Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9147090, 9 pages
https://doi.org/10.1155/2022/9147090

Research Article

Software Enterprise Risk Detection Model Based on BP Neural Network

Jiahao Shan and Hongling Wang

School of Computer Science and Technology, Soochow University, Suzhou 215000, China

Correspondence should be addressed to Hongling Wang; hlwang@suda.edu.cn

Received 21 February 2022; Revised 31 March 2022; Accepted 12 April 2022; Published 27 April 2022

Academic Editor: Mu En Wu

With the rapid development of software industry, software enterprises have many problems in risk management, and enterprises are facing a huge crisis. In order to detect the risk of software enterprise, a risk detection model based on BP neural network is proposed in this paper. Firstly, risk analysis and modeling are carried out for the software life cycle process of the enterprise. Then, BP neural network is combined with common software risk items to realize supervised deep learning based on label information in data set. Using the advantage of neural network in automatic selection of original data features, the relationship between these risk items is extracted. Meanwhile, the network model in this paper learns the complex relational rules of the input risk features, to classify the risks. Finally, the proposed combined prediction model is compared with other prediction models. The experimental results show that the risk prediction effect of the proposed algorithm is better, to avoid the loss caused by the risk of software enterprises.

1. Introduction

Software development itself is a highly systematic and complex work. Whether software development can be carried out smoothly is often affected by many factors [1]. From the point of view of practical work, the process of software development work is also accompanied by certain risks. In the process of software development, we often encounter various problems. It can even lead to software development failure if some problems cannot be solved effectively. In view of the above situation, it is necessary to dynamically manage the potential related risks in the process of software development. There are also great differences in the development risks faced by developers at different stages of software development.

At the beginning of computer software project development, we should actively communicate with customers to clarify their basic requirements, function settings, scope of use, etc., to ensure that the designed computer software projects are consistent with the requirements of customers, and then carry out the development of computer software project. Before the development of computer software project, we often neglect the investigation and research on the software project market. Moreover, in real life, there are many kinds of computer software, and the needs of the public are complicated. Customers themselves cannot accurately demand functions of computer software. It is also harder for developers to know what customers really think about computer software projects [2]. Therefore, developers and customers have two understandings of software development. The developed software cannot satisfy customers, which leads to communication conflicts between customers and commissioning companies. In order to solve such problem, the contracting and development company should achieve precise positioning according to the customer's demand for software before the project development and send professionals to conduct in-depth investigation, to ensure the quality of computer software development and develop software to meet the needs of customers. Therefore, computer software project management is of great significance for computer software development and application [3].

With the rapid development of information technology in today's era, the difficulty and complexity of computer software development are relatively high, the scope involved is also relatively wide, and the corresponding workload is

very large. Nowadays, the development of computer software projects requires the coordination and cooperation of many people. Therefore, in order to complete the task of computer software development, the management of personnel is very important. Ensure that computer software projects remain on schedule and complete tasks at a high level and with high quality, and provide a passionate and positive working environment for staff. Reduce the flow of staff in the computer software development team, and promote the harmonious coexistence of computer software development staff, software project management staff and other departments, for the goal of software engineering development progress and overall quality [4]. In order to better manage the relevant staff of the project, we should establish the corresponding reward and punishment system and promotion system and formulate the corresponding evaluation system. Improving the enthusiasm of staff at multiple levels can not only exercise the staff related to software development projects but also improve the company's management system and improve the comprehensive quality of computer software project development. To a certain extent, it reduces the comprehensive cost of software development and improves the comprehensive benefits of computer software projects [5].

In the process of computer software project development, it is necessary to strengthen the comprehensive control of computer software project and increase the communication between the company and customers [6]. Minor adjustments are made during development according to customer requirements for software. Strictly control the time and cost, risk degree, and quality requirements in the process of computer software development. Timely complete the relevant development of computer software projects according to the original plan, and avoid big mistakes in any development links. Report and deal with the problems and risks found in time to ensure the perfect conclusion of computer software development related work. Reduce the defects and risks in computer software development. Update and maintain software regularly and monitor and solve problems in time.

