LSTM, TCN converges first. In the EUS-MAN line, TCN converges after approximately 100 steps, whereas GRU and LSTM converge after approximately 150 and 160 steps, respectively. In the KGX-DAR line, TCN converges at about 110 steps, whereas GRU and LSTM converge after approximately 140 and 145 steps, respectively.
- (2) CNN + LSTM is later than the convergence of LSTM. CNN + GRU is later than the convergence of GRU. This means that the process of mining spatial characteristics by CNN increases the number of training steps. In the EUS-MAN line, CNN + LSTM and CNN + GRU increase by approximately 50 steps. In the KGX-DAR line, CNN + LSTM and CNN + GRU increase by approximately 35 steps.

- (3) CNN + TCN is later than the convergence of TCN. The result shows that the spatial feature mining process of CNN in CNN + TCN also increases the training steps. But the CNN + TCN convergence happens before that of CNN + LSTM and CNN + GRU. In the EUS-MAN line, CNN + TCN is about 50 steps faster than CNN + LSTM and about 40 steps faster than CNN + GRU, respectively. In the KGX-DAR line, CNN + TCN is about 25 steps faster than CNN + LSTM and about 20 steps faster than CNN + GRU, respectively. This shows that using TCN to mine data with temporal characteristics helps speed up convergence.

4. Conclusions

In this study, a novel prediction model named CNN + TCN is proposed. CNN + TCN divides the influencing factors of the railway system into two categories: spatiotemporal categories and time series categories. Furthermore, using a 3D CNN mines the spatiotemporal features from the real-world train data, and utilizing a TCN captures the temporal features. Finally, the research conducts comprehensive experiments on two railway lines in the United Kingdom to test the CNN + TCN model, where the delay prediction accuracy is greatly improved, and the model convergence becomes better. Numerical results mean that the CNN + TCN model is suitable for predicting the states of complex systems.

However, the TCN + CNN model has only been applied to the prediction of train delay in this paper. We can further

improve the model by validating other problems in complex systems. In particular, the 3D CNN component, responsible for processing spatiotemporal data, needs improvement in its ability to distinguish temporal, spatial, and spatiotemporal coupling features. In the future, we will improve the effectiveness of the 3D component on train delay prediction or other types of complex systems (e.g., electric vehicle charging demand prediction and sustainable energy generation forecast). Moreover, the CNN + TCN model lacks components for feature recognition of influencing factors, potentially impacting the model's accuracy. Therefore, we will investigate intelligent optimization algorithms, such as genetic algorithms combined with deep learning models, and apply them to address train delay issues.

Data Availability

The dataset analyzed during this study is available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61803147, 72201218, and 52002339) and Fundamental Research Funds for the Central Universities (no. 2682022CX028).

