Intelligent-Prediction Model of Safety-Risk for CBTC System by Deep Neural Network

Yan Zhang[1], Jing Liu[1](✉), Junfeng Sun[2], Xiang Chen[2], and Tingliang Zhou[2]

[1] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
jliu@sei.ecnu.edu.cn

[2] R&D Institute, CASCO Signal Ltd., Shanghai, China

Abstract. Safety-risk estimation aims to provide guidance of the train's safe operation for communication-based train control system (CBTC) system, which is vital for hazards avoiding. In this paper, we present a novel intelligent-prediction model of safety-risk for CBTC system to predict which kind of risk state will happen under a certain operation condition. This model takes advantages of popular deep learning models, which is Deep Belief Networks (DBN). Some risk prediction factors is selected at first, and a critical function factor in CBTC system is generated by statistical model checking. Afterwards, for each input of samples, the model utilizes DBN to extract more condensed features, followed by a softmax layer to decouple the features further into different risk state. Through experiments on real-world dataset, we prove that our new proposed intelligent-prediction model outperforms traditional methods and demonstrate the effectiveness of the model in the safety-risk estimation for CBTC system.

Keywords: Risk estimation · Deep learning · Communication-based train control system · Statistic model checking

1 Introduction

Communications-based train control (CBTC) system is the latest generation automatic train control system in the world, which is widely used in urban rail transit transportation [1]. There is no denying that CBTC system is a safety-critical system, whose failure could result in loss of life, significant property damage, or damage to the environment [15]. Safety risk estimation not only is a necessary requirement of CBTC system, but also is effective means to identify the hazardous operation conditions that may lead to a risk state. Risk estimation is a technique for identifying the operational safety of a system using either a qualitative or quantitative method, which have been utilized in safety management system. Generally, multiple system parameter factors need to be considered to explore the nonlinear relationship between them and the safety-risk state.

X. Wang et al. (Eds.): CollaborateCom 2019, LNICST 292, pp. 669–680, 2019.
https://doi.org/10.1007/978-3-030-30146-0_45

One of the main challenges for safety-risk assessment for CBTC system is uncertainty in system operation. The difference of running environment and the unreliability of system components would lead to the large deviation of pre-determined running state, which cannot be avoided or estimated in the system. However, traditional risk assessment models suffer from several limitations. Fault tree analysis [17], Failure mode and effects analysis (FMEA) [21] and Bayesian network analysis [14], all belong to static modeling analysis, would not have satisfactory results due to the uncertainty cannot be considered in the process. A second challenge stems from the need to effectively capture the correlations between multiple CBTC system safety-risk states. Considering that the different safety-risk states are parameter change of system operation, analyzing safety-risk states as independent of one another will lead to suboptimal models.

Recently, with the development of artificial neural networks, deep learning is currently the most popular method for data representation learning, time series data prediction, and image recognition, etc. Deep Neural Networks (DNNs) can be very effective in learning features from data in an unsupervised fashion without prior knowledge [7,16]. In addition, DNNs are also a well-established approach in traffic flow prediction [13], automatic driving fault prediction [4], and track circuit fault prediction [24]. These applications show that DNN has a good prospect in fault or risk prediction. The data record in CBTC system contains uncertainty information, which is another representation of the random behavior of the system. DNN can learn system features from these data to effectively solve the uncertainty problem. Meanwhile, DNN can capture the correlation between different safety-risk states.

In this paper, we proposed a deep intelligent-predictive model for hazard risk assessment in CBTC system. The model is implemented by deep neural network (DNN), considering a variety of influential factors. More specifically, a Deep Belief Network (DBN) is trained to predict which kind of risk state will happen from some safety-related factors. As a result, the proposed intelligent-predictive model has the advantage of simultaneously achieving two important desiderata: consideration of uncertainty and Capture of risk state correlation. A selection and generation method of risk factors is proposed, where one of risk factors about safety critical function, movement authority (MA), is obtained from Statistical Model Checking (SMC) [5]. Compared with traditional methods like numerical methods [3], as a kind of formal method, SMC samples behaviors of the system model and resolves the safety critical problem more efficiently.

The rest of this paper is structured as follows: Sect. 2 reviews related studies on risk assessment. Section 3 defines safety risk prediction problem and give the model framework. Our intelligent-predictive model is described in Sect. 4. Section 5 presents the experiment and result. Finally, we conclude this study with future work in Sect. 6.

