Intelligent Hazard-Risk Prediction Model for Train Control Systems

Jing Liu, Yan Zhang, Jiazhen Han, Jifeng He, Junfeng Sun, and Tingliang Zhou

Abstract—**Although there has been substantial research in system analytics for risk assessment in traditional methods, little work has been done for safety risk prediction in communication-based train control (CBTC) system, especially intelligently predicting risk caused by the uncertainty in the system operation. Risk prediction and assessment of hazards in train control systems are vital for the safety and efficiency of urban rail transit. In this paper, we propose an intelligent hazard-risk prediction model based on a deep recurrent neural network for a new communication-mode CBTC system. First, a train-to-train communication-based train control (T2T-CBTC) system is proposed to improve the drawback of CBTC in information-exchanging mode. Then we design a risk prediction feature selection and generation method and estimate a critical function feature in the T2T-CBTC system by statistical model checking. Finally, we construct our intelligent hazard-risk prediction model based on a deep recurrent neural network using a long-short-term memory (LSTM) network. The model had excellent risk prediction classification results and performance in our experiment, even for unbalanced data set. This model consistently outperforms the deep belief network trained in Accuracy, Precision, Recall and F1-score for the hazard-risk prediction problem. Specifically, the mean accuracy is 97.2% and mean F1-score is 93.9% in overall performance of model. The improvements of our model against DBN model are 8.2% for Precision, 7% for Recall and 8% for F1-score.**

Index Terms—**Risk prediction, deep learning, long-short-term memory (LSTM), communication-based train control system, statistical model checking.**

I. Introduction

URBAN rail transit systems have become key transportation systems in Chinese cities. A safe and reliable train control system is necessary to ensure their smooth operation. As the latest-generation automatic train control system, the communications-based train control (CBTC) system is widely used in urban rail transit systems [1], [2]. However, there is no denying that CBTC is a safety-critical system. When a system failure occurs, the consequences are serious. Historically, collisions have had the most serious consequences in accidents on urban rail transit systems. The most critical factor in this is movement authority (MA) calculation failure, which provides the wrong control commands to trains and wayside equipment. MA is the authority for a train to enter and travel through a specific section of track, in a given travel direction. Movement authorities are assigned, supervised, and enforced by a CBTC system to maintain safe train separation and to provide protection through interlockings. The correctness of MA plays a vital role in avoiding collisions.

Risk assessment is a technique for identifying the operational safety of a system using either qualitative or quantitative methods. Hazard-risk assessment is a critical requirement of a CBTC system. Many approaches to hazard-risk assessment are employed during the design of a system, such as fault tree analysis and preliminary hazard analysis. Such approaches probably cannot give satisfactory results, since they neglect information about the uncertainty and dynamic nature of train operation. Therefore, to guarantee high safety in an urban rail transit system, it is important to predict the hazard risk of a CBTC system.

A traditional CBTC system only has a train-to-ground communication mode, and the critical control functions are concentrated in the wayside equipment. If adjacent trains are on the same track, there is a need to exchange information through the wayside equipment, which not only greatly reduces operational efficiency, but can lead to hazards from the loss of information and delays due to excessive system interaction.

With the development of artificial neural networks, deep learning is the most popular method for functions such as data-feature learning, time-series data prediction, and image recognition. Deep neural networks (DNNs) can be effective in learning features from data in an unsupervised manner without prior knowledge [3], [4]. In addition, DNNs are well established in traffic-flow prediction [5], [6], automatic-driving fault prediction [7], [8], and track-circuit fault prediction [9]. Therefore, neural networks are an interesting option in hazard-risk prediction problems.

This paper mainly completes the following work.

1) We put forward a train-to-train communication CBTC system architecture and analyze the MA function model. In this paper, we redistribute the system key functions and reduce the track-side equipment for the new system, which greatly improves the utilization rate of the system. We also analyze the new information flow for MA calculation in the system.

Manuscript received October 10, 2018; revised February 26, 2019, May 26, 2019, and September 7, 2019; accepted September 12, 2019. This work was supported by the National Key Research and Development Project under Grant 2017YFB1001800 and in part by NSFC under Grant 61972150 and Grant 61572195. The Associate Editor for this article was M. Brackstone. *(Corresponding author: Jifeng He.)*

J. Liu and J. He are with the Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China (e-mail: jifeng@sei.ecnu.edu.cn).

Y. Zhang is with China Unionpay, Shanghai 200136, China.

J. Han is with the Department of Computer Engineering, New York University, New York, NY 10003 USA.

J. Sun and T. Zhou are with CASCO Signal Ltd., Shanghai 200071, China.

Digital Object Identifier 10.1109/TITS.2019.2945333

2) We propose an intelligent hazard-risk predictive model for our train-to-train communication CBTC system. The predictive model is implemented by a deep recurrent neural network (RNN) called a long-short-term memory (LSTM) network, which takes into account some pivotal system parameters. Specifically, the LSTM network is trained to predict the occurrence probability of a hazard from datasets and classify them. The model's classification results can provide guidance for the operation of urban rail transit systems.
3) We design a risk prediction feature selection method and perform MA calculation failure estimation based on statistical model checking (SMC). By building the system function model and describing a safety property, we use simulation-based techniques to solve the probability of MA calculation failure.

The rest of this paper is structured as follows. Section II reviews related studies on safety-risk assessment for some control systems. Section III provides the main method of our paper. Section IV explains the T2T-CBTC system architecture and MA calculation model in the new system architecture. Our intelligent risk predictive model is described in section IV. Section V presents the experiment and result. We discuss future work in section VII.

II. Related Work

Hazard risk analysis has been long considered a key functional component in industrial control systems. Over the past few decades, this problem has been substantially investigated to assist in urban rail transit management and control to improve transportation efficiency. At early as the 2000s, quantified risk assessment approaches, such as fault tree analysis (FTA), event tree analysis (ETA), and failure mode and effects analysis (FMEA), were used to analyze railway traffic system safety, and a variety of models have been proposed by researchers [10].

Z. Youpeng *et al.* illustrated and analyzed the interval signal control function for train control center using the fuzzy-failure mode, effects and criticality analysis (FMECA) method, which employs FMECA to abstract the potential failure modes in such functions, and uses fuzzy analytical hierarchy process (FAHP) to determine risk weights [11]. Taking the external door control function as an example, Mouna Rekik *et al.* discussed the network attack security problem faced by railway system, and made an assessment for the impact of potential hazards [12]. In addition, A fuzzy risk assessment method based on fuzzy analytic hierarchy process and integrating other methods is proposed by Andrew John *et al.*. Taking seaport operation as the background, this method established a system cooperation model, determined the weight of uncertain risk factors, and carried out risk analysis [13] Kuo-Sui. L *et al.* solved the failure mode problem of safety critical system in fuzzy and uncertain environment, making up for the deficiency of traditional FMEA method [14].

The formal method is mainly used for system safety verification, and it focuses on safety-critical applications. Mathieu Comptier *et al.* used formal proofs to analyze the safety of the Octys CBTC system interlocking infrastructure. They modeled and verified it with the Atelier B tool [15]. Graziana C *et al.* summarized the application of Petri Net in the field of freight Logistics and transportation. They classified several commonly used analytical methods, and discussed the feasibility and future research direction [16]. Hindolo George W *et al.* proposed a hybrid modeling method for nuclear power plant to understand system risk, integrating Monte Carlo methods, multistate modeling, and network theory [17].

