Received December 24, 2019, accepted January 11, 2020, date of publication January 20, 2020, date of current version January 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967634

Towards Safety-Risk Prediction of CBTC Systems With Deep Learning and Formal Methods

JING LIU[1], LI QIAN[1], YAN ZHANG[1], JIAZHEN HAN[2], AND JUNFENG SUN[3]
[1]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China
[2]Computer Engineering Department, New York University, New York, NY 10003, USA
[3]Research and Development Institute, CASCO Signal Ltd., Shanghai 200071, China

Corresponding author: Jing Liu (jliu@ sei.ecnu.edu.cn)

This work was supported in part by the National Key Research and Development under Project 2017YFB1001800, in part by the NSFC under Project 61972150, and in part by the Shanghai Knowledge Service Platform under Project ZF1213.

ABSTRACT Communication-Based Train Control System (CBTC) system is an automated system for train control based on bidirectional train-ground communication. Safety-risk estimation is a vital approach that strives to guide the CBTC system to guarantee the safe operation of vehicles. We propose a deep learning method to predict safety-risk states that combined with formal methods. First, the impact factors are selected, and the movement authorization (MA) failure rate is calculated by statistical model checking. Then, we use a deep neural network to model the relationship between the safe-risk states and the train operation status. Experimental results show that our method can achieve an accuracy of 97.4% on safety-risk prediction, and exceeds the baseline methods.

INDEX TERMS Communication-based train control system, risk prediction, deep learning, statistic model checking.

I. INTRODUCTION

Communication-Based Train Control (CBTC) system, is an automated train control system that authorizes safe movements of railway vehicles by bidirectional communications between vehicles [1]. It enhances the safety level and the transport efficiency of railway transportation. CBTC system is a safety-critical system, whose safety guarantee is remarkably necessary due to the unacceptable consequences of failures, such as loss of life, significant property loss, or damages to the running environment [2]. Safety-risk estimation is a technique to determine whether the operation of system is safe by the employing of quantitative or qualitative methods. Safety-risk estimation is not only a requirement of the CBTC system but also an efficient method for identifying the hazardous operation states that may cause damages. Generally, the process should consider multiple parameter factors of exploring the nonlinear function relationship between the factors and safety-risk states.

The safety-risk assessment, in many works, has encountered various challenges in CBTC systems. However, the main challenges are related to the uncertainty in system operations and the complicated relationship between multiple safety-risk states of CBTC systems. The significant deviation of predetermined running state would cause by the unreliability of system components and the difference of runtime environment, which is unavoidable and hard to estimate in the system. The uncertainty of the system suffers the traditional risk assessment methods. Fault Tree Analysis (FTA) [3], Failure Mode and Effects Analysis (FMEA) [4], Bayesian network analysis [5], and other static methods have no satisfying result of the assessment, owing to the deficiency of the consideration of how to eliminate the impact of the uncertainty. The requirement of analysis the correlations effectively between multiple safety-risk states of the CBTC system presents another challenge. Analyzing safety-risk states as independent of one another would lead to suboptimal models because the different safety-risk states reflect parameter changes in system operation. It is necessary to capture the correlation between various safety-risk states.

In this paper, we propose an intelligent-predictive method, which allows implemented by Deep Neural Networks (DNNs) for safety-risk estimation of the train control system in an uncertain environment. The model takes into account many factors related to risk for performing well. To be specific, a Deep Belief Network (DBN) is trained to predict the kind of risk states that would cause by some safety-related factors. The model is granted two critical abilities to solve

The associate editor coordinating the review of this manuscript and approving it for publication was Honghao Gao.

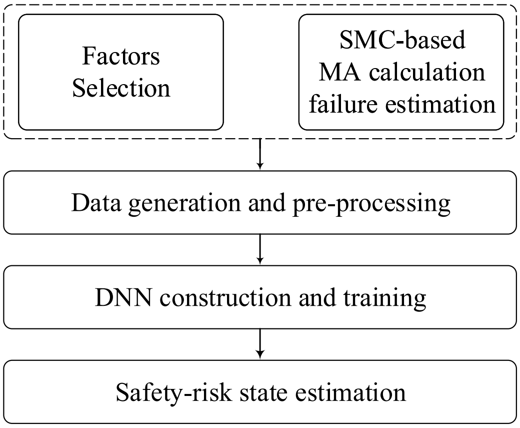


FIGURE 1. Procedure for predicting CBTC hazard risk.

the uncertainty and analysis of the correlation between the factors. It is capable of predicting the risk of the system and guiding the controller to prevent hazards. Moreover, a method to select and generate risk factors is proposed for processing the data before training. We transfer the formal verification result obtained by Statistical Model Checking (SMC) to the risk factors about safety-critical function, Movement Authority (MA). SMC samples behaviors (simulations) of the system, and solves the risk problem efficiently from formal methods perspective.