2 Related Work

Safety risk analysis and evaluation has been long considered as key functional component in urban rail transit operation.

Huang *et al.* proposed to employ FTA in railway traffic system safety. In this model, they mainly considered traffic accidents caused by human errors and hardware failure, and proposed the fuzzy fault-tree model, which simulated the failure probability of each unit of the system by defining fuzzy sets in probability space [12]. Zhang *et al.* illustrate and analyze Interval Signal Control Function for Train Control Center case using Fuzzy-FMECA method, where in this method, FMECA is used to abstract the potential failure modes in such function and FAHP is to determine risk weight [23]. The formal method is mainly used for system safety verification, and focus on safety-critical applications. Mathieu Comptier *et al.* analyzed the safety of the Octys CBTC system interlocking infrastructure using formal proofs, Event B. They modeled verified it with Atelier B tool [6]. Hybrid I/O Automaton (HIOA) framework is very effective for hybrid system verification. Mitra *et al.* designed a supervisory pitch controller for a model helicopter system and verified some safety property based on HIOA [18].

In addition, some artificial intelligence methods are also applied to solve the collision prediction problem. Nefti and Oussalah proposed using Artificial Neural Networks (ANNs) architecture to deal with the prediction problem of the system fault. By 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. Moreover, they found out the best structure of ANN for predicting railway safety and evaluated performances [19].

Deep learning algorithms was proposed in 2006 [9,10], and so many researches are published on the basis of it. Huang *et al.* [13] proposed a deep architecture to predict traffic flow, which is a multi-task regression DBN to incorporate multitask learning (MTL). In addition, to make MTL more effective, the weights in the top layer were grouped to make the experimental results better. Jinyong Wang and Ce Zhang utilized a deep learning model in software reliability and faults prediction problem. This model is made up of recurrent NN (RNN) encoder–decoder. The comparison among exist models shows that their model has better prediction performance [22].

3 Problem Definition

Our overall research goal is to build an effective intelligent-prediction model to predict the safety-risk state classification, taking as input some risk-influencing factors.

Safety-Risk Estimation. We define the four safety-risk states in line with common hazards listed in IEEE standard 1474.1-2004 [1]:

- Normal (H0)
- Collision between two trains (H1)
- Derailment of train (H2)
- Train-to-structure collisions (H3)

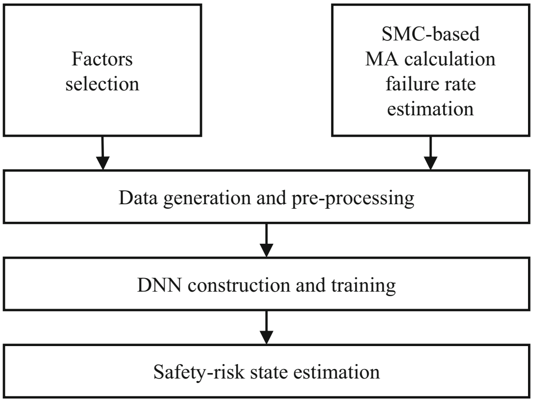

Fig. 1. Procedures for forecasting CBTC hazard risk

Figure 1 shows the implementation procedures for safety-risk prediction of CBTC system based on deep learning. First, we should have a feature selection process. After extensive investigations in published research [8,11], we built a set of risk-influencing factors. Eight factors that are most correlated with risk state are chosen. Meanwhile, for the MA calculation failure probability, we used the statistic model checking (SMC) method to calculate. Second, we generate and pre-processing a dataset, included normalizing them and dividing them into training and test datasets, the former to train DNN and the latter to participate in prediction. Afterward, we build DNN and train it with dataset. Once the training is over, the test dataset was fed to the trained DNN to forecast the safety-risk state at current situation.

4 An Intelligent-Prediction Model to Safety-Risk for CBTC System

In this section, our proposed safety-risk intelligent-prediction model is given in detail. We present a deep neural network model using DBN in this model.

4.1 Selecting Risk Factors

The prediction factor should include four kinds, such as *equipment, facilities, procedures, people* [1]. For the above four risk states, we selected eight factors that have the most closely related influence on them. They are listed in Table 1, with category and range value.