There are many uncertain risk factors in the process of computer software project development, some of which may directly lead to the overall breakdown of computer software [7]. So, it is very important to reduce the risk of computer software project development. In order to solve the above problems, the corresponding software risk warning system should be established during the development of computer software projects to prevent the occurrence of destructive risks [8]. Hold regular project meetings to report project development status and current operation status in time. Implement real-time control, real-time transfer, and real-time resolution of identified risks, to ensure the highest quality of computer software projects while maintaining the overall economy.

In the process of computer software development, project managers play a very important role in software development [9]. However, due to different majors, the level of staff is uneven, so the ratio of computer software project management team members is mismatched. Therefore, there will be a series of factors affecting the work, such as the difficulty of division of labour among the staff in the development process and the abnormal state of "alternating drought and flood." As a result, the enthusiasm of employees to work decreases and the contradictions among employees increase. It is very easy to rise to the contradiction of the development team, which greatly affects the overall work efficiency and enthusiasm of employees and reduces the process and overall quality of computer software project development. In order to solve the above problems, first, in the early stage of computer software project development, different staff should be assigned to form a team according to the specific situation of the project, to ensure that everyone in the team can give full play to their strengths. Secondly, the company should have a special coordination team department to maximize the realization of even distribution and formulate reasonable working mode, ability, and salary link mode, those who can do more work and those who can earn more and maximize the overall harmony of the development team, to ensure the smooth progress of computer software projects.

After a computer software project is released, it is evaluated synchronously with the number of downloads. Before the development of computer software projects, if there is not enough understanding of software functional requirements and relatively rough market research, it will lead to a lot of differences between software positioning and expected positioning [10]. And the user's evaluation of the software will also be reduced. Neither the developer of computer software nor the software can be fairly evaluated which greatly hit the enthusiasm of computer software development staff, resulting in lower enthusiasm for work. Do preliminary investigation to ensure accurate positioning of software. To achieve users' objective evaluation of the software, consolidate the stability of the development team and increase the enthusiasm of the staff to work.

In order to improve the quality of software operation and minimize the dynamic risk of software enterprises, this paper proposes a software enterprise risk detection model based on BP neural network [11].

The innovations and contributions of this paper are listed below.

(1) The dynamic risk of software enterprise is detected

(2) The loss caused by the risk can be avoided

(3) The validity of the proposed model is verified by experiments and analysis

This paper consists of five main parts: the first part is the introduction, the second part is related work, the third part is the proposed software enterprise risk detection model, the fourth part is the experiment and analysis, and the fifth part is the conclusion; besides, there are abstracts and references.

2. Related Work

BP neural network has efficient nonlinear data function mapping approximation function [12]. It is a powerful data

modelling tool that can capture and represent complex input and output relationships.

2.1. Network Initialization. Neural network model is mainly composed of input layer, hidden layer, and output layer. The characteristics of enterprise risk serve as input to the input layer. The hidden layer is used to receive data from the input layer. The relationship between the input layer and hidden layer can be described as input layer neurons for x, the input layer of input for X_x, x input layer neurons to the weights of hidden layer neurons y for M^l_{yx}, and threshold of hidden layer neurons y for the θ^l_y. The expression of the mapping relationship between the output B_y of the hidden layer neuron y and the input X_x of the input layer is shown in

$$B_y = f\left(\sum M^l_{yx} X_x + \theta^l_y\right), \tag{1}$$

where $f(\cdot)$ is the activation function, and Softmax activation function is used in the output process of the output layer, which is set to three units to represent the output enterprise risk characteristic type. Softmax activation function can map the output of multiple neurons to the interval (0, 1) and select the classification with the highest probability as the enterprise risk prediction result.

2.2. Calculation of Output Value. In the BP neural network model, there is a computational linear relationship between layers. After network initialization is complete, appropriate activation functions need to be selected for each layer. The purpose is to obtain as much linear transformation set space of learning input data between each layer in neural network as possible, to make full use of the advantages of multilayer representation. In the network, the hidden layer learns the linear transformation of the input layer data (the middle layer of the network uses ReLU activation function), and ReLU activation function is used for the hidden layer neuron output. When input $i < 0$, the output is 0; when $i > 0$, the output is I. The activation function enables the neural network to converge faster, and the calculation formula is shown in

$$\phi(i) = \max\,(0, i). \tag{2}$$

According to the output B_y of hidden layer neuron y, the weight M^2_y from hidden layer neuron y to output layer, and the threshold $\theta 2$ of output layer, the output value J_{pred} (enterprise risk prediction value) of output layer can be obtained. The expression is shown in

