

Research Article

Engineering Cost Prediction Model Based on DNN

Bingxin Li , **Quanying Xin**, and **Lixin Zhang**

College of Urban Construction, Hebei Normal University of Science & Technology, Qin Huangdao 066004, China

Correspondence should be addressed to Bingxin Li; lbx2237@hevttc.edu.cn

Received 11 March 2022; Revised 13 May 2022; Accepted 30 May 2022; Published 6 July 2022

Academic Editor: Man Fai Leung

Copyright © 2022 Bingxin Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A DNN-based cost prediction method is proposed for the difficult problem of cost calculation in engineering cost accounting, combined with deep neural networks. Firstly, we introduce the basic information of artificial neural network and select the DNN structure to calculate the engineering cost price according to the characteristics of the data related to engineering cost price. Secondly, the DNN-based engineering cost price prediction model is constructed, and the two types of index systems, engineering characteristics and list item characteristics, are used as model inputs. In addition, the total quotation and each subitem engineering quotation and tax are used as model outputs by analyzing previous relevant studies. Based on this, simulation experiments are conducted on the DNN-based engineering cost price prediction model, and it is concluded from the training model that the DNN model has a better prediction effect. Among them, the relative error of total price forecast by DNN is 4.203%, and the relative error of integrated unit prices V1 and V2 is 2.98% and 4.52%, respectively, with small relative error. Finally, by reasonably adjusting the integrated unit price, the cost price of the integrated unit price and the cost price of the total offer can be calculated.

1. Introduction

In recent years, with the continuous improvement of China's economic level and comprehensive national power, construction projects have been developing, and more and more construction enterprises have been born. The surge of construction enterprises brings great pressure and challenges to the construction engineering market, and it is increasingly difficult for construction engineering to gain a foothold in the market and continue to grow and develop. With the slowdown of economic development, construction enterprises should enhance their competitiveness and base themselves on domestic and international markets. The primary task is to reduce the project cost scientifically. The cost of the enterprise is controlled within a reasonable plan to maximize the profit of the enterprise. Therefore, cost forecasting for construction enterprises is an inevitable trend in the current construction industry and is an extremely important task. However, most of the engineering cost forecasting methods on the market at present have problems such as unscientific methods and uncritical processes, which cannot effectively improve the competitiveness of enterprises and are not conducive to the long-

term development of construction enterprises [1–7]. In this regard, Xu et al. proposed to use BP neural network to predict the project cost of substation and introduced sparrow search algorithm (SSA) to optimize the parameters of traditional BP algorithm. The results show that this method has high prediction accuracy. Sharqi and Bhattarai compared several machine learning algorithms such as ELM and PLS and used them to predict the cost of field channels. The results show that the SC model has great potential. Xiaodong et al. used SIMCA-P algorithm to predict the cost of dynamic real estate projects. The results show that the average error of prediction is 0.006582346, showing high accuracy. However, the above methods mainly use the traditional machine learning algorithm.

Based on this research, a DNN-based engineering cost price prediction model is constructed by combining the widely used deep learning and giving full play to its learning and classification characteristics. Through this model, the accurate prediction and classification of construction project cost are realized. It provides a scientific and effective cost prediction method for the construction industry, which has a certain practical significance. The deep learning algorithm

to predict the project cost is also an innovation of this paper, giving full play to the characteristics of deep learning and improving the accuracy of prediction.

2. DNN

Compared with other neural network structures, the main feature of DNN is its excellent nonlinear processing capability. Thanks to the compact and efficient nonlinear mapping structure, DNN can handle mathematical and physical problems with larger datasets and more complex features. In addition, DNN can take full advantage of its own multiple hidden layer structure to train a large amount of data, and generally the accuracy of the results used for prediction will be higher. More layers indicate a more complex model, which has better nonlinear characteristics and can learn richer features. Theoretically, the links between the layers of the network structure are fully linked and the neurons of each layer can also be connected to each other [8–11]. Therefore, combined with experience, DNN is selected. The neural network structure of DNN is shown in Figure 1, which contains multiple hidden layers, an input layer, and an output layer.