Some artificial intelligence methods are also applied to solve the collision prediction problem. S. Nefti and M. Oussalah proposed an artificial neural network (ANN) architecture to deal with the prediction of system faults. Taking irregularities in the positioning of rails as input and using a wavelet transformation technique to reduce the dimensionality, the ANN can predict the safety ratio of the rails. They determined the best ANN structure to predict railway safety and evaluate performance [18]. Deep learning algorithms were proposed in 2006 [19], [20], and much research was published on the basis of it. Huang *et al.* [5] proposed a deep architecture to predict traffic flow, which is a multitask regression DBN to incorporate multitask learning (MTL). In addition, to make MTL more effective, the weights in the top layer were grouped to improve the experimental results. Jinyong Wang and Ce Zhang utilized a deep learning model in software reliability fault-prediction. This model uses a RNN encoder–decoder. Comparison with existing models showed that their model had better prediction performance [21].

Quantitative analysis approaches are mainly based on Boolean values to determine an event, and at the same time, they depend on an expert system, so they lack objectivity and real-time results. Methods such as FHA are mainly used in the system design phase, and the uncertainty caused by CBTC in actual operation is not considered. Formal methods can only verify safety and cannot give a predictive value with high reliability. ANN only has three layers and uses a back-propagation algorithm to train itself to more effectively predict system safety, but it has difficulty in expressing complex data features.

This paper is based on a paper we published [22]. The paper used deep belief network (DBN) as prediction model to solve the safety prediction problem of rail transit systems. We analyzed system safety risks from another perspective using LSTM networks instead of DBNs getting better results. In addition, we have also proposed many works on system safety modeling and formal method. For the purpose of optimizing formal verification problems in CBTC system, [23] proposed two projection methods based on Problem Frames and constraints. [24] proposed a method to get safety requirements for interlocking systems in CBTC, and formally describe them by classifying and developing safety requirements specification patterns. Taking a subsystem of CBTC as an example, [25] used the slicing criteria to simplify the modeling process of SCADE, thus reducing the state space of complex system modeling.

III. Methodology

Our overall research goal is to build a prediction model that takes some risk-influencing features as input and produces

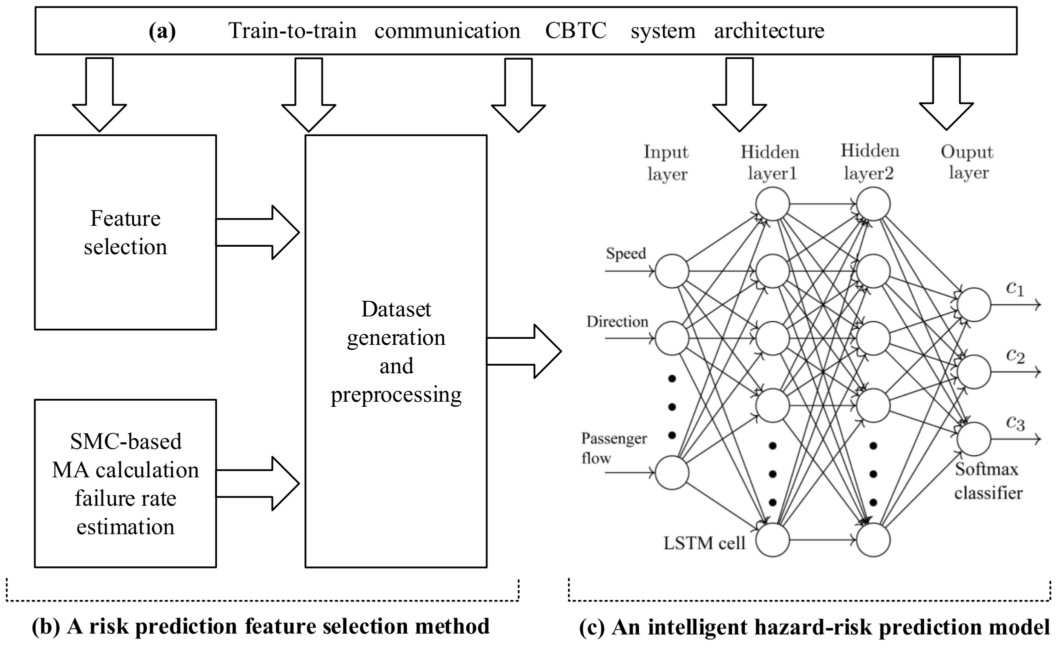

(a) Train-to-train communication CBTC system architecture
Feature selection
SMC-based MA calculation failure rate estimation
Dataset generation and preprocessing
Input layer
Hidden layer1
Hidden layer2
Ouput layer
Speed
Direction
Passenger flow
LSTM cell
c_1
c_2
c_3
Softmax classifier
(b) A risk prediction feature selection method
(c) An intelligent hazard-risk prediction model

Fig. 1. Schematic diagram of our proposed method that includes three processing steps: (a) put forward a train-to-train communication CBTC system architecture; (b) design a risk prediction feature selection method; (c) propose an intelligent hazard-risk prediction model for our train-to-train communication CBTC system.

a hazard state classification. Fig. 1 is the schematic diagram of our proposed method based on a deep neural network for the T2T-CBTC hazard risk prediction.

According to the regulations in IEEE standard 1474.1-2004 [26], for CBTC systems, it is necessary to establish a system safety program to identify hazards and prevent accidents. In this paper, we only consider the hazard of collision between two trains, especially the rear-end collision, whose main influencing factors are train separation, traffic direction, and route interlocking. We define three states of collision between two trains:

- No collision (c_1): The train is under normal operation and it will not suffer a collision.
- Slight collision (c_2): The train has a slight collision without casualties.
- Severe collision (c_3): The train body is damaged and there are casualties.

The hazard risk (HR) we defined is the likelihood of collision state:

$$HR = Prob(y = c_i | i = 1, 2, 3). \tag{1}$$

The occurrence of a train collision is related to the train's state parameters during operation. The relevant state variables during a collision at time T are different from a normal state, and the state variables before and after a collision have some correlation. Therefore, it is necessary to determine whether the system would have a collision event based on the sequence of state parameters. Our samples $\mathbf{X}$ are presented in the form of a sequence, where $\mathbf{x}(i)$ is a sample at time i:

$$\mathbf{X} = \{\mathbf{x}(1), \mathbf{x}(2), \ldots \mathbf{x}(t)\} \tag{2}$$

As Fig. 1 shows, through research and analysis of the CBTC system, we first proposed the overall architecture of train-to-train communication based train control(T2T-CBTC) system, by functional redistribution and equipment simplification. We then analyzed key functions of the train operation, MA calculation, the generation process under the T2T-CBTC system, and the system-interaction data flow. In this context, we proposed a deep neural network-based intelligent hazard-risk prediction model for the T2T-CBTC system.

After proposing T2T-CBTC system and reviewing published research [26]–[28], we built a set of prediction feature that affect the safety of CBTC. Features that are most correlated with hazard risk were chosen. Meanwhile, for the MA calculation failure probability, we used the statistic model checking (SMC) method to calculate. Dataset generation and processing included normalizing them and dividing them into training and test datasets, the former to train DNN and the latter to participate in prediction. After that, DNN construction was carried out. Two hidden layers used LSTM network and *softmax* classifier as the output layer. Once the training was over, the test dataset was fed to the trained DNN to forecast the hazard-risk class, and we analyzed the experimental results.