Fig. 1 explains the implementation process for building a neural network of safety-risk prediction of CBTC system based on deep learning. At first, essential works factor selction and MA calculation must be done. The quality of relevant factors selection can affect the effectiveness of learning algorithms in risk prediction to no small extent. We choose eight factors that are the most correlated to risk states to form a factor set base on previous researches [1], [6], [7]. At the same time, SMC is computing for the possibility of MA calculation failure. The quantitative verification can estimate the probability of a system-safety requirement by simulating and sampling the model. Next, we generate and preprocessing the data contained in the dataset, including missing value processing, outlier processing, and data regularization, etc.. We divided the dataset after preprocessing into two parts: train dataset and test dataset, used train the DNN and evaluate the training result respectively. Afterward, we build DNN and train the network with the dataset. Eventually, once the training is finished, the test dataset was supplied to the trained DNN to calculate the safety-risk status at the current condition and evaluate the performance of the trained DNN.

The remainder of this paper is as follows. In the next section, we review related work about safety-risk assessment. Subsequently, we describe the intelligent-predictive model in detail in Section III. In Section IV, we conduct several experiments and analyze the results. Finally, we conclude this paper in Section V.

II. RELATED WORK

There is a rich history of research on assisting in urban rail transit management and control for improving transportation efficiency. In many research works, safety-risk analysis and evaluation are not only be viewed as essential functional components in urban rail transit operation but also a pivotal guarantee mechanism to the safety of transportation. Researchers propose a wide variety of models that applied to analysis the safety of railway traffic.

H. Z. Huang *et al.* adopts FTA in safety analysis for railway traffic safety guarantee. In the fuzzy fault-tree model presented, a fuzzy set defined in probability space is proposed and applied to substitute the probability of failures. They mainly focus on accidents caused by human errors and hardware failures [8]. T.M. Zhu *et al.* proposed a state transition prediction approach for traffic state prediction and conflict detection based on proper State Transition Maps (STMaps) and corresponding relation matrices [9].

Safe control through formal methods is an emerging research idea of interest due to the reliability and provability of formal methods that usually applied to safety-critical systems. M. Comptier *et al.* conducts a safe analysis of RATP's CBTC system Octys. They present rigorous mathematical proof using Event-B and Atelier B tool to the safety of the system [10]. However, Event-B and some other formal methods require space and time discrete, and it may not be satisfied in Cyber-Physical Systems (CPSs). For hybrid systems, the Hybrid I/O Automaton (HIOA) framework is helpful for hybrid system verification. For making the inductive proof tractable, the HIOA model decomposes the proof into independent discrete and continuous parts [11], [12].

Moreover, artificial intelligence is an popular method to guarantee the safety of railway movement. S. Nefti and M. Oussalah proposed that Artificial Neural Networks (ANNs) may solve the problem of predicting system faults. By taking irregularities in the locating of the rails as input and using wavelet transformation to reduce the dimensionality of the input, the ANN may predict the safety rate of the rails [13].

Rencently, deep learning [14]–[16] is increasingly popular for solving complex problems. DNN is an effective method for learning features from data regardless of whether or not prior knowledge exists. DNNs have achieved enormous success in many fields, such as data representation learning, time-series data prediction, and pattern recognition, etc. [17]–[19]. Besides, DNNs are also a well-established approach in traffic flow prediction [20], [21], automatic driving fault prediction [22], and railway track circuit conflict detection [9], [23].

W.H. Huang *et al.* proposed Deep Belief Networks (DBNs) with multitask learning (MTL) that can predict traffic flow in an unsupervised fashion. They build a network architecture consists of MTL as the top layer and DBN as the lower layer. For MTL being more productive, a grouping method has been proposed based on the weights in the top layer. The lower layer and MTL layer are working together to improve the performance of the deep architecture [20].

Another essential network architecture is Recurrent Neural Network (RNN), which is also utilized in faults prediction

TABLE 1. List of safety-risk prediction factors in impact hazard.

Category	Factor Variable	Range
Facilities	Communication delay (A)	$0.5-2\ s$
	Trains maximum (B)	$10-40$
Equipment	Train speed (C)	$-0.5-60\ km/h$
	Train location accuracy (D)	$5-10\ m$
Human	Working time (E)	$0-10\ h$
	Passenger flow (F)	$30-40\ k/h$
Procedures	MA calculation failure rate (G)	$< 10^{-8}$
	MA calculation time (H)	$0.07-1\ s$

and reliability assessment. J.Y. Wang and C. Zhang use a deep learning model based on the RNN encoder-decoder to predict the fault amount and assess software reliability. Experimental results show that the RNN performs better in prediction than traditional neural networks and parameter models [24].

A method is proposed using Long-Short Term Memory Recurrent Neural Network (LSTM-RNN) for fault diagnosis in railway track circuits. LSTM-RNN is applied to achieve detection and identification of errors timely by extracting temporal and spatial dependencies from the available measurement signals [23].

However, the quantitative analysis estimates the event based on the judgment of Boolean formulae. It relies on the expert system so that the absence of objectivity and real-time. Methods like ETA that employed in the system design stage are short of the consideration to stochastic behaviors of the CBTC system in actual running environment. Formal methods are capable of verifying the safety, yet limited by the incapability to predict accurately. Neural Networks (NNs) may predicting system safety-risk efficiently, but NNs is hard to express sophisticated data features. This paper combines formal verification and deep learning to predict behaviors of the CBTC system under uncertainty.