References

- [1] F.-Y. Wang, J. Guo, G. Bu, and J. J. Zhang, "Mutually trustworthy human-machine knowledge automation and hybrid augmented intelligence: mechanisms and applications of cognition, management, and control for complex systems," *Frontiers of Information Technology & Electronic Engineering*, vol. 23, no. 8, pp. 1142–1157, 2022.
- [2] F. Wang, "Parallel system methods for management and control of complex systems," *Control and Decision*, vol. 19, no. 5, pp. 485–514, 2004.
- [3] T. Hey, S. Tansley, and K. Tolle, *The Fourth Paradigm: Data-Intensive Scientific Discovery*, Microsoft Research, Redmond, Washington, 2009.
- [4] T. Ji and T. Wang, "Building energy consumption prediction based on word embedding and convolutional neural network," *Journal of South China University of Technology. Natural Science Edition*, vol. 49, no. 6, pp. 40–48, 2021.
- [5] A. Rivas, J. M. Fraile, P. Chamoso, A. Gonzalez-Briones, I. Sitton, and J. M. Corchado, "A Predictive maintenance model using recurrent neural networks," in *14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO)*, pp. 261–270, Springer, 2019.
- [6] M. Yaghini, M. M. Khoshraftar, and M. Seyedabadi, "Railway passenger train delay prediction via neural network model," *Journal of Advanced Transportation*, vol. 47, no. 3, pp. 355–368, 2013.
- [7] J. Hou, H. Ma, D. He, J. Sun, Q. Nie, and W. Lin, "Harvesting random embedding for high-frequency change-point detection in temporal complex systems," *National Science Review*, vol. 9, no. 4, 2022.
- [8] P. Huang, J. Guo, S. Liu, and F. Corman, "Explainable train delay propagation: a graph attention network approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 184, Article ID 103457, 2024.
- [9] Y. Liu and Z. Wang, "Optimization of transfer passenger flow connection schemes under conditions of previous train delays," *Journal of Beijing Jiaotong University*, vol. 48, no. 1, pp. 1–13, 2024.
- [10] J. Wang, M. Granlof, and J. Yu, "Effects of winter climate on delays of high speed passenger trains in Botnia-Atlantica region," *Journal of Rail Transport Planning & Management*, vol. 18, Article ID 100251, 2021.
- [11] J. Guo, W. Wang, J. Guo, A. D'Ariano, T. Bosi, and Y. Zhang, "An instance-based transfer learning model with attention mechanism for freight train travel time prediction in the China–Europe railway express," *Expert Systems with Applications*, vol. 251, Article ID 123989, 2024.
- [12] N. Markovic, S. Milinkovic, K. S. Tikhonov, and P. Schonfeld, "Analyzing passenger train arrival delays with support vector regression," *Transportation Research Part C: Emerging Technologies*, vol. 56, pp. 251–262, 2015.
- [13] J. Guo, W. Wang, Y. Tang, Y. Zhang, and H. Zhuge, "A CNN-Bi_LSTM parallel network approach for train travel time prediction," *Knowledge-Based Systems*, vol. 256, Article ID 109796, 2022.
- [14] Y. Ye, P. Huang, Y. Sun, and D. Shi, "MBSNet: a deep learning model for multibody dynamics simulation and its application to a vehicle-track system," *Mechanical Systems and Signal Processing*, vol. 157, Article ID 107716, 2021.
- [15] Y. G. Tang, K. Yang, S. J. Zhang, and Z. Zhang, "Photovoltaic power forecasting: a hybrid deep learning model incorporating transfer learning strategy," *Renewable and Sustainable Energy Reviews*, vol. 162, Article ID 112473, 2022.
- [16] K. J. Wang, X. X. Qi, and H. D. Liu, "Photovoltaic power forecasting based LSTM-convolutional network," *Energy*, vol. 189, Article ID 116225, 2019.
- [17] P. Kumari and D. Toshniwal, "Long short term memory-convolutional neural network based deep hybrid approach for solar irradiance forecasting," *Applied Energy*, vol. 295, Article ID 117061, 2021.
- [18] J. Q. Qu, Z. Qian, and Y. Pei, "Day-ahead hourly photovoltaic power forecasting using attention-based CNN-LSTM neural network embedded with multiple relevant and target variables prediction pattern," *Energy*, vol. 232, Article ID 120996, 2021.
- [19] A. Agga, A. Abbou, M. Labbadi, and Y. El Houm, "Short-term self consumption PV plant power production forecasts based on hybrid CNN-LSTM, ConvLSTM models," *Renewable Energy*, vol. 177, pp. 101–112, 2021.
- [20] J. Runge and R. Zmeureanu, "A review of deep learning techniques for forecasting energy use in buildings," *Energies*, vol. 14, no. 3, Article ID 608, 2021.
- [21] M. Gong, H. Zhou, Q. Wang, S. Wang, and P. Yang, "District heating systems load forecasting: a deep neural networks model based on similar day approach," *Advances in Building Energy Research*, vol. 14, no. 3, pp. 372–388, 2020.
- [22] Z. Wang, T. Hong, and M. A. Piette, "Building thermal load prediction through shallow machine learning and deep learning," *Applied Energy*, vol. 263, Article ID 114683, 2020.
- [23] M. Cai, M. Pipattanasomporn, and S. Rahman, "Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques," *Applied Energy*, vol. 236, pp. 1078–1088, 2019.