As can be seen from the above figure, the DNN structure mainly consists of an input layer, a hidden layer, and an output layer. The network is characterized by the inclusion of multiple implicit layers. The input layer is represented as $X = [x_1, x_2, x_n]$, which is an n -dimensional column vector. The input data can be wind power, wind speed, wind direction, temperature, etc. In the input layer, the activation function is the standard constant function, and the input quantity needs to be transformed by the standard constant function and then output to the first layer. (w_n, b_n) denotes the weight parameter w_n for the n implied layers and the threshold parameter b_n . The data in the hidden layer are derived from the input of the upper layer. After nonlinear processing of the input variables using the activation function of this layer, the output of the processed data is then passed to the lower layer, and the final output combined with y is obtained, which is the value of wind power to be predicted.

The DNN can be expressed as follows [12–15].

The output value after the data into the classification water processing is transferred to the hidden layer using the input layer, and the first relationship of hidden layer input and output is obtained and expressed as

$$R_1 = f(w_1 \cdot X + b_1), \quad (1)$$

where R_1 denotes the output matrix of the first hidden layer and w_i and b_i denote the weight parameter and threshold parameter between the input layer and the first hidden layer, respectively.

If the first hidden layer variable is denoted as $r_{1,p}$, i.e., the p th variable, $w_{1,p,i}$ denotes the i th cause in the p th row of the weight matrix between the input layer and the first hidden layer, $b_{1,p}$ denotes the p th variable value in the threshold vector between the input layer and the first hidden layer, and then each output value in R_1 is the value obtained from the original column vector x transformed by the activation function f as follows:

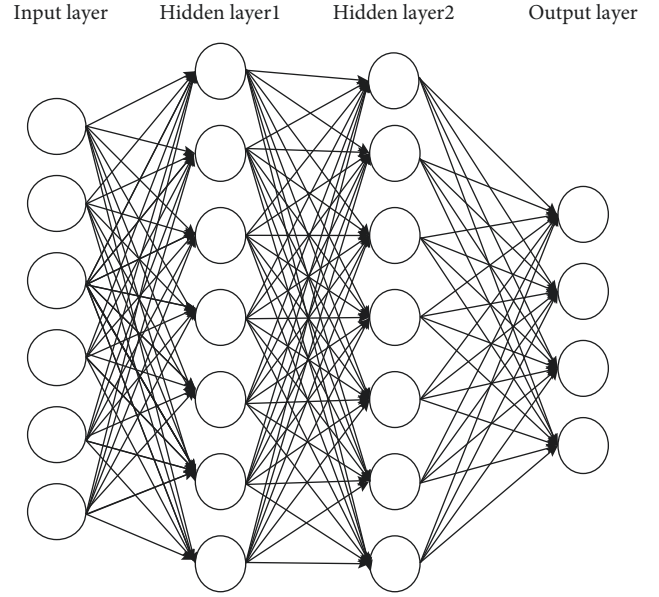


FIGURE 1: DNN with 2 hidden layers.

$$r_{1,p} = F \left(\sum_{i=1}^n w_{1,p,i} \cdot X_i + b_{1,p} \right). \quad (2)$$

According to the principle of DNN, the output of the previous hidden layer is the input of the next hidden layer, and then the output R_m expression of the m th hidden layer of the DNN model is

$$R_m = f(w_m \cdot R_{m-1} + b_m). \quad (3)$$

The input quantity X is processed by the input layer and transmitted to the hidden layer, and after performing the hidden layer processing, it is transmitted to the output layer, which can be expressed as

$$y = g(w_{n+1} \cdot R_n + b_{n+1}), \quad (4)$$

where w_{n+1} and b_{n+1} denote the weight parameter and the threshold parameter between the last hidden layer and the output layer, respectively, g denotes the activation function of output layer in DNN.

The activation function is a tool for nonlinear processing of the output quantity of each layer, which can enhance the data processing and optimization ability of neural networks. The common activation function sigmoid can be expressed as [16–19]

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}, \quad (5)$$

where x indicates the input variable and e indicates the constant, which takes the value of 2.7183.

3. Cost Prediction Model Construction Based on DNN

Combining the good nonlinear mapping ability and strong generalization ability of DNN, this paper uses the DNN model to predict engineering cost.