III. AN INTELLIGENT-PREDICTION MODEL FOR CBTC SYSTEM SAFETY-RISK PREDICTION

The object of our research is to build an appropriate intelligent-prediction model to predict the safety-risk state, taking as input with several risk-influencing factors. In this section, we demonstrate the prediction model in detail and a deep neural network is built using DBN in this model.

A. RISK FACTORS SELECTION

As major elements that consist of the system and its environment, prediction factors include four members: facilities, equipment, human, and procedures [1]. For these four types of factors, we picked eight factors that are the most closely relevant to the real-world. Table 1 lists the factors with their range value.

- *Facilities* Communication delay includes typical and worst-case transmission times of MA messages between wayside and train. The maximum number of trains is the maximum train numbers that the CBTC system can process within a given area of control. Once there are too many trains in the control area, the performance and efficiency of system control would decrease, leading to accidents possibly.
- *Equipment* Train speed means the driving velocity in train operation during the whole running process. Train location accuracy implies the precision of the train position measurement and reflects the size of the error.
- *Human* Working time indicates to the duration a driver has been working continuously. Passenger flow refers to the number of humans that take the subway per hour.
- *Procedures* MA calculation failure means that result of calculation is not right. MA calculation failure rate is the proportion of MA calculation error, is the ratio of calculation error and calculation number. MA calculation time is the time spent on MA computation per cycle.

B. SMC-BASED FAILURE RATE ESTIMATION

The correctness of MA plays a vital role in avoiding collisions. Incorrect MA computation could provide unreliable control commands for the train and trackside equipment. EOA refers to an endpoint of the safety block that the track area from the current position of the train to this endpoint is safe to move, no other train can enter this region simultaneously. MA is the most crucial factor associated with collision events, determines the end of authority (EOA).

The Zone Controller (ZC) is the heart subsystem of the CBTC system, which is responsible for calculating the MA. Generally, ZC figures out the MA based on the train position and current state of related equipment on the track. The results would be transmitted to the train to adjust its driving behavior. In this paper, we used SMC to get the probability of the MA computation failure. The module for computation of the EOA is *CAL_EOA*, which is built using MATLAB/Simulink in our preliminary work to grasp the computation scenario simulation samples.

We sample the data in every short time for collect the data about the information about train running. In each sampling point, the wrong information about the end of the travel may result in incorrect control command, which causes accidents. However, it is not easy to determine whether the current MA is wrong. A better method is to use MA calculation failure rate at current system parameters.

1) FORMAL SPECIFICATION

We convert the MA failure probability computation problem into a quantitative verification of safety requirements. We obtained the probability by employing system simulation on probabilistic and formal verification techniques. In this paper, the SMC is applied. SMC is a simulation-based approach to verify properties specified by temporal logic [25]. The inputs of the SMC model checker are a system model and a specification in temporal logic. Temporal logic is the formal language for describing system properties, consists of a set of propositions and temporal operators. The commons are Linear Temporal Logic (LTL), Computational Tree LogicL (CTL), Probabilistic Computational Tree Logic (PCTL), and Bounder Linear Temporal Logic (BLTL). Describing the requirements with temporal logic can well verify the satisfaction of the constraint during system operation. Model construction and sampling executions trace of the model is the first steps. After sampling, the statistical inference would be applied.

Definition 1: Given a *CAL_EOA* model $\mathcal{M}$, and a safety requirement property ϕ, statistical model checking estimation would verify whether the model $\mathcal{M}$ satisfies the specification ϕ with higher or equal probability to the threshold θ, the verification target formally defined as:

$$\mathcal{M} \models P_{\geq\theta}(\phi) \tag{1}$$

Five critical sensors are involved in the calculation of the MA. As long as each sensor is guaranteed to receive and transmit correct values at each moment, the MA calculation would not fail. We described the requirement utilizing BLTL, described as:

$$\phi = F^{100}G^{1}(\phi_0(t) \wedge \phi_1(t) \wedge \phi_2(t) \wedge \phi_3(t) \wedge \phi_4(t)), \tag{2}$$

where ϕ is the properties specification, $\phi(i), i = 1, 2, 3, 4$ means invalid values and wrong MA that never generate by the sensors at the moment t, follows the Bernoulli distribution, denote as:

$$\phi_i(t) = InvalidValueDetected(t). \tag{3}$$

where $InvalidValueDetected(t)$ is the function that indicate whether the value generated by sensor and MA is wrong. Formula 2 states that within 100 cycles, at any moment, five sensors would not produce invalid values and wrong MA would not be generated.