For the factor G, MA is the authority for a train to enter and travel through a specific section of track, in a given travel direction. It is the most influential factor associated with collision events. The Zone Controller (ZC) is the core subsystem of the CBTC system and is responsible for calculating the MA of the train on the track. Generally, taking the train head as the starting point and

Table 1. List of safety risk prediction factors that influence hazard

Category	Factor variable	Range
Facilities	Communication delay (A)	0.5 s $\sim$ 2 s
	Maximum number of trains (B)	10 $\sim$ 40 trains
Equipment	Train speed (C)	$\pm$0.5 km/h $\sim$ $\pm$60 km/h
	Train location accuracy (D)	$\pm$5 m $\sim$ $\pm$0 m
People	Working time (E)	0 h $\sim$ 10 h
	Passenger flow (F)	3 $\sim$ 4 w/h
Procedures	MA calculation failure rate (G)	$<10^{-8}$
	MA calculation time (H)	0.07 s $\sim$ 1 s

considering current state of relevant equipment on track, ZC calculates MA, and calculation results will be transmitted to the train to adjust its driving behavior. This end point is called end of authority (EOA), which is determined by the MA. The most likely cause of train collision is that the train has wrong information about the end of the current travel range, and results in an incorrect speed control command. Therefore, in each sampling point, once the MA calculation error, the train in this state must have an accident. However, it is not easy to judge whether the current MA is wrong, and a better method is to use MA calculation failure rate at current system parameters.

4.2 Calculating MA Failure Rate

For the purpose of obtaining MA calculation failure probability by SMC, MA calculation scenario simulation samples need to be realized by establishing a CAL_EOA module. Generation of the wrong MA is a rare event in ZC system. In order to obtain its failure probability, we convert it into a formal verification problem about safety requirements. Now, we use Statistical Model Checking (SMC) method estimates MA failure rate [5]. Statistical model checking is a simulation-based model checking approach to verify properties specified in a temporal logic [2]. In this paper, we estimated the MA failure probability by establishing a CAL_EOA model and defining a temporal logic formula for safety requirements.

Before estimation, we should give the formal definition of this problem in the system.

Definition 1. *Given a CAL_EOA model M, and a safety requirement property ϕ, statistical model checking estimation is to verify whether M satisfy ϕ with greater than or equal to a probability θ, that is $M \models P_{\geq\theta}(\phi)$.*

And 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 failure. We described the requirement utilizing BLTL. That is described as:

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

where $\phi_i(t)$ is:

$$\phi_i(t) = InvalidValueDetected(t) \tag{2}$$

Section 4.2 states that within 100 cycles, at any moment, five sensors would not produce invalid values and wrong MA would not be generated, where $InvalidValueDetected(t)$ follows the Bernoulli distribution.

SMC method use random sampling of system execution paths. Unlikely the classical statistical model checking, the improved SMC is merged with importance sampling and cross-entropy method to reduce sample state space. The basic idea is based on Monte Carlo method, which generate N random simulations sequence $\chi_1, \ldots, \chi_N$, followed Bernoulli distribution. Importance sampling is an effective technique to reduce samples space in the application of SMC [20]. Importance sampling starts by introducing a weighting function $W(\chi_i)$ on the observed random variables, without changing their expectancy $E(\chi_i)$ and reducing their variance. Therefore, finding a good weighting function distribution is a crucial problem. Suppose we have a weighting function and random variables χ_i with optimal density f_*, the general idea can be written as:

$$E(\chi_i) = \frac{1}{N}\sum_{i=1}^{N} B(\chi_i \models \psi)W(\chi_i) \tag{3}$$

The weighting function and optimal density are:

$$W(\chi_i) = \frac{f(\chi_i)}{f_*(\chi_i)} \tag{4}$$

$$f_*(\chi_i) = \frac{B(\chi_i)f(\chi_i)}{E(\chi_i)} \tag{5}$$

The cross-entropy method can select the appropriate members that minimize Kullback-Leibler divergence from the optimally biasing, through sampling from the original unbiased distribution. We got the appropriate distributions using the cross-entropy method. Once the density distributions had been decided, the probability was calculated. MATLAB/Simulink is used for model implementation platform.

4.3 Network Architecture

Our proposed DNN model is constructed by DBN. More specifically, stacking 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.