3.1. Inputs to the DNN Model. There are many factors affecting the cost price of construction projects, and these factors influence and relate to each other. In order to analyze the project cost comprehensively, this paper refers to the research results of some scholars and divides these influencing factors into engineering characteristics and list item characteristics.

3.1.1. Engineering Characteristics. For the engineering characteristics, the indicators are divided as shown in Table 1.

3.1.2. List Item Characteristics. The list items are usually coded with 12 bits, and the first 9 bits of the code are set according to the different categories. Based on this, conversions are made between the inventory codes and the model. The last 3 bits of the code are coded in order from 001 according to the specific conditions of the construction drawings, and the item code cannot be repeated.

The list item characteristics are the same as the engineering characteristics, which need to be classified and given the corresponding values according to the description of the item characteristics, and if there is no subitem in the list, it will be given 0. Due to the large amount of inventory item data, quantification of the entire engineering characteristics has great complexity, so this paper takes the two subprojects of solid brick walls and beamed slabs as examples for quantification, as shown in Table 2.

3.2. Outputs of the DNN Model. When judging the project cost, we usually use methods such as checking and comparing to reasonably judge the total quotation and each subitem project quotation and tax, among which the total quotation and subitem project quotation are the main judging contents. For this feature, it is used as the output of the DNN model to reasonably predict the price, and thus a reference is provided for the expert's evaluation. In summary, the inputs and outputs of the DNN model are shown specifically in Table 3.

3.3. DNN Structure Design

3.3.1. Network Parameter Selection

(1) Activation Functions. In order to make the DNN model more expressive, a nonlinear function needs to be introduced as the activation function. The commonly used activation functions include sigmoid function, ReLU function, and tanh function. Since the DNN model is prone to gradient disappearance or explosion when reverse transfer is performed, the ReLU function can solve this problem, and the convergence speed of the ReLU function is faster, so the ReLU function is used as the activation function in this paper. Its function image is shown in Figure 2.

(2) Loss Functions. The loss function is used to represent the difference between the true value Y and the predicted value $f(x)$ of the model. If the loss function is smaller, the model

TABLE 1: Engineering characteristic indicators and categories.

	Building height X_1
	Total building area X_2
	Standard floor building area X_3
	Standard floor height X_4
	Number of floors above ground X_5
	Basement floor height X_6
	Number of basement floors X_7
Engineering characteristic influencing factor	Basement floor area X_8
	Floor area X_9
	Construction period X_{10}
	Building type X_{11}
	Structure type X_{12}
	Foundation form X_{13}
	Foundation soil type X_{14}
	Type of door and window X_{15}
	Site condition X_{16}

TABLE 2: Quantification of solid brick wall and beam slab project.

Project code	Project name	Project characteristics
0103022001	Solid brick wall	Quantity of work U10
		Brick variety U11
		Wall type U12
		Mortar strength U13
106062001	Beam slab	Quantity of work U20
		Concrete type U21
		Strength grade U22

has stronger fitting ability and higher precision. In general, the loss function is denoted by $L(Y, f(x))$, and its standard form is [20]

$$L(Y, f(x)) = \sum (Y - f(x))^2. \quad (6)$$

(3) Weight Initialization. Before the model is trained, the weights need to be initialized. The current weight initialization methods include two main types: one is to initialize the weights to very small values, and the other is to set the weights to +1 and -1 with equal numbers. Considering that the second method is too subjective, in order to reach better effect of the model training, the first method is used in this paper to initialize the weights with random numbers in the range of $[-1, 1]$ that fit the normal distribution.

(4) Network Structure. The network structure mainly includes network nodes and hidden layer, which directly affect the performance of DNN. Generally, only one hidden layer is considered, and under this hidden layer, the optimal hidden node is found. Figure 3 shows the error of different nodes under a hidden layer after 500 calculations. According to the results of Table 4, when the hidden node is 23, the error is the smallest, but it is still difficult to meet the requirements.

Generally speaking, the more the hidden layers, the higher the accuracy of the model, so a DNN model with 2 hidden layers is used. After determining the number of hidden layers, the number of hidden layer nodes is

TABLE 3: Input and output parameters of the DNN model.