Unlikely the classical statistical model checking, SMC used in this paper method use random sampling of system execution paths. The improved SMC cooperate with importance sampling and cross-entropy method to reduce sample state space. Based on Monte Carlo method, which generate N random simulations sequence $\chi_1, \ldots, \chi_N$, the probability is compted as:

$$\hat{\rho} = \frac{1}{N}\sum_{i=1}^{N} B\left(\chi_i \models \psi\right), \tag{4}$$

where $\hat{\rho}$ is the proportion of $\chi_i \models \psi$, B is an indicator function that returns 1 if ψ is satisfied in χ_i or 0. When the sample size is sufficiently large, the Bernoulli distribution would be closed to a normal distribution, whose the variance is:

$$D(\chi_i) = N\hat{\rho}(1 - \hat{\rho}). \tag{5}$$

However, when $\hat{\rho}$ keeps variance as low as possible, larger samples space is required.

2) IMPORTANCE SAMPLING

Importance sampling is a useful technique to reduce sample space in the employment of SMC [26]. By using weighted system simulations, it is possible to realize rare properties. Importance sampling works by introducing a weighting function $W(\chi_i)$ on the observed random variables without expectancy $E(\chi_i)$ change and variance diminishing. Therefore, finding a proper weighting function distribution is a crucial problem.Suppose the weighting function and random variables χ_i with optimal density f_* exist, expectancy $E(\chi_i)$can be written as:

$$\begin{aligned} E(\chi_i) &= \frac{1}{N}\sum_{i=1}^{N} B(\chi_i \models \psi)W(\chi_i), \\ W\left(\chi_i\right) &= \frac{f\left(\chi_i\right)}{f_*\left(\chi_i\right)}, \\ f_*\left(\chi_i\right) &= \frac{Nf\left(\chi_i\right)}{\sum_{i=1}^{N} B(\chi_i \models \psi)E\left(\chi_i\right)}. \end{aligned} \tag{6}$$

Zero-Variance is hard to achieve, but proximal distributions can be obtained by checking members of a parameterized family of distributions. The cross-entropy method may select the appropriate members that minimize Kullback-Leibler divergence from the optimally biasing, through sampling from the original and unbiased distribution. Once appropriate density distributions computed using the cross-entropy method, and then the probability is calculated base on the distribution. MATLAB/Simulink is applied for the model implementation platform.

C. NETWORK ARCHITECTURE

We use DBN to construct our proposed model. More specifically, stacking Restricted Boltzmann Machines (RBMs) to form a DBN and using a softmax regression layer at the output layer, and we can perform supervised fine-tuning on the whole network.

1) RBM AND DBN

Deep learning succeeds in many fields, is a combination of several RBMs. An RBM consists of two layers, and one layer is binary stochastic hidden units and another visible units, where the hidden layer of each sub-network serves as the visible layer for the next layer. The hidden variables have binary values, either 0 or 1. The values of all units are stochastic variables. Generally, they obey Bernoulli distribution or Gaussian distribution. All visible layer units are fully-connected to all hidden layer units, and no connection in visible layers. The visible layer unit describes an aspect or feature of observed data, and the meaning of the hidden layer unit can be view as a feature extraction layer.

RBM is an Energy-Based model, which associate scalar energy to each configuration of variables. Its energy function defines the probability distribution over variables. The energy function of the configuration $(\mathbf{v}, \mathbf{h})$ is a linear function and defined as:

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{i=1}^{|V|}\sum_{j=1}^{|H|} w_{ij}v_ih_j - \sum_{i=1}^{|V|} b_iv_i - \sum_{j=1}^{|H|} a_jh_j, \tag{7}$$

where $\mathbf{v}$ is visible vector units, h is hidden vector units, $\theta = (\mathbf{w}, \mathbf{b}, \mathbf{a})$ is the parameter tuple, w_{ij} is the symmetric weight between visible unit j with b_i and a_j as their bias. $|V|$ is the number of visible layer units and $|H|$ is the hidden. It is easy to compute the conditional probability distributions with this the probability of a visible vector $\mathbf{v}$:

$$p(\mathbf{v}; \theta) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h};\theta)}}{\sum_{\mathbf{u}} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h};\theta)}}. \tag{8}$$

Through training a sequence of RBMs iteratively, DBN may learns the feature of the input data. The parameters θ is the heart of training algorithm, which can learned by $p(\mathbf{v}|\mathbf{h}; \theta)$ and prior distribution over hidden vectors $p(\mathbf{h}|\theta)$ [20]. Therefore, the probability of visible variables generation as follows:

$$p(\mathbf{v}) = \sum_{\mathbf{h}} p(\mathbf{h}|\theta) p(\mathbf{v}|\mathbf{h}; \theta). \tag{9}$$

Once the data is not binary and cannot be modeled with the original RBM, the DBN would be modeled with real value that follows a normal distribution instead of the binary RBM, which is called real-time RBM. Corresponding energy function and the conditional probability distributions are as follow:

$$E(\mathbf{v}, \mathbf{h}; \theta) = \sum_{i=1}^{|V|} \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{j=1}^{|H|} a_j h_j - \sum_{i=1}^{|V|} \sum_{j=1}^{|H|} \frac{v_i}{\sigma_i} w_{ij} h_j \tag{10}$$

$$p(h_j|\mathbf{v}; \theta) = \operatorname{sigm}\left(\sum_{i=1}^{|V|} w_{ij} v_i + a_j \right) \tag{11}$$

$$p\left(v_i|\mathbf{h}; \theta\right) = N\left(\sigma_i \sum_{j=1}^{|H|} w_{ij} h_j + b_i, \sigma_i^2 \right) \tag{12}$$

where σ is the standard deviation vector of normal distribution visible units, and $N\left(\mu, \sigma^2\right)$ is the normal distribution with mean μ and variance σ.