RBM and DBN. Deep Learning is a class of machine learning algorithms, which proposed by Hinton *et al.* in recently year [9]. DBN model is a probabilistic generative module and composed of stochastic variables. It is a combination of a number of Restricted Boltzmann Machines (RBMs). An RBM consists of two layers, that is, one layer of binary stochastic hidden units and one layer of binary stochastic visible units, where each sub-network's hidden layer serves as the visible layer for the next. Generally, they obey a Bernoulli distribution or a Gaussian distribution. All visible layer units are full-connected to all hidden layer units and no connection within the layer. Corresponding energy function $E(\mathbf{v}, \mathbf{h}; \theta)$ and conditional probability distributions $p(h_j|\mathbf{v}; \theta), p(v_i|\mathbf{h}; \theta)$ are given as following:

$$E(\mathbf{v}, \mathbf{h}; \theta) = \sum_{i=1}^{|V|} \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{j=1}^{|H|} a_j h_j - \sum_{i=1}^{|V|} \sum_{j=1}^{|H|} \frac{v_i}{\sigma_i} w_{ij} h_j \tag{6}$$

$$p(h_j|\mathbf{v}; \theta) = sigm(\sum_{i=1}^{|V|} w_{ij} v_i + a_j) \tag{7}$$

$$p(v_i|\mathbf{h}; \theta) = N(\sigma_i \sum_{j=1}^{|H|} w_{ij} h_j + b_i, \sigma_i^2) \tag{8}$$

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

Our network has three hidden layers containing 256 units each. Between each adjacent layer is an RBM, which is stacked to form a DBN. In our network, we have three RBMs. This configuration was obtained after many experiments and produced the best results for the current problem. The complexity of the research problem and dataset determined the size of our neural network altogether. The network structure that is too large or too small can not effectively improve the experimental performance.

Input and Output. Corresponding to the Table 1, the dimension of a sample in our dataset is eight, which corresponds to the eight safety-related factors we selected. Suppose $\mathbf{I}$ is the input sample vector of DBN, and the input layer unit is eight:

$$\mathbf{I} = (A, B, C, D, E, F, G, H) \tag{9}$$

Each unit refers to one of sample data features.

For output, we used *softmax* unit in the output layer to implement the task of hazard classification. The *softmax* layer contains 4 units, one for the normal state and three for each hazard. Suppose vector $\mathbf{O}$ is the output vector, $\mathbf{O}$ is composed of the likelihood of safety risk hazard o_k occurred. The greater the value, the greater the likelihood that this hazard will occur.

The $\mathbf{O}$ could be denoted as

$$\mathbf{O} = \{o_0, o_1, o_2, o_3\} \tag{10}$$

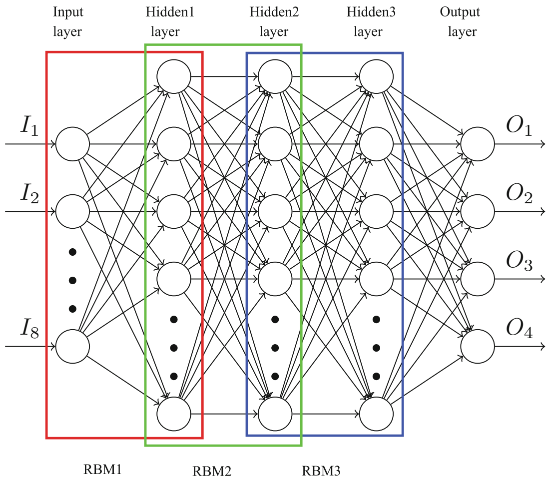

Fig. 2. The structure of intelligent-prediction model for safety-risk. The model is built by a DBN with three RBMs.

where o_0 refers to no collision occurred and system as a normal state, o_1 refers to collision between two trains, o_2 means derailment of train and y_3 is train to structure collision. The outputs of last hidden layer are the inputs to the *softmax* output layer. They give the probability of each category o_k as

$$P(Y = o_k) = \frac{e^{sigm(\mathbf{w}_k \mathbf{v} + \mathbf{a}_k)}}{\sum_{k=0}^{3} e^{sigm(\mathbf{w}_k \mathbf{v} + \mathbf{a}_k)}} \tag{11}$$

This DBN architecture is illustrated in Fig. 2.