Input	Output	
Building height X_1	Total quotation Y	
Total building area X_2		
Standard floor building area X_3		
Standard floor height X_4		
Number of floors above ground X_5		
Basement floor height X_6		
Number of basement floors X_7		
Basement floor area X_8	Integrated unit price $V1$	
Floor area X_9		
Construction period X_{10}		
Building type X_{11}		
Structure type X_{12}		
Foundation form X_{13}		
Foundation soil type X_{14}		
Type of door and window X_{15}		
Site condition X_{16}		
0103022001 (solid brick wall)	Quantity of work U_{10} Brick variety U_{11} Wall type U_{12} Mortar strength U_{13}	Integrated unit price $V2$
106062001 (with beam slab)	Quantity of work U_{20} Concrete type U_{21} Strength grade U_{22}	

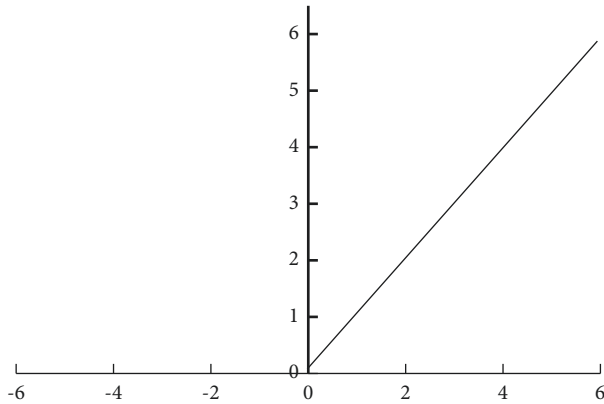


FIGURE 2: ReLU function image.

determined. Generally speaking, if the number of nodes in the hidden layer is insufficient, it will cause poor learning ability of the model, thus increasing the error; if it is too large, it will cause overfitting phenomenon and reduce the generalization ability of the model. By analyzing the previous experience, the optimal ratio of the number of nodes in first hidden layer to that in second hidden layer is 3 : 1. Therefore, when the number of nodes in the two layers is set to 15 and 5, respectively, the final DNN structure is shown in Figure 3 [21–24].

4. Simulation Experiments

4.1. DNN Model Training

4.1.1. Model Training Samples and Environment. The sample data of this model come from an engineering cost website, in which nearly 20 sets of engineering cost data from a certain

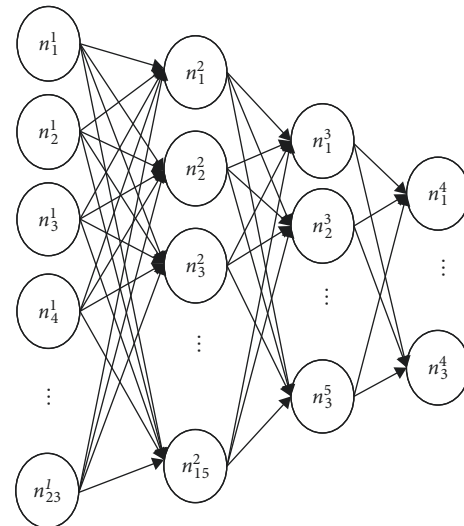


FIGURE 3: Schematic diagram of DNN structure.

region are selected as samples, including 15 sets of training samples and 5 set of testing samples. The model writing procedure uses the Spyder software in Python environment, and the model is trained using an improved BP algorithm that introduces a momentum term, which allows the error surface to avoid falling into local minimum and thus obtains a more optimal solution. Let the learning rate η be 0.001, momentum coefficient α be 0.8, iterations be 5000, and the error range ε be less than 0.001.

4.1.2. Model Training and Results. Before conducting the model training, the sample data need to be normalized to control the data output and input within the interval, and the

TABLE 4: Relationship curve between node and error under single hidden layer.

Number of hidden layer nodes	Error (%)
10	0.83
11	0.52
12	0.67
13	0.48
14	0.49
15	0.55
16	0.8
17	0.7
18	0.45
19	0.53
20	0.3
21	0.27
22	0.38
23	0.3
24	0.32
25	0.54
26	0.39
27	0.46
28	0.34
29	0.36
30	0.42

error value is about 0.00289 after 5000 iterations of this model calculation, as shown in Figure 4. The fitted curves of the total quotation and the integrated unit price of the branch subitem project are shown in Figure 5, from which it can be seen that the curve fitting effect of the sample value and the actual value is consistent.