The neuaral network built consist of three hidden layers that every layer contains 256 units. There is an RBM between adjacent layers, RBMs stacked to form a DBN. Through the many experiments and result analysis, the best configuration of the RBM number is determined to 3. The structural relationship between layers is shown in Fig. 2. Furthermore, the value of hyper-parameters influences the learning effect of neural networks, such as the hidden layers amount and the number of units per layer. Proper network structure and its parameter can improve the training to be optimal.

2) INPUT AND OUTPUT

Table 1 indicates safety-risk factors selected, the dimension of the date for input is eight. Suppose $\mathcal{I}$ is the input vector of the network:

$$\mathcal{I} = (A, B, C, D, E, F, G, H) \tag{13}$$

where $A, B, \ldots, H$ represent the input dimension of each portion separately.

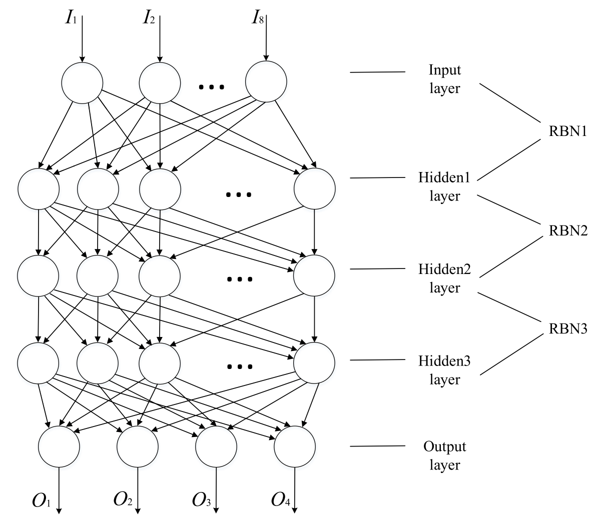


FIGURE 2. The network structure of the intelligent-prediction model for safety-risk estimation.

We use the softmax regression layer in the output layer, which layer contains 4 units, one for the normal state and three for each hazard. Suppose vector $\mathcal{O}$ is the output vector, composed of the possibility $o_k (k = 0, 1, 2, 3)$ of safety risk occurring. The probability of the risk occurring increases with the increasing of the value of o_k.

$$\mathcal{O} = \{o_0, o_1, o_2, o_3\} \tag{14}$$

where o_0 represents the safe state which no any risk occurs, o_1 means to the train-to-train collisions, o_2 depicts derailment of train and o_3 is train-to-structure collisions. The softmax layer takes the output of the last layer as its input. The probability of each risk o_k can be computed by:

$$P(Y = o_k) = \frac{e^{\operatorname{sigm}(\mathbf{w}_k \mathbf{v} + \mathbf{a}_k)}}{\sum_{i=0}^{3} e^{\operatorname{sigm}(\mathbf{w}_i \mathbf{v} + \mathbf{a}_i)}} \tag{15}$$

where $Y(t)$ is the prediction result at time t, sigm $(\cdot)$ is the sigmod fuction.

D. TRAINING ALGORITHM

In our method, during the training process, each layer is training separate from other layers. The first visible layer as input and the first hidden layer as output, while the weights are optimal. The output of this layer would serve as the input for the next hidden layer. Repeatedly, the method repeat in a next new network, first hidden layer as the input layer, and the current second hidden layer as the output. This procedure is iterated until training finished, that except the output layer, every layer learns their weights. The loss function is crucial for learning the optimal weight of the network. The optimize purpose of training is to minimize the objective function, which accumulated by the loss function of all layers. We choose the cross-entropy loss function as our loss function:

$$\operatorname{loss}(t) = -\log\left(P\left(Y(t) = o_k\right)\right) \tag{16}$$

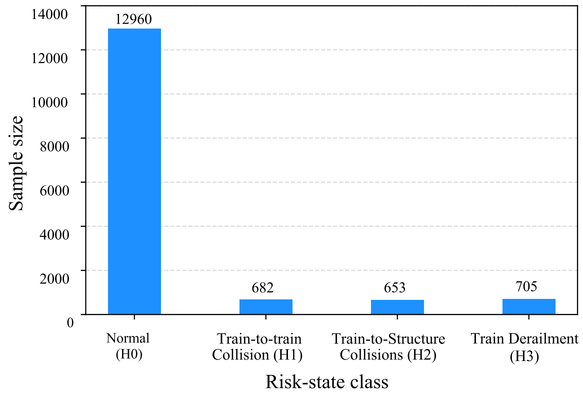


FIGURE 3. Risk-state class distribution of unbalanced dataset.

where $P(Y(t) = o_k)$ is the probability of correct prediction described in Equation 15. Cross-entropy describes the distance between two probability distributions. The smaller the cross entropy, the closer the two are. The object function is the function to minimize the loss function:

$$\min J(\theta) = min \sum_t \text{loss}(t) \tag{17}$$

IV. EXPERIMENT

In this section, we implement our intelligent-prediction model after training to estimate the safety-risk state of the CBTC system, and evaluate the performance of the model. Besides, the experiment result would be compared with other classical models.