5 Experiment and Results

In this section, we present the details of experiments on our intelligent-estimation model. We first introduce the dataset and evaluation metrics in experiments, and show the results of the model as well as compared models.

5.1 Dataset

We gathered datasets from a resourceful train control signal system company we cooperated, CASCO Ltd., and divided it into two parts, consisting of training and test. datasets. The total dataset has 15000 samples. The training dataset has 12500 samples and the test dataset has 500 samples. Our dataset is an unbalanced dataset. There are 12960 samples in H0, 682 samples in H1, 705 samples in H2 and 653 samples in H3. 86.4% of samples are in the normal state, while only 13.6% are in the risk states.

The dataset we received needs some extra preparation steps specific to our problem. The first step is to make the data comparable in the model. Data comparable preparation works by proper data normalization. We use

$$x = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{12}$$

to have a mean of zero, or to be centered, with a standard deviation of one. It can ensure that all data values obeys normal distribution.

5.2 Evaluation Metrics

In our experiments, the evaluation metrics include Accuracy, Precision, Recall. They are computed as

$$Accuracy = \frac{1}{n}\sum_{k=1}^{m} \mathbb{I}(f(\mathbf{x_k}) = y_k), \tag{13}$$

where $\mathbb{I}$ is Indicator function, y_k is the real value and y'_k is a prediction value.

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

Precision and Recall 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.

5.3 Results and Discussion

In our experiment, a group of 500 samples from dataset are used to estimate the hazard risk prediction performance of our intelligent model. Table 2 shows the confusion matrix for the safety-risk prediction task on test dataset with 500 samples, and Table 3 shows evaluation metrics results on five groups.

Table 2. Confusion Matrix for the safety-risk prediction task on test dataset with 500 samples.

True	Predicted			
	H0	H1	H2	H3
H0	430	1	2	1
H1	2	19	-	1
H2	1	1	15	2
H3	1	-	1	23

Table 3. Model evaluation metrics results in five group experiments

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

The test dataset was divided into 4 kinds of state, of which 485 samples were identified successfully, and the confusion matrix is shown in Table 2. The row means the true category and the columns delegate the prediction category. For all classification, the accuracy is 0.972 for five times. In the test dataset, most of the samples belong to the normal state data that were accurately classified, a total of 430 groups. However, for the collision between two trains and train-structure collision, there are some misclassification. The misclassification of H1 as H3 and the misclassification of H3 as H1 may be due to the judgment of the end point type is different in the EOA calculation, the former is the end of the train, and the latter is the turnout, the end of the railway or buffers. The communication delay could lead to the misclassification of H2 as H1. The most likely cause is the interlocking system in the two cases cannot receive the control command in time and cannot be interlocked. Table 3 is the evaluation metrics results in five group of test datasets. As we can see in Table 3, the mean of accuracy is 0.972 and it is stable in such experiments. Recall in these experiments is 0.930, which means that the intelligent prediction model can handle unbalanced datasets.

To a large extent, the most likely cause is that we consider multiple influential factors and utilized a multilayer neural network, where made our model more precise. This result shows that using our predictive model can make for the decline of occurrence of a hazard and the deep neural network is very effective in train collision classification. Taken together, these results suggest that our intelligent-prediction model is effective in system hazard risk prediction by considering multiple influential factors and using deep learning.

6 Conclusion

In this paper, we proposed a novel intelligent-prediction model, in which the model was learned from historical operation parameter samples in CBTC system, to predict which kind of risk state will occur. The design of intelligent-prediction model takes advantage of popular DBN. More importantly, we use a formal method, statistical model checking, to calculate one of prediction factors, the MA failure rate. Finally, experiments on real-world dataset validated the performance of our new proposed intelligent-prediction model and demonstrated the effectiveness of the deep learning framework in the safety-risk estimation for CBTC system. Without relevant domain knowledge, the dependence with the considered hazards can be learned by DNN from the dataset. It is shown that the proposed novel can significantly predict the hazard risk in the CBTC system.

In future work, more prediction factors and types of hazards will be taken into account, which can potentially improve prediction performance. We are also interested in exploring the relationship between a hazard and spatial-temporal data, and other deep learning algorithms may help solve it.