IV. Train-To-Train Communication CBTC System Architecture

A. Train-To-Train CBTC System

The CBTC system is the latest generation of automatic train control systems utilizing high-resolution train location determination, independent of track circuits; continuous, high-capacity, bidirectional train-to-wayside data communications;

and trainborne and wayside processors capable of implementing vital functions [26]. The CBTC system consists of a station system, track-side system, and onboard system. The station control system includes an automatic train supervision system (ATS), data storage unit (DSU) subsystem, and maintenance center. The trackside system includes zone controller (ZC), computer-based interlocking system (CBI), and trackside wireless access point (AP). The onboard system takes charge of automatic train protection (ATP) and automatic train operation (ATO). Unlike previous-generation train control systems, CBTC uses moving blocks instead of fixed blocks to protect trains. The term "block" refers to a section of track that is processed by a ZC. In the fixed blocks, the track is divided into a plurality of fixed-occlusion sections by some fixed logic points. There can be at most one train in a fixed interval, resulting in a large waste of track resources. In contrast, moving block technology cancels these dividing points and assigns a safe driving area in front of the train as an occlusion interval, according to the train state and other related equipment on the track in each calculation cycle. This improves the utilization of the track and reduces the waste of resources.

From the above structural analysis, it can be seen that the CBTC system does not establish direct communication between trains, and there are many problems in information interaction and long communication delays. On the one hand, the interaction processes is indirect, and data flow is from the front train, to the trackside equipment, and then to the current train. If any of these links failed, the system would not work properly. On the other hand, the core function of train control is located on the trackside equipment, the ZC, which is too concentrated for information processing. If the ZC failed, it would affect the operation of multiple trains within the zone.

A T2T-CBTC system eliminates ZC and CBI equipment, integrates their functions into VOBC, and adds a vital system database (VSDB) to store train and track informations. For each subsystem, main changes are as follows:

- For station system, VSDB is added, storing information such as track and train data.
- For trackside system, the ZC and CBI have been canceled, and the track system controller (TSC) has been added to retain some functions of ZC, which only has the basic functions of monitoring train operation and storing information of forwarding train data.
- For onboard system, the importance and intelligence of it have been promoted. Functions includes onboard movement authority (OBMA), onboard computer interlocking (OBCI), and onboard management controller (OBMC) have been added. By itself, VOBC accomplishes the detection of obstacles in front, the query of route information, the collaborative completion of MA calculation, and realizes the train as the core design principle.

The T2T-CBTC system and the information interaction between subsystems is shown in Fig. 2. As shown in Fig. 2, the VSDB is responsible for storing track information, receiving and storing obstacle information, and providing static obstacle information for the OBMA. VSDB is also responsible for the registration of new trains. TSC is used to control, record, and provide block information, and it is responsible for establishing communication with trains entering its zone of jurisdiction, which can avoid preemption of block resources by different OBCIs. The VOBC system adds OBCI, OBMC, and OBMA modules. OBCI is responsible for route management, planning and execution, and management of branch information, and it provides access information for other modules in VOBC. OBMC is the core function of VOBC, and the key realization of train-to-train communication. It is responsible for communication with the front train, realizing the exchange of data with other trains on the route, information management of the current train, train identification, train communication maintenance, and fault status processing. OBMA is the implementation of the MA function in the original ZC system. According to various information provided by OBCI, OBMC, and VSDB, MA is generated for the train, which is a key function to ensure operational safety.

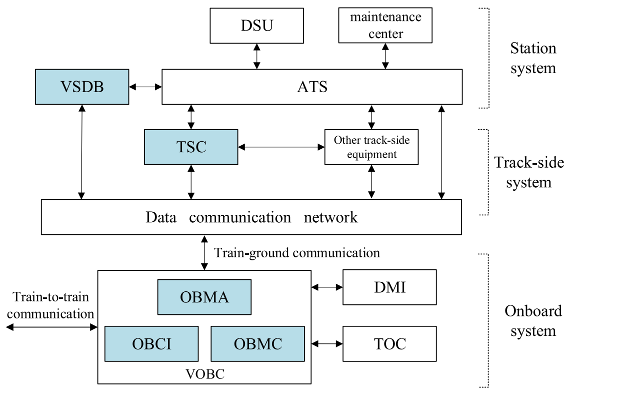

DSU
maintenance center
Station system
VSDB
ATS
TSC
Other track-side equipment
Track-side system
Data communication network
Train-ground communication
OBMA
DMI
Train-to-train communication
OBCI
OBMC
TOC
VOBC
Onboard system

Fig. 2. T2T-CBTC system architecture. The blue blocks represent the new functional modules in the train control system.

B. MA Calculation Model

The MA function is considered as a basic and most crucial function to prevent trains from colliding with each other. MA is a direct condition for determining the distance traveled by a train, and it is also a prerequisite for realizing train speed control. The final role is to provide relevant information for train speed protection in ATP. Generally, taking the train head as the starting location reference point along the authorized path, the system considers the current state of the relevant equipment on the track and calculate MA. The calculation results will be transmitted to the ATP and ATO to adjust the train driving behavior. The calculation algorithm must determine an endpoint of the safety block, which denotes that the track area from the current position of a train to this endpoint is safe for movement, and no other train can enter this region at the same time. This endpoint is called the end of authority (EOA). The information flow of MA calculation function is shown in Fig. 3.

The main interactions with OBMA are OBCI, OBMC, and VSDB. The information required by the calculation algorithm includes the position, speed, and direction of the front train, and the current train information from OBMC, track and block, static obstacle information from the VSDB, and route

TABLE I
LIST OF HAZARD-RISK PREDICTION FEATURES

Category	Feature Variables	Range	Description
Facilities	Communication delay	0.5 s $\sim$ 2 s	Train-to-train message communication delays
	Maximum number of trains	10 $\sim$ 40 trains	Maximum number of trains in a block zone
	Train location accuracy	$\pm$ 5 m $\sim$ $\pm$ 10 m	Accuracy of measured train location
Equipment	Train speed	$\pm$ 0.5 km/h $\sim$ $\pm$ 80 km/h	Train speed in operation
	Train direction	Up or Down	Train direction in operation
	Block direction	Up or Down	Block direction in configuration
	Reaction times	0.07 s $\sim$ 0.75 s	System equipment reaction times
People	Passenger flow	3 $\sim$ 4 w/h	Passenger flow per hour
	Working time	0 h $\sim$ 8 h	Hours a driver has been working continuously
	Driver number	3 $\sim$ 6	Number of divers
	Operation type	Auto or manual	Operation type of train
Procedures	MA calculation failure rate	$< 10^{-8}$	The probability of MA calculation error
	MA calculation time	0.07 s $\sim$ 1 s	MA calculation time cycle

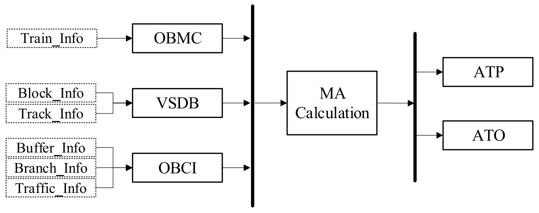


Fig. 3. Information flow in OBMA model for MA calculation.

and branch information from the OBCI. The train requests communication with the front train, and the front train sends the position, speed, and direction information to the following train; at the same time, the train requests data from OBCI and VSDB. After that, OBMA calculates the MA, which is sent directly to the ATP and ATO to guide the train operation. Trains can only passively move based on the MA. The purpose of the MA is to enable adjacent trains to be safely separated. Once MA is at fault, it may lead to a collision, with catastrophic consequences.

Calculation of EOA is the core algorithm in MA. MA calculation scenario simulation must be realized by establishing a CAL_EOA module. For this paper, we built a CAL_EOA module in MATLAB/Simulink in our preliminary work. This will be used to solve for the MA calculation failure probability.