A. EXPERIMENT SETUP

1) DATASETS

We gathered data from a company in which we cooperate, CASCO Ltd. The company dedicated to the train signal control system and has much data about the train control system. We split the dataset into training and test dataset. The total dataset has 15000 samples. We use the hold-out method to split our dataset into test dataset and training dataset. We randomly select 2500 samples from the dataset as the test dataset, and 12500 samples as the train dataset.

We use a combination of collection and simulation to generate a dataset. The Data Storage Unit (DSU) subsystem stores the used line information and configuration file information of each subsystem in the CBTC system. The DSU is composed of static track database, train database system, configurable parameter database, and dynamic track database. It includes trains' parameters such as length, speed, train time responses and track information like track, grade, curvature, maximum speed. For equipment, facilities, and human, we collected data from historical data in DSU of related companies. MA calculation failure probability, the non-DSU data, is not stored in the database.

Fig. 3 shows the sample class distribution in the dataset. The dataset is unbalanced obviously, 86.4% of samples are in the safe state, while only 13.6% are in the risk states.

2) PRE-PROCESING

The datasets need some extra preparation steps for suitable to the problem and compatible with the network. In this paper, before the training process, we preprocess the datasets, including missing-value processing, outlier processing, and data normalization. In the dataset, some value of the sample is missing, the value-missing presents its randomness, and the number of missing values is small. Therefore, we delete the samples whose feature value is not complete. Because of some unknown error, sensors may gather outliers, so attribute specification is used for removing the outlier. Besides, the datasets are normalized by min-max normalization:

$$x = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{18}$$

Data normalization may make the data have a mean of zero or to be centered, with a standard deviation of one, which assures that total data obey normal distribution.

3) HAZARD PREDICTION

IEEE standard 1474.1-2004 [1] indicates that for identify hazards and prevent accidents, establishing a system safety assessment program is necessary for CBTC systems. Seven common critical/catastrophic system hazards of CBTC systems are listed in the IEEE standard 1474.1-2004. For simplifying the complexity of the problem and highlighting our method, we select the three most essential and severe hazards and a safe condition as follows.

- Safe Condition (H0)
- Train-to-Train Collision (H1)
- Train-to-Structure Collisions (H2)
- Train Derailment (H3)

Safe Condition (H0) is a status without any hazards. The implication of train-to-train collision (H1) the impact of two trains at different angles include head-on, rear-end and sideswipe. For H1 accidents, the main related factors are train separation, route interlock status, and traffic direction reversal interlocks. Train-to-structure collisions (H2) include collisions between trains and original equipment, temporary equipment, or other buildings on the track. H2 collisions can be addressed through restricted route protection and end-of-track protection. Train derailment (H3) means that the vehicle departure from the correct orbit into other or non-orbital areas and may lead to other unforeseen disasters. H3 failures can be prevented by overspeed protection, interlocking protection, and broken rail detection.

4) EVALUATION METRICS

In the experiments, we adopt four evaluation metrics for measuring the performance of the model: Accuracy, Precision, Recall, and confusion matrix. Accuracy indicates the proportion of the correct prediction in total samples. Precision presents the portion of a risk state among the retrieved risk states. Recall means the proportion of specific risk states which is retrieved over the total amount of individual

TABLE 2. Confusion matrix for the safety-risk prediction task on a group of test dataset with 500 samples.

True \ Prediction	H0	H1	H2	H3
H0	430	1	2	2
H1	2	17	1	0
H2	1	0	23	1
H3	1	1	3	15

risk states. Confusion matrix is an evaluation metric that summarizes the records in the dataset in a matrix form according to the real category and the classification criteria predicted by the model.

$$Accuracy = \frac{1}{n}\sum_{k=1}^{m}\mathbb{I}\left(f\left(\mathbf{x_k}\right) = y_k\right) \tag{19}$$

$$Precision = \frac{TP}{TP+FP} \tag{20}$$

$$Recall = \frac{TP}{TP+FN} \tag{21}$$

where $\mathbb{I}(\cdot)$ is *Indicator* function that return 1 if the formula is true, else it would return 0. y_k is the real value and y'_k is a prediction value. TP is the true positive, FP is the false positive, TN is the true negative, and FN is the false negative. Confusion matrix records the full prediction result.