Acknowledgment. This paper is partially supported by funding under National Key Research and Development Project 2017YFB1001800, NSFC 61572195, NSFC 61802251 and Shanghai SHEITC Project 2017-GYHLW-01036.

References

1. Ieee standard for communications-based train control (cbtc) performance and functional requirements. IEEE Std 1474.1-2004 (Revision of IEEE Std 1474.1-1999), pp. 1–45 (2004). https://doi.org/10.1109/IEEESTD.2004.95746
2. Agha, G., Palmskog, K.: A survey of statistical model checking. ACM Trans. Model. Comput. Simul. **28**(1), 6:1–6:39 (2018). https://doi.org/10.1145/3158668, http://doi.acm.org/10.1145/3158668
3. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.P.: Model-checking algorithms for continuous-time markov chains. IEEE Trans. Softw. Eng. **29**(6), 524–541 (2003)
4. Berriel, R.F., et al.: Heading direction estimation using deep learning with automatic large-scale data acquisition. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2018)
5. Clarke, E.M., Zuliani, P.: Statistical model checking for cyber-physical systems. In: Bultan, T., Hsiung, P.-A. (eds.) ATVA 2011. LNCS, vol. 6996, pp. 1–12. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24372-1_1
6. Comptier, M., Deharbe, D., Perez, J.M., Mussat, L., Pierre, T., Sabatier, D.: Safety analysis of a CBTC system: a rigorous approach with event-b. In: Fantechi, A., Lecomte, T., Romanovsky, A. (eds) International Conference on Reliability, Safety and Security of Railway Systems, vol. 10598, pp. 148–159. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68499-4_10
7. Ekanadham, C.: Sparse deep belief net models for visual area v2. In: Advances in Neural Information Processing Systems, vol. 20, pp. 873–880 (2008)
8. Ferrari, A., Itria, M.L., Chiaradonna, S., Spagnolo, G.O.: Model-based evaluation of the availability of a CBTC system. In: Avgeriou, P. (ed.) SERENE 2012. LNCS, vol. 7527, pp. 165–179. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33176-3_12
9. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527 (2006)
10. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–7 (2006)
11. Hordvik, S., Øseth, K., Blech, J.O., Herrmann, P.: A methodology for model-based development and safety analysis of transport systems. In: ENASE, pp. 91–101 (2016)

12. Huang, H.Z., Yuan, X., Yao, X.S.: Fuzzy fault tree analysis of railway traffic safety. In: International Conference on Transportation and Traffic Studies, pp. 107–112 (2000)
13. Huang, W., Song, G., Hong, H., Xie, K.: Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. **15**(5), 2191–2201 (2014)
14. Jensen, F.V., et al.: An Introduction to Bayesian Networks, vol. 210. UCL press, London (1996)
15. Knight, J.C.: Safety critical systems: challenges and directions. In: Proceedings of the 24th International Conference on Software Engineering ICSE 2002, pp. 547–550. ACM, New York (2002)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
17. Lee, W.S., Grosh, D.L., Tillman, F.A., Lie, C.H.: Fault tree analysis, methods, and applications: a review. IEEE Trans. Reliab. **R-34**(3), 194–203 (1985)
18. Mitra, S., Wang, Y., Lynch, N., Feron, E.: Safety verification of model helicopter controller using hybrid input/output automata. In: Maler, O., Pnueli, A. (eds.) HSCC 2003. LNCS, vol. 2623, pp. 343–358. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36580-X_26
19. Nefti, S., Oussalah, M.: A neural network approach for railway safety prediction. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 4, pp. 3915–3920 (2004)
20. Srinivasan, R.: Importance Sampling: Applications in Communications and Detection. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-662-05052-1
21. Stamatis, D.H.: Failure Mode and Effect Analysis: FMEA from Theory to Execution. ASQ Quality Press, Milwaukee (2003)
22. Wang, J., Zhang, C.: Software reliability prediction using a deep learning model based on the RNN encoder-decoder. Reliab. Eng. Syst. Saf. **170**, 73–82 (2018)
23. Zhang, Y.P., Xu, Z.J., Su, H.S.: Risk assessment on railway signal system based on fuzzy-FMECA method (2013)
24. Zhu, T., de Pedro, J.M.M.S.: Railway traffic conflict detection via a state transition prediction approach. IEEE Trans. Intell. Transp. Syst. **18**(5), 1268–1278 (2017)