$$J_{\text{pred}} = f\left(\sum M^2_y \cdot B_y + \theta^2\right). \tag{3}$$

2.3. Model Training. In order to make BP neural network predict enterprise risk reliably, it must be trained properly. In the training process, BP neural network algorithm uses gradient descent method to find the optimal solution [13]. Between the output layer and the hidden layer, the error value is segmented according to the weight ratio, and the specific error value related to each link is calculated. By reorganizing these error values to the error values associated with the neuron nodes of the hidden layer, these error values are again segmented according to the weight between the input layer and the hidden layer to complete the back propagation of the error. Among them, the most important step is to update the weight value and threshold value between layers, and the updating rule expression is shown in

$$M^2_y = M^2_y + \alpha \cdot \left(J_{\text{pred}} - J_{\text{real}}\right) \cdot J_{\text{pred}} \cdot \left(1 - J_{\text{pred}}\right) \cdot B_y, \tag{4}$$

$$\theta^2 = \theta^2 + \beta \cdot \left(J_{\text{pred}} - J_{\text{real}}\right) \cdot J_{\text{ured}} \cdot \left(1 - J_{\text{pred}}\right), \tag{5}$$

$$M^1_{yx} = M^1_{yx} + \alpha \cdot \left(J_{\text{pred}} - J_{\text{real}}\right) \cdot M^2_y \cdot B_y \cdot \left(1 - B_y\right) \cdot X_x, \tag{6}$$

$$\theta^1_y = \theta^1_y + \beta \cdot \left(J_{\text{pred}} - J_{\text{real}}\right) \cdot M^2_y \cdot B_y \cdot \left(1 - B_y\right), \tag{7}$$

where α and β are the learning rate. In the training process, the target output of BP neural network needs to be obtained. J_{real} is the true value of enterprise risk, which is output as the final target. If the error between the target output J_{real} and the predicted target J_{pred} is less than the currently set threshold, or the number of training iterations reaches the preset threshold, the model training will be completed.

3. The Software Enterprise Risk Detection Model Proposed in This Paper

3.1. Risks in the Software Development Cycle. This paper constructs a three-dimensional model (process dimension, time dimension, and logic dimension) to manage software risk.

3.1.1. Software Process Dimension. From the perspective of project management, risk identification is based on contract, project plan, work task breakdown WBS, various historical reference materials (like project information), various assumptions, preconditions, and constraints of the project [14]. From the perspective of the software development life cycle, the outputs of each stage (various documents) are the basis for risk identification in the next stage, and many technical risks can be analysed accordingly.

The process-oriented risk management model of the project is based on the process area model of CMMI. Software risk identification is based on process areas. The component model parts of a process area in the CMMI standard are shown in figure. It is a set of related practices within a domain. If the risks in these practices are all under control, the purpose of effective risk management in this process area can be achieved. Therefore, to focus on process risk is to focus on the risk of each subpractice in the process area.

In these subpractices, the risk of common practices can be viewed as a consideration of common problems in each process area. Proprietary practices require the establishment of risk lists for each proprietary practice.

The activity of the risk identification process is to transform project implementation uncertainty into explicit risk

statements. The key of this process is to identify risks systematically, not only to determine the source of risks but also to determine when they occur and the conditions under which they occur. Describe the process risk characteristics and determine which risk events are likely to affect the project. The project risk identification is process-oriented, based on process areas, key practices, and subpractices. Based on the software process-oriented metrics, the first, second, and third level metrics are taken as the basic risk metrics. Risk identification is not a one-time activity and should be carried out regularly throughout the project execution.

3.1.2. Logical Dimensions. Logical dimension refers to the logical thinking process of analysing and solving problems. Software projects are faced with many complicated risk factors. And the risk levels and the interrelationships among all kinds of risks are complicated, resulting in the objectivity and diversity of project risks. Any measures taken to control project risks will bring positive and negative impacts to the project. (1) Risk control measures can reduce the impact of risks on project cost and time limit. (2) Risk control actions require additional costs and time. These effects depend not only on whether managers act but also on the intensity and timing of risk control measures. Therefore, the risk management of software projects should aim at minimizing the total cost and duration of the project and decide whether and when to take a risk control measure according to the analysis of risk factors and the risk acceptance criteria of the subject. The risk is controlled in an acceptable and reasonable range, rather than blindly take measures to control all risks in the project.