V. A Feature-Selection Method for Risk Prediction

A. Feature Selection

In a practical train-operation environment, there are so many collision-influencing factors that it is too complicated to account for them all to forecast a train collision. According to IEEE Standard 1474.1 [26], in terms of the major elements that make up the system and its environment, prediction features should include four categories: *equipment, facilities, procedures, and people*. For the hazard we considered above, we selected some features that influence it the most. They are listed in Table I, with category, range values and description. Note that some features have different measurement units. Some features have units such as the train speed, whereas others, e.g., train direction and operation type, cannot be given a unit since they are category-based. These can be solved during data preprocessing.

For procedures, we have specifically chosen MA-related factors. MA function is the most influential factor associated with collision events. The most likely cause of a train collision is that a train has wrong information about the end of the current travel range, which results in an incorrect speed-control command. The calculation of the end of the current travel range, i.e., EOA, is determined by the MA. Therefore, at each sampling point, once the MA calculation is in error, the train in this state must have an accident. However, it is not easy to judge whether the current MA is wrong, so it is better to use the MA calculation failure rate under the current system parameters. Moreover, the time of MA calculation is another feature for prediction. Too long calculation time leads to slowler update of information and error in EOA calculation, which easily leads to the occurrence of collision events. We proposed a verification method for MA calculation failure based on SMC for rare events, which is described in next section.

In summary, the input vector of the risk-prediction model at t can be expressed as

$$F(t) = \{f_i | i = 1, 2, \ldots, n\}, \tag{3}$$

where i is the dimension of the input vector, and the f_i are prediction features.

B. Estimation of MA Calculation Failure

Generation of a wrong MA is a rare event in T2T-CBTC, but it has a great impact on the generation of collisions. Rare events are those that occur with low frequency, such as less than 10^{-8}, whose occurrence could have a substantial negative impact. According to the IEEE standard, the system that calculates MA has the highest level of safety in the CBTC, and the system generates uncertainty during operation, so a rigorous and effective method is needed to evaluate the safety of the system. The formal method is a way of specifying a description, model reasoning, and verification using formal language. It combines the knowledge of mathematical logic, programming language, hardware design, and theoretical computer science, and is widely used in the verification of hardware and software in industry. In this paper, we used a statistical model checking (SMC) algorithm based on importance sampling (IS), a kind of formal method to estimate the MA failure rate [29]. We called it an IS-SMC algorithm. We constructed a CAL_EOA model in MATLAB, simulated the MA scenario, and obtained a rare-event occurrence probability by means of the IS-SMC algorithm.

SMC is a simulation-based model-checking approach to verify properties specified in a temporal logic [30]. SMC needs a system model and a requirement specification. The system model is an abstract description of system behavior. Sampling executions trace a system model at first. After sampling, statistical inference can be applied. Estimation is a common statistical inference, which is used to determine the probabilities of system parameters. Temporal logic is a logical language for describing system requirements in formal verification methods. Describing the requirements with temporal logic can well verify the requirement formula during system operation.

Before estimation, we should formally define this problem.

Definition 1: Given a CAL_EOA model M and a safety requirement property ψ, statistical model checking is to estimate the probability p that M satisfies ψ, and it is greater than or equal to a precision parameter θ, i.e., $M \models P_{\geq\theta}(\psi)$.

The safety requirement we considered is: *MA calculation failure does not occur*. Considering the simple case, five key sensors are involved in the calculation of the MA. As long as the value of each sensor is guaranteed to be correct at each moment, the MA calculation will not fail. We describe the requirement property ψ utilizing bounded linear temporal logic (BLTL)

$$\psi = F^{100}G^{1}(SysOutError = 0). \tag{4}$$

Operator F and G are temporal connectives. F means some future state. G means all future states. Formula (4) states that within 100 cycles, at any moment, system failures due to unreliable sensor parameters and randomness of the system do not occur, where each $SysOutError$ follows the Bernoulli distribution.

For the purpose of the probability, we use random sampling of system execution paths. However, it is a challenge for this purpose to use a classical SMC algorithm based on simulation. The main reason is that to obtain samples with low probability, the number of system simulations would be at an exponential level, which would require much time, and would greatly reduce the algorithm's efficiency. Unlike classical statistical model checking, the IS-SMC is merged with importance sampling and a cross-entropy method to reduce the sample state space.

IS-SMC is based on the Monte Carlo method, which generates N_{MC} random simulation sequences $\sigma_1, \ldots, \sigma_n$, following a Bernoulli distribution, and computes the probability $\hat{\rho}$ as follows:

$$\begin{aligned}\hat{\rho} &= E[B(\sigma)]\\ &= \frac{1}{N_{MC}}\sum_{i=1}^{N_{MC}} B(\sigma_i \models \psi)\end{aligned} \tag{5}$$

where $E[\cdot]$ is expectation function, B is an indicator function that returns 1 when ψ is satisfied in σ_i, and 0 otherwise. With a sufficiently large sample, the Bernoulli distribution will be close to a normal distribution, whose variance is

$$D(\sigma) = N_{MC}\hat{\rho}(1-\hat{\rho}). \tag{6}$$

However, when $\hat{\rho}$ is much lower, to keep the variance as low as possible, a larger sample space is required.

Importance sampling is an effective technique to reduce the sample space in the application of SMC [31]. Using weighted system simulations, it is possible to realize a rare event. Importance sampling starts by introducing a weighting function $W(\sigma)$ on the observed random variables, without changing their expectation $E[B(\sigma)]$, and reducing their variance.

$$\begin{aligned}\rho &= E[B(\sigma)]\\ &= \int B(\sigma)f(\sigma)d\sigma\\ &= \int B(\sigma)W(\sigma)f_*(\sigma)d\sigma\\ &= E_*[B(\sigma)W(\sigma)]\end{aligned} \tag{7}$$

The weighting function is

$$W(\sigma) = \frac{f(\sigma)}{f_*(\sigma)}, \tag{8}$$

where $f(\sigma)$ is the probability density function. $W(\sigma)$ makes the $E[B(\sigma)]$ with $f(\sigma)$ become the $E_*[B(\sigma)W(\sigma)]$ with $f_*(\sigma)$. $f_*(\sigma)$ is a probability density function that is suitable for sampling, so that the number of simulation samples can be reduced. The ρ-based IS-SMC is

$$\rho_{IS} = \frac{1}{N_{IS}}\sum_{i=1}^{N_{IS}} B(\sigma_i)W(\sigma_i). \tag{9}$$

Therefore, finding a good weighting function distribution $f_*(\sigma)$ is the crucial problem, and $N_{IS} \ll N_{MC}$.

Actually, $f_*(\sigma)$ is dependent on ρ, and we should find the most proximal distributions of $f_*(\sigma)$ by checking members of a parameterized family of distributions $f(\cdot,\tau)$. The cross-entropy (CE) method can select the appropriate members that minimize Kullback-Leibler divergence $CE(f_*(x), f(x,\tau))$ from the optimal biasing, through sampling from the original unbiased distribution. Therefore, the optimal parameter τ_* can be determined by the following:

$$\begin{aligned}\tau_* &= argmin\ CE(f_*(x), f(x,\tau))\\ &= argmin \int f_*(x)\log f_*(x) - \int f_*(x)\log f(x,\tau_*)dx\\ &= argmax \int f_*(x)\log f(x,\tau)dx\\ &= argmax \int b(x)W(x,w)f(x,w)\log f(x,\tau)dx\\ &= argmax\ E_w[b(x)\log f(x,\tau)],\end{aligned} \tag{10}$$

and we introduce another weighted function,

$$W(\sigma, w) = \frac{f(\sigma)}{f(\sigma, w)}, \tag{11}$$

where w is a tilting parameter that can be initialized in the range $0.1 \sim 0.01$. Therefore, we can obtain the parameters τ_* by sampling the system N_{CE} times with the probability density $f(\sigma, w)$.