In order to compare the prediction effect of DNN, BP neural network is introduced while DNN prediction is carried out. The comparison results show that the prediction of DNN is closer to the real price, while the error between BP neural network and the real value is large. It is concluded that the prediction effect of DNN is better.

In order to verify the effectiveness of the DNN model in practical applications, the trained DNN model is used to test the test sample, i.e., by analyzing the test sample error, if it is within the error range, it means the DNN model has validity. In this paper, a set of test samples is randomly selected for testing, and the output values are shown in Table 5.

As can be seen from the above table, the relative errors between the output values of DNN model and the true values are all below 5%, which indicates that the DNN model can predict the engineering cost well.

4.2. Determination of Cost. The project cost has been predicted above, but the prediction result is not the cost price because (1) the model sample is random and does not consider factors such as individual enterprise management ability, so the predicted value of the model only belongs to the social average and (2) when the model is trained, its output value is the settlement price, including the enterprise profit, while the cost price should be the difference between the settlement price and the profit. Therefore, the model predictions need to be adjusted in order to obtain the cost price.

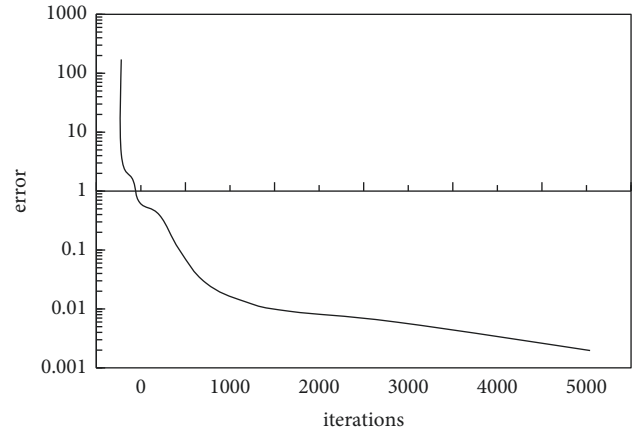


FIGURE 4: Model training error convergence.

4.2.1. Cost Price of Branch Subitem Project. Considering the characteristics of the branch subitem project cost and the proportion of the integrated unit price, this paper adopts the weighted average algorithm to multiply the integrated unit price by the adjustment factor to obtain the cost price, and the adjustment method is shown in Table 6. Assuming that P denotes the cost price and V denotes the model output value, the adjustment equations are shown in equations (7) and (8) [25–29].

$$\text{Adjustment coefficient } \phi = \sum_{i=1}^5 \alpha_i \beta_i, \tag{7}$$

$$\text{cost price } P = V \phi. \tag{8}$$

Because of the different proportions of the various costs of the integrated unit price, statistics on the percentage of the integrated unit price costs α are required. According to the first 9 digits of the code, the computer finds the matching list items, and using Table 6 as an example for proportional calculation, we can get the percentage of labor cost α_1 as follows:

$$\alpha_1 = 0.207. \tag{9}$$

In the same way, $\alpha_2 = 0.657$, $\alpha_3 = 0.008$, $\alpha_4 = 0.025$, and $\alpha_5 = 0.103$. Then, the cost coefficients of the integrated unit price are calculated. According to the relevant research, the labor costs, material costs, construction tool usage costs, and enterprise management costs are generally not lower than the average social components of 90%, 98%, 70%, and 70%, respectively, while some enterprises can make use of zero profit to expand the market, so the profit can be ignored. In this regard, the integrated unit cost factors are shown in Table 7.

Substituting the above table data into formula (7), the integrated unit price adjustment factor of solid brick wall can be calculated as 0.85. Since the integrated unit price output of the solid brick wall is \$468, the cost price is

$$P_1 = V_1 \phi_1 = 468 \times 0.85 = 397.8 \text{ yuan}. \tag{10}$$

According to the above method, each integrated unit price can be adjusted to obtain the corresponding cost price.