Software risk management refers to the general steps used in risk management of software projects. As mentioned above, we adopted the common 5 steps of risk management in this project: risk identification, risk assessment, risk sequencing, risk mitigation, and risk supervision [15].

3.1.3. Time Dimension. The time dimension selects the basic activities in the software development process in ISO/IEC 18019 and organizes these activities according to the software life cycle time sequence.

The three-dimensional structure model of software process risk management in the whole life cycle aims to improve the ability of risk response and decision-making. Identify and manage risks throughout a software project and provide a holistic response to multiple risks to improve capital allocation capabilities in an effective assessment of overall project requirements.

According to the characteristics of the project, the time dimension can be divided into different stages. In each life cycle stage, the project risk is integrated with knowledge dimension according to logical dimension order and stage characteristics.

3.2. Risk Detection Model Based on BP Neural Network. In the process of enterprise risk detection by machine learning algorithm, the data set contains only one type of enterprise risk, which cannot reflect the actual problems existing in the software design process. This paper proposes an enterprise risk detection method based on BP neural network and classifies enterprise risks.

The use of neural network to detect enterprise risk is to take the vector form composed of measure feature information and label information as the input of neural network input layer and get features through the network and input them into the neural network classifier of objective function for training. The expected output of the classifier is the label of the sample. After several iterations of training, the trained neural network classifier can be obtained. Figure 1 shows the flow of the enterprise risk detection method in this paper.

3.2.1. Data Set Preprocessing. In the preprocessing of the data set, enterprise risk instances, measurement features, and labels need to be extracted. Due to the different results of measurement feature extraction by different researchers, this paper mainly extracts enterprise risk measurement features according to the following measures. The specific steps of extracting measurement features, enterprise risk instances, and labels are as follows:

Step 1. Use the automatic enterprise risk detection tool to detect enterprise risks, extract enterprise risk instances, and label them to generate labels.

Step 2. Calculate the measurement features of an enterprise risk instance and an enterprise risk instance. A sequence of floating-point numbers is used to encode the measure characteristics of enterprise risk and enterprise risk free instances, where 0 indicates that a measurement feature is not a factor affecting enterprise risk, and the pure decimal value indicates that a measurement feature is a factor affecting enterprise risk.

Step 3. Combine the measurement features and labels in Step 1 and Step 2 to generate two data sets at the method level and class level to form a training set.

The concrete realization process of using metric features and labels to form training sets is as follows:

The obtained metric features are combined with label information, and the metric features and label information are converted into vector representation $(w_{11}, w_{12}, ..., u_{1t})$. W represents metric features and U represents labels. After the merger, the structure of the two enterprise risk training sets is as follows: each row represents the enterprise risk instance, each column represents the measure feature, and the last column represents the label information, thus forming a matrix data W.

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & u_{1t} \\ w_{21} & w_{22} & \cdots & u_{2t} \\ \vdots & \vdots & & \vdots \\ w_{z1} & w_{z2} & \cdots & u_{zt} \end{bmatrix}. \tag{8}$$