The pseudocode of the IS-SMC algorithm is as follows.

Algorithm 1 SMC Algorithm Based on IS

```
Input: Model $M$, Safety requirement property $\psi$
Output: The property $\rho$ of $M \models \psi$
 1: $sat_{CE} = 0$, $sat_{IS} = 0$
 2: for $i = 1$ to $N_{CE}$ do            ▷ Simulation for parameters $\tau_*$
 3:     $\sigma$ = Simulate model $M$ with $f(x, w)$
 4:     if $\sigma \models \psi$ then
 5:         $sat_{CE}$++
 6:     end if
 7: end for
 8: According to the formula (10) to get $\tau_*$ and $f_(x, \tau_*)$
 9: for $j = 1$ to $N_{IS}$ do            ▷ Simulation for $\rho$
10:     $\sigma$ = Simulate model $M$ with $f(x, \tau_*)$
11:     if $\sigma \models \psi$ then
12:         $sat_{IS}$++
13:     end if
14: end for
15: According to the formula (9) to get probability $\rho$
```

VI. An Intelligent Risk-Prediction Model Using Deep RNN

A deep neural network (DNN) is a form of deep learning that provides modeling for complex nonlinear systems. The breakthrough application of DNN in speech-recognition and image-recognition has led to explosive growth in its application [32], [33]. For hazard-risk prediction, we hope to use DNN's powerful learning ability to find correlations and dependencies between data. In this section, we describe our proposed intelligent hazard-risk prediction model using DNN.

A. Network Architecture

To handle time correlation, RNN is a good choice. Due to its unique loop structure, it can remember past memory. RNNs, as a type of artificial neural network, have loops in each cell that hold a continuous information stream. They can recognize and deal with sequences of data, especially time-series data. Nevertheless, RNNs suffer from the vanishing-gradient problem, which occurs when the gradient shrinks quickly during back-propagation. A long short-term memory (LSTM) cell can solve this problem well. LSTM is a variant RNN that learns long-term and short-term dependencies, which is proposed by Hochreiter and Schmidhuber in 1997 [34]. It mainly deals with random length sequence data. It has good application in unsupervised time series classification, sentiment analysis and activity recognition [35], [36].

An intelligent safety-risk prediction model is created by using LSTM-RNN to form a DNN. More clearly, by stacking some LSTM-RNN layers to form a deep recurrent neural network and using a softmax unit at the top layer for classification in our model, we can easily perform supervised fine-tuning on the whole model.

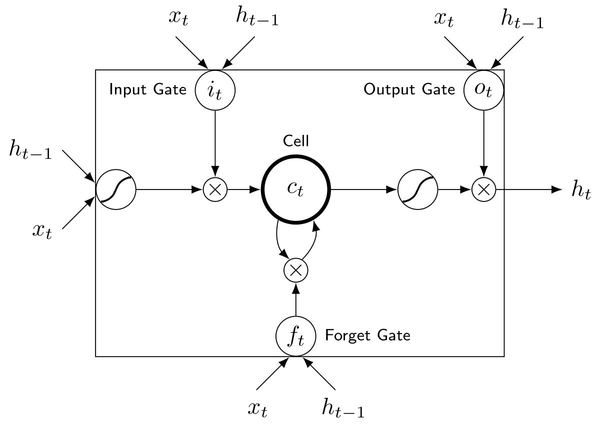


Fig. 4. Structure of the LSTM cell.

1) LSTM Cell and Deep RNN: The LSTM cell structure is shown in Fig. 4. Unlike RNNs, in addition to the external basic loop, it also has a loop of internal LSTM memory cells, i.e., a self-loop. This can keep the cell state and maintain the information flow unchanged, as shown by the cell c_t in Fig. 4. This is the key to solving the vanishing-gradient problem with LSTM. Eq. (16) shows that the c_t between adjacent moments will only perform some minor linear interactions. Moreover, the LSTM cell structure adds several gates to control the influence scale of information about the last moment, can achieve control, and can learn long-term memory, like input gate, forget gate, and output gate. i_t is the *input gate*. It determines how much of the input information at the current time is saved to the LSTM memory cell. f_t is the *forget gate*, which determines how much of the LSTM memory cell c_{t-1} at the previous moment can be retained until the current moment. o_t is the *output gate*, which has a decision on output of useful information. The equations that describe these relationships are

$$\mathbf{f}_t = sigm(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{12}$$
$$\mathbf{i}_t = sigm(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{13}$$
$$\mathbf{o}_t = sigm(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{14}$$
$$\tilde{\mathbf{C}}_t = tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C) \tag{15}$$
$$\mathbf{C}_t = \mathbf{f}_t * \mathbf{C}_{t-1} + \mathbf{i}_t * \tilde{\mathbf{C}}_t \tag{16}$$
$$\mathbf{h}_t = \mathbf{o}_t * tanh(\mathbf{C}_t) \tag{17}$$

In our method, we use a deep LSTM network architecture, where the model has one input layer, one output layer, and two LSTM layers as a hidden layer, each with 256 LSTM units, and a learning rate of 0.07. Such a network architecture is based on much empirical research [37]. Too many or too few hidden layers would make the fitting degree less than optimal, and the proper number of layers and units can achieve the best results. At the same time, for each hidden layer, we set the dropout rate to 0.2, which is inspired by the research of Srivastava, Hinton *et al.* [38]. To prevent the network model from overfitting, some units do not work with a certain probability during the training process.

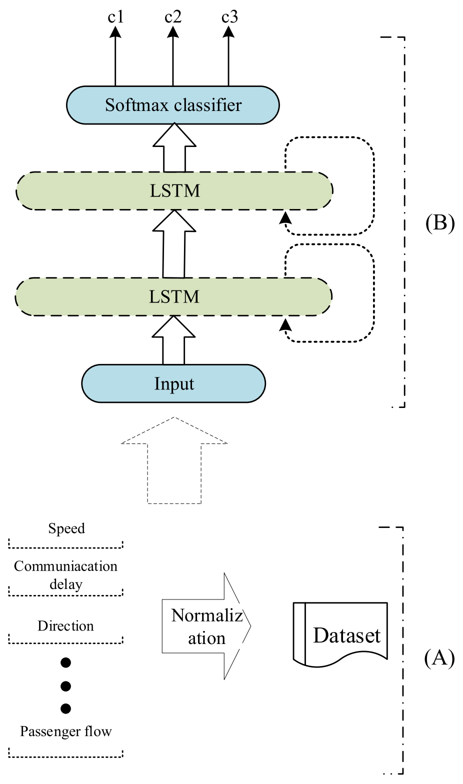


Fig. 5. Intelligent hazard risk prediction model architecture based on LSTM.