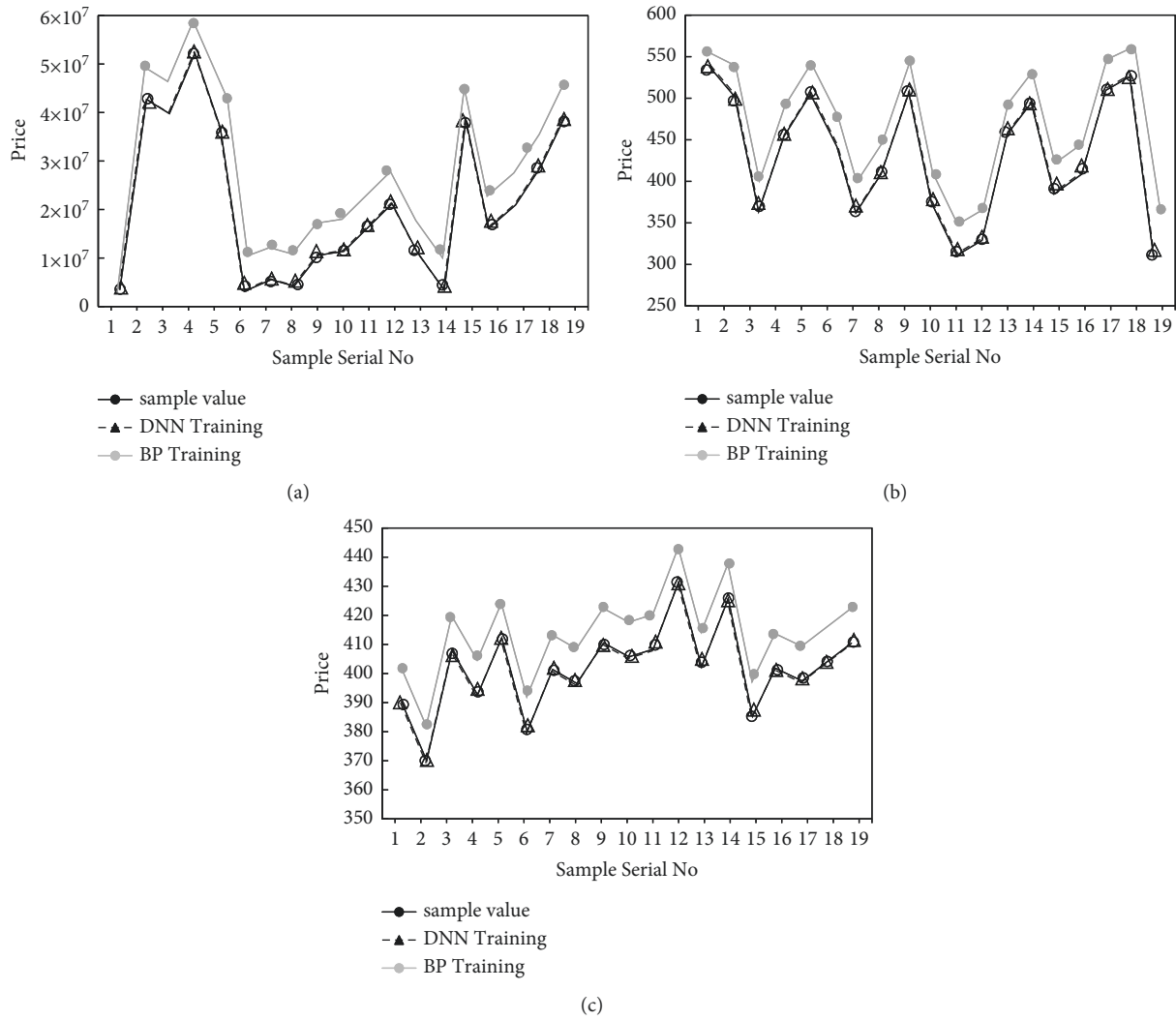


FIGURE 5: DNN model curve fitting effect. (a) Total quotation fitting. (b) Integrated unit price V1 fitting. (c) Integrated unit price V2 fitting.

TABLE 5: Model validation result.

Output variable	Output value	True value	Difference value	Relative error
Total quotation Y	40 423 575	42 197 507	1773932	4.203%
Integrated unit price V_1	488	503	15	2.98%
Integrated unit price V_2	359	376	17	4.52%

TABLE 6: Integrated unit price adjustment method.