3.2.2. Design of Neural Network Model. On the basis of electromyography (EMG) signal, artificial neural network is

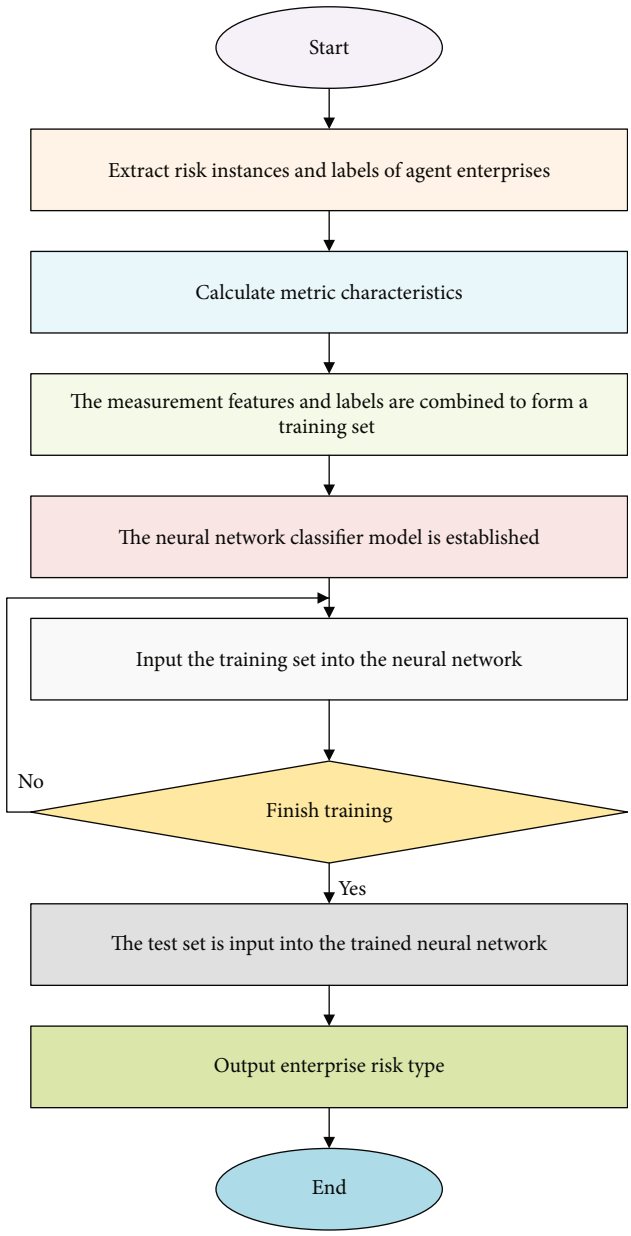

Figure 1: Procedure of detection method of bad smell in code based on BP neural network.

used to detect different predefined hand movements (up, down, left, and right). The designed network can successfully recognize hand movements. The convolution neural network and long-term and short-term memory neural network are combined to detect the defects of user camouflage intrusion mode [16]. The detection effect is better than the benchmark system, which verifies the effectiveness of this method. Neural network has advantages in classification and BP neural network model can be used to predict enterprise risk. The structure of BP neural network model is shown in Figure 2.

After preprocessing the data set, the data set is input into the neural network model, and the specific steps of enterprise risk detection are as follows:

The input refers to the labelled enterprise risk matrix data sample M and the output refers to the enterprise risk category.

Step 1. Establish a neural network classifier model, and the constructed neural network structure adopts the fully connected form. Layer 1 is an input layer and layer 2 is a hidden layer. The final layer of the network is the output layer. The output layer takes the Softmax function and outputs the categories of enterprise risks.

Step 2. Take the matrix data M of enterprise risk measurement features after data set preprocessing as the input of the input layer. Label information is represented as J_{real} as the network output benchmark, and the output value of the output layer is represented as J_{pred}. If the error between J_{real} and J_{pred} is less than the currently set threshold or the number of training iterations reaches the threshold, the neural network training for enterprise risk prediction output

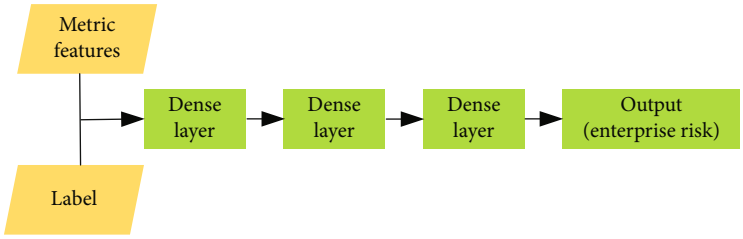

Figure 2: Structure of BP neural network classification model.

Table 1: Time efficiency analysis table.

Data group	Literature [17] model (%)	Literature [18] model (%)	Proposed model
1	0.69	0.66	0.76
2	0.74	0.75	0.83
3	0.88	0.85	0.91
4	0.76	0.83	0.93
5	0.66	0.71	0.88
6	0.73	0.72	0.8
7	0.78	0.73	0.79
8	0.82	0.83	0.88
9	0.84	0.77	0.91
10	0.59	0.51	0.77

will be completed; otherwise, the neural network input layer will be returned for model training.

Step 3. Take the enterprise risk open data set as the benchmark enterprise risk test set, and merge and transform the test data in vector form according to the method described in the data set preprocessing. Input the test set into the trained neural network model, and the model automatically outputs the predicted enterprise risk category.