2) Input and Output: Corresponding to Table I, there are 13 input units in the first layer. Therefore, the inputs of the network are

$$\mathbf{x}(t) = [f_1(t), f_2(t), \ldots, f_{13}(t)]. \tag{18}$$

The output of the network is the collision type of train-operation status described in section III-B. The output of the last layer is used as the input of this layer. The output $\mathbf{O}$ can be denoted as

$$\mathbf{O} = \{o_0, o_1, o_2\}, \tag{19}$$

where o_0 refers to a normal state and no collision, o_1 refers to slight collision, and o_2 means a severe collision. The outputs of the last LSTM hidden layer are the inputs to the *softmax* output layer. They give the probability of each category o_k as

$$P(Y = o_k) = \frac{e^{sigm(\mathbf{w}_k \mathbf{v} + \mathbf{a}_k)}}{\sum_{k=0}^{2} e^{sigm(\mathbf{w}_k \mathbf{v} + \mathbf{a}_k)}}. \tag{20}$$

This architecture of the intelligent hazard-risk prediction model is illustrated in Fig. 5. (A) is dataset generation. Data from feature were normalized to generate dataset. (B) is network classification process. The two hidden layers are LSTM networks and softmax classifier realized three classifications.

B. Network Training

1) Preparation of Dataset: We divided datasets into three parts, consisting of training, validation, and test datasets.

We used a combination of collection and simulation to generate a dataset. Some of data were collected from DSU system, and some by sampling according to [26]. The DSU stores the used line information and configuration file information of each subsystem in the CBTC system. Data from the DSU were recorded and updated continuously while a train was operating. System logs are also an effective source of data.

There are some problems in the dataset, such that the variable units and range are inconsistent. To better understand the relationship between the data and eliminate the effects of differences in the range of values between variables for the predictive model, some preprocessing steps are needed for these data to match our model. This paper uses standardization to preprocess the datasets. This can ensure that all data values are in a specified interval. The result is a standard Gaussian, or standard normal distribution.

2) Training Algorithm: Back-propagation through time (BPTT) is a classic algorithm for training RNN, presented by P.J. Werbos in 1991 [39]. The loss in our paper is defined as the cross entropy loss and add L2 regularization optimization, given by

$$loss(t) = -log(P(Y(t) = o_k)) + \lambda \sum_i w^2. \tag{21}$$

λ is regularization parameter to control intensity and w is weight. The role of regularization is to prevent model overfitting by modifying the network's training performance function, and improve the generalization ability of the model. The training goal of the network is to minimize the loss value.

$$min J(\theta) = \sum_t loss(t). \tag{22}$$

For each LSTM layer, the training algorithm is as follows. First, the sequence of input and output is provided to the network. Then we unroll the network by time steps and calculate $loss(t)$ across each time step. After that, we update the weight by Eq. (11) to Eq. (16) and roll up the layer. This procedure is repeated until all layers, except the output layer, are trained with their respective weights.

The output of the network is the classification of each sequence sample, and each softmax unit gives a probability value belonging to each category. An occurrence probability of a severe collision greater than 0.3 indicates that a serious collision will occur.

VII. Experiment and Results

In this experiment, we have three datasets, which are training, validation, and test datasets. These datasets respectively had 12825, 1425, and 750 samples. For each sequence, the properties of the system state and the properties of the hazard-risk are stochastically determined, and there are 100 observed points. This relates to a time period of 300 days. Our simulations were implemented on a Keras framework and MATLAB platform. For simplicity, we use c1,c2 and c3 to represent no collision, slight collision and severe collision respectively.

TABLE II
HAZARD RISK PREDICTION MODEL PERFORMANCE WITH DIFFERENT HIDDEN LAYER UNITS

Hidden layer units (hidden1+hidden2)	Accurancy	Precison	Recall	F1-score	Time
32 + 32	0.922	0.854	0.802	0.825	6.03
64 + 32	0.927	0.895	0.775	0.817	7.78
64 + 64	0.929	0.849	0.788	0.848	8.02
128 + 64	0.931	0.896	0.805	0.839	8.19
128 + 128	0.931	0.899	0.822	0.838	8.92
256 + 128	**0.936**	**0.864**	**0.866**	**0.860**	**8.78**
256 + 256	0.929	0.861	0.844	0.851	13.05
512 + 256	0.914	0.862	0.765	0.805	12.46
512 + 512	0.910	0.857	0.772	0.811	31.66

TABLE III
MODEL EVALUATION CRITERIA RESULT WITH DIFFERENT L2 PARAMETER

L2 parameter λ	Precision			Recall			F1-score		
	c1	c2	c3	c1	c2	c3	c1	c2	c3
0.02	0.920	0.928	0.958	0.986	0.828	0.534	0.956	0.875	0.696
0.015	0.920	0.934	0.961	0.986	0.815	0.586	0.956	0.870	0.739
0.01	0.936	0.923	0.945	0.986	0.847	0.637	0.965	0.883	0.778
0.003	0.951	0.964	0.833	0.980	0.853	0.775	0.970	0.905	0.803
0.001	**0.978**	**0.961**	**0.836**	**0.985**	**0.895**	**0.896**	**0.989**	**0.958**	**0.876**
0.0008	0.986	0.961	0.844	0.979	0.849	0.879	0.987	0.950	0.861

A. Model Evaluation Criteria

We quantified prediction performance of the model in the experiment by Accuracy, Precision, Recall and F1-score.

$$Accuracy = \frac{1}{n}\sum_{k=1}^{m}\mathbb{I}(f(\mathbf{x_k}) = y_k), \tag{23}$$

where $\mathbb{I}$ is an indicator function, y_k is the real value, and $f(\mathbf{x_k})$ is a prediction value.

$$Precision = \frac{TP}{TP + FP} \tag{24}$$

$$Recall = \frac{TP}{TP + FN} \tag{25}$$

$$F1 = \frac{2 \times TP}{sample\ size + TP - TN} \tag{26}$$

Precision, Recall and F1-score were based on the results of the confusion matrix, in which TP is true positive, FP is false positive, TN is true negative, and FN is false negative.

Precision focuses on the real correct sample rate in the forecast results, the recall is more concerned with how many positive class samples are recognized, even if there are some negative class samples. F1-score can consider comprehensively both precision and recall. It has a high value when the model is ideal. For the scenario in this paper, it is the *recall* that we focus on. For risk classification, we hope to find out all of them. Even if the normal class were recognized, it could be reanalysis by manual procedures. But if the risk class was missed, the consequences could be catastrophic. In experiments, we payed more attention to the Recall and F1-scores.

B. Network Architecture Parameter Discussion

We explored the impact of different hyper-parameter choices on model performance. We focused on tuning two important hyper-parameters: the number of units in hidden layer and the L2 regularization parameter λ. To do so, we fixed other parameter and varied two hyper-parameters. We concerned to observe these evaluation criteria. We chose to test with nine different hidden layer sizes and six different λ values. Table II and Table III show the results from the experiment. As can be seen, for hidden layer size, the setting where the number of units size is $256 + 128$ has the best result. Each of evaluation criteria is the highest in 256+128 units size, and the training time is acceptable. For L2 regularization parameter λ, the setting where $\lambda = 0.001$ has the highest value of recall and F1-socre in c2 and c3, which means this setting effectively avoids the impact of the unbalanced dataset. So, the setting we chose is the number of units size $= 256 + 128$ and $\lambda = 0.001$.