B. EXPERIMENT RESULT AND ANALYSIS

In experiments, we split the test dataset into 5 groups randomly. We use the 5 groups of 500 samples to estimate the performance of the intelligent-prediction model after the training process. Table 2 shows the confusion matrix of the safety-risk prediction task on a group of the test dataset, and Table 3 shows evaluation metrics results on five groups. For express conveniently, we use H0, H1, H2 and H3 denote the different risks.

There are four kinds of the state contained in the test dataset, in which 485 samples were identified correctly. Table 2 presents the confusion matrix. The row means the correct category, and the columns delegate the prediction category.

In the evaluation, most of the predictions are correct. However, concerning the prediction of collisions between train-to-train and train-to-structure collision, a few mistakes are taken. The reason for misclassification that predicts H1 as H3 and H3 as H1 may be the judgment of the endpoint type in the EOA calculation is different. In EOA calculation, the endpoint has two types: the one is the end of the train, and another is the turnout, which is the end of the railway or buffers. The communication delay may be the cause of the misclassification of H2 as H1.In the two cases, the interlocking system cannot receive or execute control commands timely, so that cannot be interlocked.

Table 3 shows the evaluation matrix in five groups of test datasets. The average accuracy of the test in five groups reaches 97.2%, and it is stable in the experiments.

TABLE 3. Model evaluation metrix in 5 groups.

Group	Accuracy	Precision	Recall
1	0.976	0.948	0.947
2	0.972	0.937	0.939
3	0.972	0.937	0.937
4	0.968	0.929	0.925
5	0.973	0.946	0.936
Mean	**0.972**	**0.947**	**0.930**

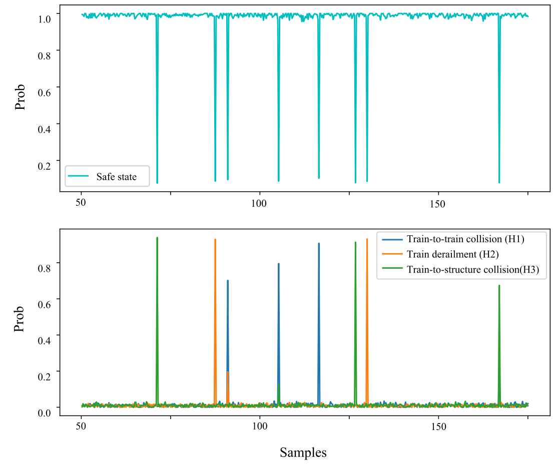


FIGURE 4. Probability of each risk of some sample points in the test dataset.

The recall of the test is 93.0%, which implies that the intelligent-prediction model may still perform well on unbalanced datasets. We consider multiple influential factors and employ a multilayer neural network that makes the prediction model more precise. The result confirms that the risk-prediction model we proposed is capable of the decline of occurrence of hazards, and the deep learning so beneficial for train collision prediction.

Fig. 4 indicates the probability of each risk of some sample points in the test dataset. The values represent the probability that the current sample encounter such hazards. At the sample point 119, there is a very high probability in H1, which implies that the sample may occur H1 risk. By analyzing the sample data, we found that the MA calculation failure rate is 0.0003 at the sampling point, which is much larger than 10^{-8} means that it is hazardous for the system that may be the main reason for this collision. Accordingly, there is a very high probability for H2 at 82 and 130. At 71 and 127, sample points are classified as H3 wrongly, which means when the occurrence probability of a specific hazard at the sample point increase, the probability of the safe state would decrease correspondingly.

Through these analyses, we can infer that the prediction model has reached a stable learning rate; the trained DNN has fully learned characteristics of hazards or safe state.

C. MODEL COMPARISON

To evaluate the performance difference between our model and the other method, we compare the performance of our model with others in the CBTC system. Several models are used for comparative research as baselines as follows:

TABLE 4. Comparison of accuracy, precision and recall for different model.

Model \ Metric	Accuracy	Precision	Recall
DBN	**0.972 ± 0.005**	0.923±0.004	**0.930 ± 0.006**
DCNN	0.923±0.009	**0.941 ± 0.011**	0.890±0.009
MLP	0.907±0.008	0.871±0.009	0.881±0.008
Bayesian network	0.907±0.012	0.889±0.019	0.853±0.012

- Deep Convolutional Neural Network (DCNN).
- Multi-layer Perceptron Neural Network (MLP).
- Bayesian network.

We compare our model with others concerning accuracy, precision, and recall on all test datasets, shown in Table 4. Among the results in Table 4, the DBN which implement for our intelligent-prediction model achieves the best performance in accuracy and recall compared with all baselines. Bias range is lower than DCNN, even if it is not the highest in precision. The precision of DCNN is the highest in all models, while the gap between DCNN and DBN in precision is not huge. However, the recall metric of DBN is much higher than DCNN. The one reason may be that DCNN would miss some risk-state samples in the classification.

Moreover, the performance ranking is $DBN > DCNN > MLP > Bayesian$. In this experiment, the neural network model is superior to static methods like Bayesian network. These static methods do not consider the uncertainty caused by various complicated factors in the system. Furthermore, our method gets the best performance in the experiment.