4. Experiment and Analysis

In this study, the financial affairs of 158 enterprises were collected and sorted out, and the enterprises that were dealt with due to financial anomalies were set as dangerous enterprises. The data were combined to form a sample set, and the prediction was made based on the data of 2019 by using the model. At the same time, the prediction index system adopted in this study is summarized based on the prediction variables commonly used by the company at present, and some factors are input into the overall structure of the model. As the literature [17] model and literature [18] model are also relatively common prediction models with good prediction effects, therefore, the model proposed in this study is compared and analysed with the literature [17] model and literature [18] model to verify the advantages and disadvantages of different models.

4.1. Time Efficiency. Among the input factors of the model, the actual situation, profit situation, operation situation, and growth situation have a strong relationship with time efficiency. The time efficiency of prediction is fast, and the overall situation of enterprise risk can be timely understood, so that enterprises can quickly make risk prevention and control measures. Therefore, in time efficiency analysis, the input of prediction model is mainly based on these four factors.

During the experiment, the enterprise risk data were grouped into 10 groups, and the test data were used to compare and analyse the time spent by the three models in predicting enterprise risk crisis, as shown in Table 1 and Figure 3.

According to the experimental results, the execution efficiency of the proposed model is high, which can improve the execution efficiency of the algorithm to a certain extent.

4.2. Prediction Accuracy. Losses and accuracy on samples need to be monitored during training and testing. Accuracy, F1 value, and AUC value were used to evaluate the classification performance of the neural network model. The specific calculation formulas of accuracy and F1 value are shown in

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%, \tag{9}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%, \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%, \tag{11}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\%, \tag{12}$$

where TP represents the positive sample class, TN represents the negative sample class, FP is the number of correct samples classified into incorrect samples, and FN is the number of correct samples classified into incorrect samples.

F1 value can be regarded as a weighted average of model accuracy and recall rate, and its value ranges from 0 to 1. In the classification task, both accuracy and recall rates are expected to be very high, but in practice, they are impossible to achieve. Therefore, you need to strike a balance between the two. The value of F1 can be regarded as this equilibrium point. The higher the F1 value is, the higher the accuracy and recall rate are and the balance is achieved.

The AUC value is defined as the area under the ROC curve, which is not affected by the threshold. As numeric types, classifiers with higher AUC values work better.

Through the in-depth analysis of the method proposed in this study, the prediction model of enterprise risk portfolio based on BP neural network has certain advantages.

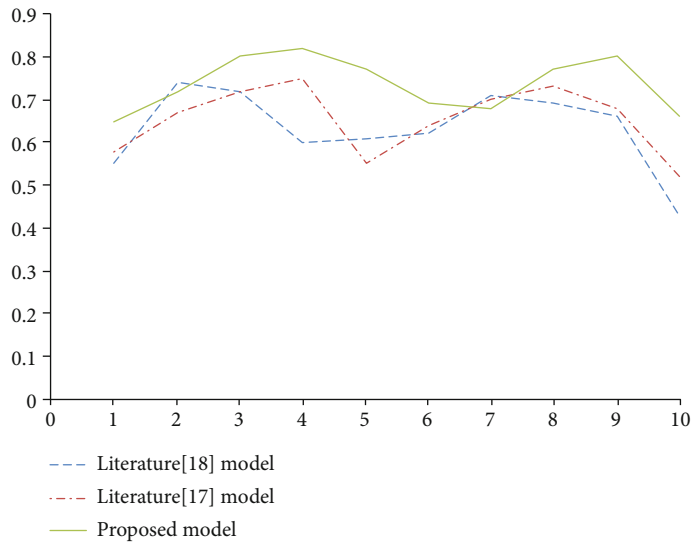

Figure 3: Time efficiency analysis diagram.

Table 2: Test results of data class data set.

Detection method	Evaluating indicator		
	Accuracy (%)	F1 value (%)	AUC value (%)
J48	82.70	58.78	1.44
Random forest	72.21	11.11	1.39
Naive Bayes	66.04	48.18	1.42
JDeodorant	74.37	12.09	1.32
Proposed method	95.08	94.19	1.51

Table 3: Security analysis table.