TABLE IV
CONFUSION MATRIX FOR THE HAZARD-PREDICTION TASK ON TEST DATASET WITH 500 SAMPLES

true / pred	c1	c2	c3
c1 (No Collision)	530	3	2
c2 (Slight Collision)	3	150	4
c3 (Severe Collision)	3	3	52

TABLE V
MODEL EVALUATION CRITERIA RESULT

Num	Accuracy	Loss	Precision	Recall	F1-score
1	0.976	0.177	0.948	0.947	0.948
2	0.972	0.179	0.937	0.939	0.938
3	0.972	0.177	0.937	0.937	0.937
4	0.968	0.178	0.929	0.925	0.927
5	0.973	0.175	0.946	0.936	0.941
6	0.973	0.178	0.942	0.939	0.941
7	0.973	0.176	0.938	0.944	0.941
Mean	0.972	0.177	0.947	0.930	0.939

C. Results and Discussion

In our experiment, 750 test samples from the dataset were used to estimate the hazard-risk prediction performance of

TABLE VI
EVALUATION CRITERIA FOR EACH CLASS

Num	Precision			Recall			F1-score		
	c1	c2	c3	c1	c2	c3	c1	c2	c3
1	0.955	0.916	0.896	0.959	0.893	0.896	0.957	0.904	0.896
2	0.959	0.923	0.864	0.973	0.835	0.879	0.947	0.894	0.871
3	0.974	0.909	0.864	0.964	0.899	0.879	0.962	0.907	0.871
4	0.957	0.910	0.844	0.959	0.893	0.844	0.958	0.901	0.844
5	0.959	0.904	0.892	0.961	0.893	0.862	0.960	0.898	0.877
6	0.955	0.910	0.879	0.974	0.973	0.879	0.959	0.901	0.879
7	0.957	0.910	0.866	0.962	0.893	0.896	0.960	0.901	0.881
Mean	0.957	0.910	0.872	0.962	0.893	0.896	0.960	0.901	0.881

our model. The sample size of c1, c2 and c3 are 535, 157 and 58 respectively.

The test dataset was divided into 3 classes of state, of which 732 samples were identified successfully. The confusion matrix is shown in Table IV. Confusion matrix is a standard applicable to supervised learning to display the results of the algorithm in the form of matrix. The row means the true classification and the column denotes the prediction classification. According to Table V, the model evaluation criteria result is shown in seven experiments. For all of the classifications, the mean accuracy is 0.972 and mean F1-score is 0.939. These results are still stable in seven experiments. It can be seen from these result in Table V that the classification result has high confidence. To a large extent, the most likely cause is that we considered multiple influential factors and utilized a multi-layer neural network, which made our model more precise. This result shows that using our deep RNN predictive model is insensitive to parameter changes due to the risk state. It can be applicable for guiding to decrease the occurrence of hazards, and the deep neural network is effective in train collision classification.

Our dataset is unbalanced, and most samples are in the no-collision state, since severe collision is a few. Table VI shows the criteria of precision, recall and f1-score for each class in seven experiments. The precision, recall and f1-score of c1 are high, reaching 0.95 for the no-collision state. But, c3's precision is only around 0.872, probably because samples of other classes, like c1, are predicted to be of c3, which leads to a larger denominator in precision calculation. This problem can often be resolved through subsequent manual reviews. For recall, c2 and c3 has good result, reaching around 0.86. It means most of the c2 and c3 samples could be identified, and the model did not miss the dangerous samples. Based on these results, our model is effective for this problem, which means the deep RNNs are suited for the forecast classification problem of hazard risk in the T2T-CBTC system.

However, there are some misclassifications in our results. The misclassification of c2 as c3 and c3 as c2 are easily explained. The reason is that the operation parameters of the train in these sequences were close, but the driver's reaction ability would cause an change in the collision situation. When a collision was found, the driver has already taken timely braking measures and no serious collision occurred. We can also get the verification from the classification probability. In the case of misclassification, the probability values of the two categories are very close. Communication delay also can lead to misclassification. The most likely cause is that long communication delays prevented from speed decreases in a short period.

TABLE VII
CONFUSION MATRIX OF DBN MODEL FOR HAZARD-PREDICTION TASK ON TEST DATASET WITH 500 SAMPLES

true / pred	c1	c2	c3
c1 (No Collision)	517	9	9
c2 (Slight Collision)	7	145	5
c3 (Severe Collision)	10	8	40

According to these data, we can infer that the predictive model has been trained well, and the system state parameter characteristics under normal or abnormal conditions have been learned completely. Taken together, these results suggest that our intelligent hazard risk prediction model is effective in system hazard risk prediction by considering multiple influential prediction features and using deep learning.

D. Comparison on Deep Belief Network (DBN)

We compared the performance of depth generation model DBN and our model on this classification problem. The DBN model is a probabilistic generative model that is composed of hidden and stochastic variables. It is a combination of some restricted Boltzmann machines (RBMs), where each sub-network's hidden layer serves as the visible layer for the next. All of the visible-layer units are fully connected to all hidden-layer units, and there is no connection within a layer. The DBN we considered used the same dataset for training and testing. The DBN contains two hidden layers, with 250 units per layer, and the output layer uses *softmax* classifier.

Table VII and VIII give classification results of the DBN. The result was compared with our proposed model based on LSTM-RNN. As we can see, the classification result in all evaluation criteria of our model was significantly higher than that of the DBN model, especially in recall and f1-score. In DBN, the number of samples correctly classified decreases in each class. Especially for c3, DBN performs more worse than LSTM. Only 40 are correctly classified, and the remaining 18 are misjudged. The most likely cause is that a severe collision is caused by multiple time-step accumulations, rather than a sudden change of a parameter. DBN has no ability to process time-series data, so it may not be suitable for

TABLE VIII
MODEL EVALUATION CRITERIA RESULT OF DBN MODEL

Num.	Accuracy	Loss	Precision	Recall	F1-score
1	0.932	0.274	0.856	0.862	0.851
2	0.931	0.224	0.855	0.864	0.855
3	0.943	0.247	0.859	0.879	0.875
4	0.936	0.232	0.867	0.866	0.866
5	0.939	0.240	0.859	0.857	0.858
6	0.934	0.216	0.874	0.852	0.863
7	0.925	0.228	0.872	0.852	0.865
Mean	0.936	0.235	0.865	0.860	0.859

the hazard-risk prediction problem in a T2T-CBTC system. We can integrate the two networks somehow to optimize our model.

VIII. CONCLUSION

This paper focuses on the problem of train-to-train CBTC system hazard-risk prediction. An intelligent hazard risk prediction model using LSTM-based deep RNN is proposed for a T2T-CBTC system. We propose a T2T-CBTC system architecture and movement authority (MA) function-generation process, which simplify the system architecture and improve operational efficiency. We take the collision between two trains as a hazard, and quantify it with the degree of collision. Based on the operating parameters of the system and train, the classification prediction of the severity of the collision is made, and the probability that each hazard will occur during the train operation is given. Four types of risk-prediction features were selected according to IEEE standards, especially considering the impact of the MA calculation on collisions. It is shown that of 750 test samples, 732 were classified correctly. The influence of the with- or without-MA attribute on the prediction performance is compared. The results show that the MA attribute can effectively improve the prediction accuracy of the model. We also compare our model with DBN. The results show that the accuracy of the LSTM-based deep RNN model is better than that of DBN, indicating that LSTM can learn the time correlation between data.

In future work, more features and types of hazards will be taken into account, which will make the prediction function of our model more comprehensive. Some feature-extraction methods will be used to improve the model's accuracy.

REFERENCES

[1] X. Yang, H. Yin, J. Wu, Y. Qu, Z. Gao, and T. Tang, "Recognizing the critical stations in urban rail networks: An analysis method based on the smart-card data," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 1, pp. 29–35, 2019.