Cost components	Labor costs	Material costs	Machinery costs	Management costs	Profit
Cost proportion	α_1	α_2	α_3	α_4	α_5
Cost coefficient	β_1	β_2	β_3	β_4	β_5

TABLE 7: Coefficient adjustment calculation.

Cost components	Labor costs	Material costs	Machinery costs	Management costs	Profit
Cost proportion	0.207	0.657	0.008	0.025	0.103
Cost coefficient	0.85	0.98	0.70	0.70	0

The total cost price of the total quotation can be obtained by aggregating the adjusted cost prices.

5. Conclusion

In summary, the key to solving the cost pricing problem lies in the accurate calculation of the project cost price. Based on previous research, this paper constructs a DNN engineering cost price prediction model, divides the engineering index system into two categories: engineering characteristics and project characteristics, and then randomly selects 20 sets of sample data to train and predict the model. The results show that the DNN prediction model has less error and can effectively predict the engineering cost. Since the engineering cost prediction is random in nature and the result output is the settlement price, the DNN prediction result is reasonably adjusted. Specifically, the weighted average algorithm is used to calculate the cost price of the integrated unit price of each subitem project and the cost of the total quotation, and finally the cost price of the whole project is obtained. Also, the study provides new ideas for the information budget of engineering cost.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] P. Pandey, N. D. Gundi, P. Basu et al., "Challenges and Opportunities in Near-threshold DNN accelerators around Timing errors," *Journal of Low Power Electronics and Applications*, vol. 10, no. 4, p. 33, 2020.
- [2] C. Tang, N. Luktarhan, and Y. Zhao, "SAAE-DNN: deep learning method on Intrusion Detection," *Symmetry*, vol. 12, no. 10, p. 1695, 2020.
- [3] W. Zhou and Z. Zhu, "A novel BNMF-DNN based speech reconstruction method for speech quality evaluation under complex environments," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 4, pp. 959–972, 2020.
- [4] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," in *Proceedings of the 2018 IEEE international conference on acoustics speech signal processing*, pp. 1927–1930, IEEE, Calgary, AB, Canada, April 2018.
- [5] Y. Shanshan, L. Jian, and C. Lin, "Low-thrust spacecraft trajectory optimization via a DNN-based method[J]," *Advances in Space Research*, vol. 66, no. 7, pp. 1635–1646, 2020.
- [6] G. Yue and O. Takashi, "Co-design of Binary processing in Memory ReRAM Array and DNN model optimization algorithm:Regular Section[J]," *IEICE - Transactions on Electronics*, vol. E103.C, no. 11, pp. 685–692, 2020.
- [7] M. C. Pravin, M. Mukilan, G. Vishnu Prakash, P. Nithish, B. Monish Kanna, and E. Logesh, "Predicting the Emissive characteristics of an IC engine using DNN," *IOP Conference Series: Materials Science and Engineering*, vol. 995, no. 1, Article ID 012010, 2020.
- [8] X. Xu, L. Peng, Z. Ji, S. Zheng, Z. Tian, and S. Geng, "Research on substation project cost prediction based on sparrow search algorithm optimized BP neural network," *Sustainability*, vol. 13, no. 24, Article ID 13746, 2021.
- [9] S. S. Sharqi and A. Bhattarai, "Evaluation of several machine learning models for field canal improvement project cost prediction [J]," *Complexity*, vol. 2021, Article ID 8324272, 2021.
- [10] T. Xiaodong, X. Tang, and X. Zhang, "Dynamic real estate project cost prediction based on SIMCA-P[J]," *IOP Conference Series: Earth and Environmental Science*, vol. 526, no. 1, Article ID 012178, 2020.
- [11] A. Yousefi, H. Guo, M. Birouk, B. Liko, and S. Lafrance, "Effect of post-injection strategy on greenhouse gas emissions of natural gas/diesel dual-fuel engine at high load conditions [J]," *Fuel*, vol. 32, p. 290, 2021.
- [12] K. Li, X. Chen, R. Zhang, and E. Pickwell-MacPherson, "Classification for Glucose and Lactose Terahertz Spectrums based on SVM and DNN methods," *IEEE TRANSACTIONS ON TERAHERTZ SCIENCE AND TECHNOLOGY*, vol. 10, no. 6, pp. 617–623, 2020.
- [13] H. Han, L. Xu, X. Cui, and Y. Fan, "Novel Chiller fault Diagnosis using deep neural network (DNN) with simulated Annealing (SA)[J]," *International Journal of Refrigeration*, vol. 121, pp. 269–278, 2020.
- [14] C. Xiao, X. Wang, Q. Chen, F. Bin, Y. Wang, and W. Wei, "Strength Investigation of the Silt-based Cemented Paste Backfill using Lab experiments and deep neural network[J]," *Advances in Materials Science and Engineering*, vol. 2020, Article ID 6695539, 2020.
- [15] R.-E. Mihaescu, M. Chindea, C. Paleologu, S. Carata, and M. Ghenescu, "Person Re-Identification across data distributions based on general Purpose DNN Object Detector," *Algorithms*, vol. 13, no. 12, p. 343, 2020.
- [16] Y. Ibrahim, H. Wang, J. Liu et al., "Soft errors in DNN accelerators: a comprehensive review," *Microelectronics Reliability*, vol. 115, Article ID 113969, 2020.
- [17] A. Makuvaza, D. S. Jat, and A. M. Gamundani, "Deep neural network (DNN) Solution for real-time Detection of distributed Denial of Service (DDoS) Attacks in software Defined networks (SDNs)[J]," *SN Computer Science*, vol. 2, no. 2, 2021.
- [18] J. Chen, H. Zheng, H. Xiong et al., "FineFool: a novel DNN Object Contour Attack on image recognition based on the Attention Perturbation Adversarial Technique[J]," *Computers & Security*, vol. 104, Article ID 102220, 2021.
- [19] K. Inoue, S. Hara, M. Abe, N. Hojo, and Y. Ijima, "Model architectures to extrapolate emotional expressions in DNN-based text-to-speech," *Speech Communication*, vol. 126, pp. 35–43, 2021.
- [20] A. K. Mishra and V. Ponnusamy, "DNN-based distributed sequential uplink processing in cell-free massive MIMO based on radio stripes[J]," *IET Networks*, vol. 10, no. 3, 2021.
- [21] A. Yazid, C. Benoit, and Z. Wei-Ping, "DNN-based Calibrated-Filter models for speech Enhancement[J]," *Circuits, Systems, and Signal Processing*, vol. 40, pp. 1–24, 2021.
- [22] H. Deng and C. T. Albert, "A Parametric level set method for Topology optimization based on deep neural network[J]," *Journal of Mechanical Design*, vol. 143, no. 9, 2021.
- [23] S. Li, J. Hou, A. Yang, and J. Li, "DNN-based distributed Voltage Stability online Monitoring method for large-Scale power Grids[J]," *Frontiers in Energy Research*, vol. 9, 2021.
- [24] D. Ding, L. Wang, Z. Yang, K. Hu, and H. He, "ACIMS: Analog CIM simulator for DNN Resilience," *Electronics*, vol. 10, no. 6, p. 686, 2021.