Data group	Literature [17] model (pcs)	Literature [17] model (pcs)	Proposed model (pcs)
1	4	2	1
2	3	2	2
3	3	2	1
4	2	4	1
5	1	3	1
6	4	2	2
7	2	1	1
8	3	3	1
9	1	2	1
10	4	2	1

However, as the actual situation of enterprise risk is changeable, the test of the actual effect of the model needs continuous research. Subsequently, the input parameters of the model can be constantly adjusted to test the reliability of the model according to the changes in the actual situation of enterprise risk.

The experiment analyses the enterprise risk detection results of the enterprise risk detection method based on BP neural network and machine learning algorithm on the same data set from the following three aspects.

The effect of different types of enterprise risks on neural network classifier is included in the data set. In order to explore the influence of different types of enterprise risk in the data set on the detection effect of neural network classifier, enterprise risk data sets at method level and class level were used as the classification input of neural network, respectively. The accuracy, F1 value, and AUC value of each detection method in different data sets are shown in Table 2.

Experimental results show that the accuracy of the proposed method reaches 95.08% in the case of different risk types in data sets. The F1 value reached 94.19%, and there was no significant difference in AUC value compared with other detection methods. However, it is superior to other detection methods in overall classification effect. Taking J48 method as an example, the accuracy of the proposed method is improved by 13.01%, and the F1 value is improved by 35.41%. Compared with Deodorant, a measurement and rule-based corporate risk detection tool, the method improved accuracy by 20.71% and F1 values by 82.1%. Combining machine learning-based corporate risk detection with measure-based corporate risk detection tool Deodorant, the paper's approach improves overall average accuracy by 21.25% and average F1 value by 61.65%.

Among the input factors of the model, the actual situation, growth situation, and cash flow have a strong relationship with the security of software enterprise risk. Because of the confidentiality of enterprise risk data, the model should minimize the occurrence of security problems. Therefore, the predictive model input can be made from these three factors in security analysis.

During the experiment, the enterprise risk data were grouped into 10 groups, and the problems of the three models were compared and analysed by using the test data. These are shown in Table 3 and Figure 4.

According to the experimental results, the model proposed in this study has fewer security problems in the process of prediction and has better performance than other

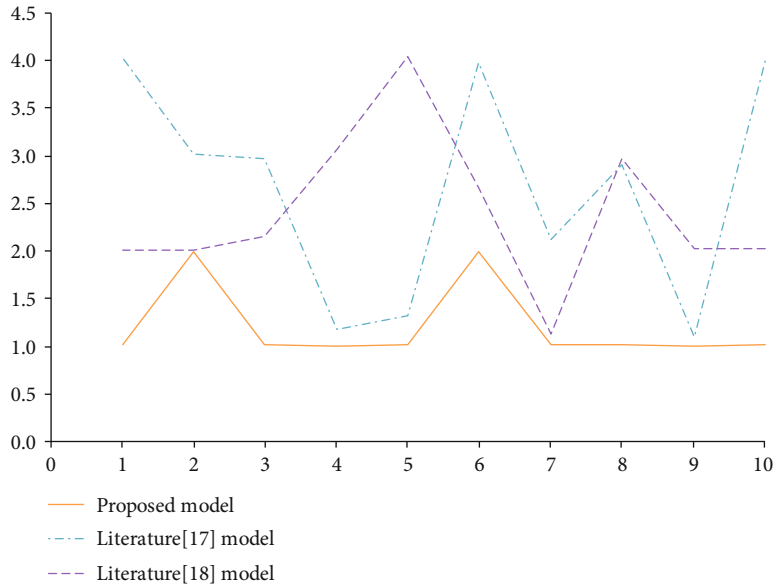

Figure 4: Security analysis diagram.

models. And it can reduce the occurrence of security problems to a certain extent.

5. Conclusions

In order to predict software enterprise risk more accurately and reduce enterprise loss, this paper proposes a method of enterprise risk detection based on BP neural network. In order to analyse the enterprise risk, firstly, the risk of the enterprise is modelled from the three dimensions of process dimension, time dimension, and logic dimension. As the input information of neural network, BP neural network is combined with common software risk items to extract the relationship between these risk items. Learn the complex relationship rules of input risk characteristics, to classify risks. Experimental results show that the risk prediction effect of the proposed algorithm is better than other comparison algorithms in terms of timeliness, accuracy, and security. However, in the actual risk detection, there is a great imbalance between the number of positive samples and the number of negative samples in the relevant data set collected, which affects the accuracy of the prediction results. Therefore, the next step is to detect risks based on generative adversarial neural network to solve the problem of unbalanced positive and negative samples in the data set.