- [24] A. Rahman and A. D. Smith, "Predicting heating demand and sizing a stratified thermal storage tank using deep learning algorithms," *Applied Energy*, vol. 228, pp. 108–121, 2018.
- [25] G. Suryanarayana, J. Lago, D. Geysen, P. Aleksiejuk, and C. Johansson, "Thermal load forecasting in district heating networks using deep learning and advanced feature selection methods," *Energy*, vol. 157, pp. 141–149, 2018.
- [26] J. Yang, K. K. Tan, M. Santamouris, and S. E. Lee, "Building energy consumption raw data forecasting using data cleaning and deep recurrent neural networks," *Buildings*, vol. 9, no. 9, Article ID 204, 2019.
- [27] Y. Qu, X. Wen, M. Zhang, and Z. Liu, "Develop an objective post-processing system with artificial neural network to improve numerical weather prediction for the olympic winter games beijing 2022," *Acta Scientiarum Naturalium Universitatis Pekinensis*, vol. 58, no. 2, pp. 210–220, 2022.
- [28] B. Wang, S. Liu, B. Wang, W. Wu, J. Wang, and D. Shen, "Multi-step ahead short-term predictions of storm surge level using CNN and LSTM network," *Acta Oceanologica Sinica*, vol. 40, no. 11, pp. 104–118, 2021.
- [29] S. Zou, X. Ren, C. Wang, and J. Wei, "Impacts of temporal resolution and spatial information on neural-network-based PM_{2.5} prediction model," *Acta Scientiarum Naturalium Universitatis Pekinensis*, vol. 56, no. 3, pp. 417–426, 2020.
- [30] S. Mehrkanoon, "Deep shared representation learning for weather elements forecasting," *Knowledge-Based Systems*, vol. 179, pp. 120–128, 2019.
- [31] Q. Fu and H. Wang, "Intelligent prediction for remaining useful life of complex system based on multi-layer LSTM," *Journal of Ordnance Equipment Engineering*, vol. 43, no. 1, pp. 161–169, 2022.
- [32] L. Oneto, E. Fumeo, G. Clerico et al., "Dynamic delay predictions for large-scale railway networks: deep and shallow extreme learning machines tuned via thresholdout," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2754–2767, 2017.
- [33] W. Mou, Z. Cheng, and C. Wen, "Predictive model of train delays in a railway system," in *8th International Conference on Railway Operations Modelling and Analysis (ICROMA)—RailNorrköping 2019*, pp. 913–929, Linköping University Electronic Press, 2019.
- [34] C. Wen, W. Mou, P. Huang, and Z. Li, "A predictive model of train delays on a railway line," *Journal of Forecasting*, vol. 39, no. 3, pp. 470–488, 2020.
- [35] L. Sun, R. Song, S. He, and W. Yin, "Prediction method of train delay time in technology service station," *Journal of Beijing Jiaotong University*, vol. 42, no. 1, pp. 94–98, 2018.
- [36] L. Oneto, E. Fumeo, G. Clerico et al., "Train delay prediction systems: a big data analytics perspective," *Big Data Research*, vol. 11, pp. 54–64, 2018.
- [37] Q. Zhang, F. Chen, T. Zhang, and Z. Yuan, "Intelligent prediction and characteristic recognition for joint delay of high speed railway trains," *Acta Automatica Sinica*, vol. 45, no. 12, pp. 2251–2259, 2019.
- [38] J. Ma, S. Ma, W. Hu, T. Peng, and W. Gui, "Scheduling-measure dependent modelling of delay propagation on a single railway line: a matrix transformation approach," *IET Intelligent Transport Systems*, vol. 15, no. 11, pp. 1429–1439, 2021.
- [39] Y. D. Zhang, L. Liao, Q. Yu, W. G. Ma, and K. H. Li, "Using the gradient boosting decision tree (GBDT) algorithm for a train delay prediction model considering the delay propagation feature," *Advances in Production Engineering & Management*, vol. 16, no. 3, pp. 285–296, 2021.
- [40] P. Huang, Z. Li, C. Wen, J. Lessan, F. Corman, and L. Fu, "Modeling train timetables as images: a cost-sensitive deep learning framework for delay propagation pattern recognition," *Expert Systems with Applications*, vol. 177, Article ID 114996, 2021.