Summarily, the results suggest that our intelligent-prediction model is effective in the CBTC system for hazard risk prediction by the combination of formal verification and deep learning.

V. CONCLUSION

We present an algorithm to predict the safety-risk state for guaranteeing the safety of the railway vehicles that control by CBTC systems. The method combines formal verification and deep learning as a means to predict the safety-risk condition of the uncertain system without relevant prior knowledge. The main contributions include establishing the safety-risk estimation model, takes advantages of both deep learning and formal methods. Through experiments, we validate the availability and effectiveness of the method. The model is capable of predicting the hazard of the CBTC system accurately, and the accuracy of our approach reaches 0.974, which precedes other existing methods.

In future work, more prediction factors and types of hazards would be taken into account, which can potentially improve prediction performance and safety of systems. We are also interested in exploring the relationship between hazards and spatial-temporal data, and other deep learning algorithms may be implemented for solving it.

ACKNOWLEDGMENT

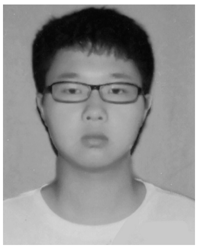
This article was presented in part at the 2019 15th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2019), London, U.K., August 2019.

REFERENCES

[1] C. Schifers and G. Hans, "IEEE standard for communications-based train control (CBTC) performance and functional requirements," in *Proc. Veh. Technol. Conf. (VTC)*, May 2000, pp. 1581–1585.
[2] J. Knight, "Safety critical systems: Challenges and directions," in *Proc. 24th Int. Conf. Softw. Eng. (ICSE)*, Jun. 2003, pp. 547–550.
[3] W.-S. Lee, D. L. Grosh, F. A. Tillman, and C. H. Lie, "Fault tree analysis, methods, and applications: A review," *IEEE Trans. Rel.*, vol. R-34, no. 3, pp. 194–203, Aug. 1985.
[4] D. H. Stamatis, *Failure Mode and Effect Analysis: FMEA from Theory to Execution*. Milwaukee, WI, USA: ASQ Quality Press, 2003.
[5] F. V. Jensen, *An introduction to Bayesian networks*, vol. 210. London, U.K.: UCL Press, 1996.
[6] T. Xu and T. Tang, "The modeling and analysis of data communication system (DCS) in communication based train control (CBTC) with colored Petri nets," in *Proc. 8th Int. Symp. Auto. Decentralized Syst. (ISADS)*, Mar. 2007, pp. 83–92.
[7] R. Pascoe and T. Eichorn, "What is communication-based train control?" *IEEE Veh. Technol. Mag.*, vol. 4, no. 4, pp. 16–21, Dec. 2009.
[8] H.-Z. Huang, X. Yuan, and X.-S. Yao, "Fuzzy fault tree analysis of railway traffic safety," in *Proc. Traffic Transp. Stud.*, Jul. 2000, pp. 107–112.
[9] T. Zhu and J. M. Mera Sanchez 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] M. Comptier, D. Deharbe, J. M. Perez, L. Mussat, T. Pierre, and D. Sabatier, "Safety analysis of a cbtc system: A rigorous approach with event-B," in *Proc. Int. Conf. Rel., Saf. Secur. Railway Syst.* Pistoia, Italy: Springer, 2017, pp. 148–159.
[11] S. Mitra, Y. Wang, N. Lynch, and E. Feron, "Safety verification of model helicopter controller using hybrid input/output automata," in *Proc. Int. Workshop Hybrid Syst., Comput. Control*. Berlin, Germany: Springer, 2003, pp. 343–358.
[12] L. Doyen, G. Frehse, G. J. Pappas, and A. Platzer, "Verification of hybrid systems," in *Handbook of Model Checking*. Springer, 2018, pp. 1047–1110.
[13] S. Nefti and M. Oussalah, "A neural network approach for railway safety prediction," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 4, Apr. 2005, pp. 3915–3920.
[14] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.
[15] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[17] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 499–515.
[18] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: https://arxiv.org/abs/1511.06434
[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[20] 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.
[21] 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.
[22] R. F. Berriel, L. T. Torres, V. B. Cardoso, R. Guidolini, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "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.
[23] 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.
[24] 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.
[25] G. Agha and K. Palmskog, "A survey of statistical model checking," *ACM Trans. Model. Comput. Simul. (TOMACS)*, vol. 28, no. 1, pp. 1–39, Jan. 2018.

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

JING LIU is currently a Professor of computer science with East China Normal University, China. In recent years, she has been involved in the area of model-driven architecture. She currently focuses on the design of real-time embedded systems and cyber-physical systems.



LI QIAN received the B.S. degree in computer science and technology from Anhui Agriculture University, in 2018. He is currently pursuing the master's degree in software engineering with East China Normal University. His research interests include formal methods and machine learning.

YAN ZHANG received the B.S. degree in electrical engineering and automation from the Nanjing University of Posts and Telecommunications, in 2016, and the M.S. degree in software engineering from the Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, China, in 2019. 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.

JUNFENG SUN is currently an 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.