Data Availability

The labelled data sets used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no competing interests.

Acknowledgments

This work is supported by the Soochow University.

References

[1] R. Hoda, N. Salleh, and J. Grundy, "The rise and evolution of agile software development," *IEEE Software*, vol. 35, no. 5, pp. 58–63, 2018.

[2] T. Menzies and T. Zimmermann, "Software analytics: what's next?," *IEEE Software*, vol. 35, no. 5, pp. 64–70, 2018.

[3] N. Nasri, S. Orts-Escolano, and M. Cazorla, "An semg-controlled 3d game for rehabilitation therapies: real-time time hand gesture recognition using deep learning techniques," *Sensors*, vol. 20, no. 22, p. 6451, 2020.

[4] S. Saeed, N. Z. Jhanjhi, M. Naqvi, and M. Humayun, "Analysis of software development methodologies," *International Journal of Computing and Digital Systems*, vol. 8, no. 5, pp. 446–460, 2020.

[5] P. Kruchten, R. Nord, and I. Ozkaya, *Managing Technical Debt: Reducing Friction in Software Development[M]*, Addison-Wesley Professional, 2019.

[6] A. M. Dima and M. A. Maassen, "From waterfall to agile software: development models in the IT sector, 2006 to 2018. Impacts on company management," *Journal of International Studies*, vol. 11, no. 2, pp. 315–326, 2018.

[7] A. K. Sangaiah, O. W. Samuel, X. Li, M. Abdel-Basset, and H. Wang, "Towards an efficient risk assessment in software projects-fuzzy reinforcement paradigm," *Computers & Electrical Engineering*, vol. 71, pp. 833–846, 2018.

[8] T. Deng, "Strengthening and upgrading of laboratory safety management based on computer risk identification," in *2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT)*, pp. 552–555, Chongqing, China, 2021.

[9] M. L. Fioravanti, B. Sena, L. N. Paschoal et al., "Integrating project based learning and project management for software engineering teaching: an experience report," in *Proceedings of*

the 49th ACM technical symposium on computer science education, pp. 806–811, Baltimore, MD, USA, February 2018.

[10] C. Tam, E. J. da Costa Moura, T. Oliveira, and J. Varajão, "The factors influencing the success of on-going agile software development projects," *International Journal of Project Management*, vol. 38, no. 3, pp. 165–176, 2020.

[11] X. Li, Q. Jiang, M. K. Hsu, and Q. Chen, "Support or risk? Software project risk assessment model based on rough set theory and backpropagation neural network," *Sustainability*, vol. 11, no. 17, p. 4513, 2019.

[12] N. Kriegeskorte and T. Golan, "Neural network models and deep learning," *Current Biology*, vol. 29, no. 7, pp. R231–R236, 2019.

[13] T. Li, J. Sun, and L. Wang, "An intelligent optimization method of motion management system based on BP neural network," *Neural Computing and Applications*, vol. 33, no. 2, pp. 707–722, 2021.

[14] F. Moshtaghian, M. Golabchi, and E. Noorzai, "A framework to dynamic identification of project risks," *Built Environment*, vol. 9, no. 4, pp. 375–393, 2020.

[15] A. S. Filippetto, R. Lima, and J. L. V. Barbosa, "A risk prediction model for software project management based on similarity analysis of context histories," *Information and Software Technology*, vol. 131, article 106497, 2021.

[16] C. Tian, J. Ma, C. Zhang, and P. Zhan, "A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network," *Energies*, vol. 11, no. 12, p. 3493, 2018.

[17] W. Zhang, Z. Wang, X. Zhu, W. Li, B. Gao, and H. Yu, "A risk assessment of a water-sand inrush during coal mining under a loose aquifer based on a factor analysis and the fisher model," *Journal of Hydrologic Engineering*, vol. 25, no. 8, p. 04020033, 2020.

[18] D. K. Nguyen and D. T. Vo, "Enterprise risk management and solvency: the case of the listed EU insurers," *Journal of Business Research*, vol. 113, pp. 360–369, 2020.