- [25] T. Yao, F. Gao, Q. Zhang, and Y. Ma, "Multi-feature gait recognition with DNN based on sEMG signals," *Mathematical Biosciences and Engineering: MBE*, vol. 18, no. 4, pp. 3521–3542, 2021.
- [26] Z. Yonggang, X. Yuanlun, Z. Yan, Q. Junbo, and W. Sunxin, "The adoption of deep neural network (DNN) to the prediction of soil liquefaction based on shear wave velocity[J]," *Bulletin of Engineering Geology and the Environment*, vol. 80, pp. 5053–5060, 2021.
- [27] W. M. Zhang, L. Zhang, Z. Zhang, and M. Sun, "IBD1: the metrics and evaluation method for DNN processor benchmark while doing Inference task[J]," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 5, pp. 1–13, 2021.
- [28] S. Jung, J. H. Bong, S.-J. Kim, and S. Park, "DNN-based FES control for gait Rehabilitation of Hemiplegic Patients[J]," *Applied Sciences*, vol. 11, no. 7, p. 3163, 2021.
- [29] D. Preethi and N. Khare, "An Intelligent network Intrusion Detection system using Particle Swarm optimization (PSO) and deep network networks (DNN)[J]," *International Journal of Swarm Intelligence Research*, vol. 12, no. 2, pp. 57–73, 2021.