[2] H. Sun, J. Wu, H. Ma, X. Yang, and Z. Gao, "A bi-objective timetable optimization model for urban rail transit based on the time-dependent passenger volume," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 604–615, Feb. 2019.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[4] H. Lee, C. Ekanadham, and A. Y. Ng, "Sparse deep belief net model for visual area V2," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2008, pp. 873–880.

[5] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[6] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[7] R. F. Berriel *et al.*, "Heading direction estimation using deep learning with automatic large-scale data acquisition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/8489154/

[8] T. de Bruin, K. Verbert, and R. Babuska, "Railway track circuit fault diagnosis using recurrent neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 523–533, Mar. 2017.

[9] T. Zhu and J. M. M. S. de Pedro, "Railway traffic conflict detection via a state transition prediction approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1268–1278, May 2017.

[10] R. I. Muttram, "Railway safety's safety risk model," *Proc. Inst. Mech. Eng. F, J. Rail Rapid Transit*, vol. 216, no. 2, pp. 71–79, 2002.

[11] Y.-P. Zhang, Z.-J. Xu, and H.-S. Su, "Risk assessment on railway signal system based on fuzzy-FMECA method," *Sensors Transducers J.*, vol. 156, no. 9, pp. 203–210, Sep. 2013.

[12] M. Rekik, C. Gransart, and M. Berbineau, "Cyber-physical security risk assessment for train control and monitoring systems," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, May/Jun. 2018, pp. 1–9.

[13] A. John, D. Paraskevadakis, A. Bury, Z. Yang, R. Riahi, and J. Wang, "An integrated fuzzy risk assessment for seaport operations," *Safety Sci.*, vol. 68, pp. 180–194, Oct. 2018.

[14] K.-S. Lin and C.-C. Chiu, "Vague set based FMEA method for risk evaluation of safety-related systems," in *Proc. IEEE Conf. Dependable Secure Comput. (DSC)*, Dec. 2018, pp. 1–8.

[15] M. Comptier, D. Déharbe, J. M. Perez, L. Mussat, P. Thibaut, and D. Sabatier, "Safety analysis of a CBTC system: A rigorous approach with event-B," in *Proc. 2nd Int. Conf., Rel., Saf., Secur. Railway Syst. Modeling, Anal., Verification, Certification (RSSRail)*, Pistoia, Italy, Nov. 2017, pp. 148–159.

[16] G. Cavone, M. Dotoli, and C. Seatzu, "A survey on Petri net models for freight logistics and transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1795–1813, Jun. 2018.

[17] H. George-Williams, M. Lee, and E. Patelli, "Probabilistic risk assessment of station blackouts in nuclear power plants," *IEEE Trans. Rel.*, vol. 67, no. 2, pp. 494–512, Jun. 2018.

[18] S. Nefti and M. Oussalah, "A neural network approach for railway safety prediction," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 4, Oct. 2004, pp. 3915–3920.

[19] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, p. 1527, 2006.

[20] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[21] J. Wang and C. Zhang, "Software reliability prediction using a deep learning model based on the RNN encoder–decoder," *Rel. Eng. Syst. Saf.*, vol. 170, pp. 73–82, Feb. 2018.

[22] Y. Zhang, J. Han, W. Li, T. Zhou, J. Sun, and J. Luo, "Safety prediction of rail transit system based on deep learning," *Proc. IEEE/ACIS 16th Int. Conf. Comput. Inf. Sci. (ICIS)*, May 2017, pp. 851–856.

[23] Z. Yuan, X. Chen, Y. Yu, H. Sun, T. Zhou, and Z. Jin, "Simplifying the formal verification of safety requirements in zone controllers through problem frames and constraint-based projection," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 11, pp. 3517–3528, Nov. 2018.

[24] L. Han, W. Li, T. Zhou, J. Sun, and X. Chen, "Safety requirements specification and verification for railway interlocking systems," in *Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 1, Jun. 2016, pp. 335–340.

[25] J. Qian, W. Li, X. Chen, and J. Sun, "Modeling and verification of zone controller: The SCADE experience in China's railway systems," in *Proc. IEEE/ACM 1st Int. Workshop Complex Faults Failures Large Softw. Syst. (COUFLESS)*, May 2015, pp. 48–54.

[26] *IEEE Standard for Communications-Based Train Control (CBTC) Performance and Functional Requirements*, IEEE Standard 1474.1-2004 (Revision IEEE Std 1474.1-1999), 2004, pp. 1–45.

[27] A. Ferrari, M. L. Itria, S. Chiaradonna, and G. O. Spagnolo, "Model-based evaluation of the availability of a CBTC system," in *Proc. Int. Workshop Softw. Eng. Resilient Syst.* Berlin, Germany: Springer, 2012, pp. 165–179.

[28] S. Hordvik, K. Øseth, J. O. Blech, and P. Herrmann, "A methodology for model-based development and safety analysis of transport systems," in *Proc. ENASE*, 2016, pp. 91–101.

[29] E. M. Clarke and P. Zuliani, "Statistical model checking for cyber-physical systems," in *Proc. Int. Conf. Automated Technol. Verification Anal.*, 2011, pp. 1–12.

[30] G. Agha and K. Palmskog, "A survey of statistical model checking," *ACM Trans. Model. Comput. Simul.*, vol. 28, no. 1, pp. 6:1–6:39, Jan. 2018. [Online]. Available: http://doi.acm.org/10.1145/3158668

[31] R. Srinivasan, *Importance Sampling: Applications in Communications and Detection*. Berlin, Germany: Springer, 2013.

[32] M. Lee and J.-H. Chang, "DNN-based speech recognition system dealing with motor state as auxiliary information of DNN for head shaking robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 1859–1863.

[33] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.

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

[35] C.-Y. Ma, M.-H. Chen, Z. Kira, and G. AlRegib, "TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition," *Signal Process., Image Commun.*, vol. 71, pp. 76–87, Feb. 2019.

[36] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *Proc. AAAI*, 2018, pp. 1–8.

[37] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[39] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

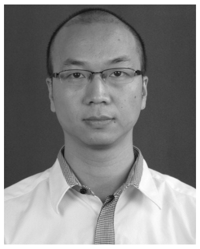
Jing Liu is currently a Professor of computer science with East China Normal University (ECNU), China. In recent years, she is currently involved in the area of model driven architecture. Her current works focus on the design of real-time embedded systems and cyber-physical systems.

Yan Zhang received the B.E. degree in electrical engineering and automation from the Nanjing University of Posts and Telecommunications in 2016 and the M.E. degree in software engineering from the Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, China, in 2019. She is currently an Assistant Engineer with China UnionPay. Her research interests include formal methods and model driven architecture.

Jiazhen Han received the B.S. degree in computer science from Jilin University in 2018. He is currently pursuing the master's degree in computer engineering with New York University (NYU), USA. He was an Intelligent Performance Architect with NVIDIA Company from July 2018 to July 2019. His research interests include machine leaning algorithms and computer architecture.

Jifeng He is currently a Professor of computer science with East China Normal University (ECNU), China. He is an Academician with the Chinese Academy of Sciences. He is also the Dean of the Software Engineering Institute, ECNU. In recent years, he has also involved in the mathematical model about the co-design of software and hardware. His works focus on the design of real-time embedded systems, cyber-physical systems, and the Internet of Things.

Junfeng Sun is currently the Assistant President of CASCO Signal Ltd., where he is also the Vice Director of the Research and Development Institute. In recent years, he is involved in the area of train control system design. His current work focuses on the design of railway signal security computer platforms.



Tingliang Zhou is currently the Director of the Research and Development Institute, CASCO Signal Ltd. In recent years, his work focuses on the design of automatic train control systems.