- [41] D. Wang, H. Xu, L. Dai, L. Zhang, and J. Guo, "Enhancing the utilization of renewable generation on the highway with mobile energy storage vehicles and electric vehicles," *Electric Power Systems Research*, vol. 231, Article ID 110311, 2024.
- [42] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [43] C. Cheng, P. Lv, and B. Su, "Spatiotemporal pyramid pooling in 3D convolutional neural networks for action recognition," in *25th IEEE International Conference on Image Processing (ICIP)*, pp. 3468–3472, IEEE, 2018.
- [44] Y. Wang, X. J. Shen, H. P. Chen, and J. X. Sun, "Action recognition in videos with spatio-temporal fusion 3D convolutional neural networks," *Pattern Recognition and Image Analysis*, vol. 31, no. 3, pp. 580–587, 2021.
- [45] J. Li, H. Zhang, W. Wan, and J. Sun, "Two-class 3D-CNN classifiers combination for video copy detection," *Multimedia Tools and Applications*, vol. 79, no. 7–8, pp. 4749–4761, 2020.
- [46] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *IEEE International Conference on Computer Vision*, pp. 4489–4497, IEEE, 2015.
- [47] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1003–1012, IEEE, 2017.
- [48] Y. Liu, H. Dong, X. Wang, and S. Han, "Time series prediction based on temporal convolutional network," in *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)*, IEEE, 2019.
- [49] Y. Zhang, Y. Ma, and Y. Liu, "Convolution-bidirectional temporal convolutional network for protein secondary structure prediction," *IEEE Access*, vol. 10, pp. 117469–117476, 2022.
- [50] J. Zhang, C. Liu, and L. Ge, "Short-term load forecasting model of electric vehicle charging load based on MCCNN-TCN," *Energies*, vol. 15, no. 7, Article ID 2633, 2022.
- [51] C. Guo, X. Kang, J. Xiong, and J. Wu, "A new time series forecasting model based on complete ensemble empirical mode decomposition with adaptive noise and temporal convolutional network," *Neural Processing Letters*, vol. 55, pp. 4397–4417, 2022.
- [52] P. Huang, C. Wen, Z. Li, Y. Yang, and Q. Peng, "A neural network model for realtime prediction of high-speed railway delays," *Chinese Safety Science Journal*, vol. 29, no. S1, pp. 20–26, 2019.
- [53] Y. Sun, T. Zhang, N. Liu, Y. Zhao, and A. Li, "Analysis of factors causing high speed railway train delay based on improved ISM-DEMATEL," *Railway Transport and Economy*, vol. 44, no. 3, pp. 1–6, 2022.
- [54] T. Tang and J. Gan, "A train running time prediction model based on domestic and foreign railway operation data," *China Safety Science Journal*, vol. 32, no. 6, pp. 123–130, 2022.
- [55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.
- [56] X. J. Shi, Z. R. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, "Convolutional LSTM network: a machine learning approach for precipitation nowcasting," in *29th Annual*

- Conference on Neural Information Processing Systems (NIPS)*, MIT Press, 2015.
- [57] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014.
- [58] J. Liu, T. Zhang, Y. Gou, X. Wang, B. Li, and W. Guan, "Convolutional LSTM networks for seawater temperature prediction," in *2019 IEEE International Conference on Signal, Information and Data Processing*, IEEE, 2019.
- [59] C. J. Mantas, J. G. Castellano, S. Moral-García, and J. Abellán, "A comparison of random forest based algorithms: random credal random forest versus oblique random forest," *Soft Computing*, vol. 23, no. 21, pp. 10739–10754, 2019.
- [60] Y. Pan, W. Zhai, W. Gao, and X. Shen, "If-SVM: iterative factoring support vector machine," *Multimedia Tools and Applications*, vol. 79, no. 35-36, pp. 25441–25461, 2020.