An Enterprise Investment Method Integrated with Improved Feature Optimization and Stochastic Configuration Networks

Qiang Zhang[2,1,*], DongQiang Wang[1,2]

*Corresponding author. Email: 779425065@qq.com

CCCC FIRST HIGHWAY CONSULTANTS CO.LTD, China[1]
Department of Artificial Intelligence, Chongqing University of Technology, China[2]

Abstract. To solve the problem that enterprise investment analysis is challenging to analyze efficiently and accurately, this paper proposed a new method based on artificial intelligence method to process amounts of data and make behavioral predictions. Efficient investment promotion has a great impact on the future economic benefits of a company or industrial park. The data used in this paper is derived from the CCCC Industry Operational Platform. Firstly, a multidimensional feature set of enterprise data is gathered and defined that a shareholding ratio greater than 50% indicates investment behavior by the enterprise. Secondly, normalization methods are adopted to organize the multidimensional feature set, reducing differences in magnitude between variables and optimizing model computational efficiency. Finally, the SCN algorithm is utilized to predict enterprise investment behaviors. Finally, taking two industries as the training data. The experiment shows that the precision of predicting the investment behavior of scientific research enterprises reaches 93.95%, and the precision of predicting the investment behavior of information transmission enterprises reaches 90.45%. Meanwhile, comparative experiments have proven that the method proposed in this article not only directly improves the efficiency of the enterprise investment analysis process, but also enhances the accuracy of enterprise investment analysis.

Keywords: Machine Learning, Stochastic Configuration Networks, Data Mining.

1 INTRODUCTION

As for investment in Chinese industrial parks, enterprise investment is both the driving force behind and the outcome of government investment attraction. Therefore, it is crucial to understand why enterprises decide to invest, in order to identify which types of enterprises are more likely to do so. Business professionals in industry research and park investment have historically relied on traditional analytical methods to evaluate industry trends and forecast enterprise investment behavior. This approach is time-consuming and prone to human error, leading to imprecise analytical outcomes. Recently, several scholars have explored the practical applications of machine learning and deep learning methodologies in macroeconomic and microeconomic analyses, highlighting their predictive superiority over other econometric models [5,9,10].

ICBBEM 2023, May 19-21, Hangzhou, People's Republic of China

DOI 10.4108/eai.19-5-2023.2334425

Based on previous research, this paper proposes a model that combines artificial intelligence algorithms with enterprise behavior analysis. The training dataset is obtained from the CCCC Industry Operational Platform and contains multi-dimensional information on enterprises. This model seeks to enhance the accuracy and efficacy of enterprise behavior analysis by using advanced machine learning techniques.

Effective features used to identify potential investors and provide investment attraction personnel with insights to screen prospective customers. After constructing the dataset, quantile transformation [11] is utilized to map the data into a uniform distribution for standardized feature weights. In addition, PCA [6] is used to aggregate and reduce the dimensions of the features, eliminating irrelevant information. To facilitate iterative optimization of the model, SCN is employed as the backbone network. This approach replaces traditional business personnel with AI methods to analyze enterprise investment behavior, which enhances analytical efficiency and accuracy while minimizing potential subjective factors. The proposed model achieves an accuracy of 93.95% in analyzing investment behavior for scientific research enterprises and 90.45% for information transmission enterprises, resulting in an overall accuracy of 92.7%. The main contributions of this study are:

1. Establishing the method which enterprise investment behavior data set, which gathers and integrates the enterprise characteristic set of investment behavior and non-investment behavior.

2. A feature optimization scheme for numerical value and dimension, which reduces dimension and optimizes multi-dimensional features of enterprises, significantly improving the reasoning speed and accuracy of the model.

3. Enterprise investment behavior prediction model applicable to enterprise data sets shows robust prediction performance for investment behavior after training, and can be used to predict whether other enterprises will have investment behavior.

2 BASIC OF ANALYSIS MODEL

For the input data set $Y = \{X_i \in R^F\}_{i=1,2,\ldots,n}$, where n is the number of enterprise records included in the data set and each enterprise contains F pieces of information. Given the unique nature of enterprise data, no similar feature data sets are available for training and learning currently. Therefore, we first integrate the terminal data of the CCCC Industry Operational Platform to construct an enterprise feature set as the input of the model. Furthermore, nonlinear mapping methods were employed to normalize and standardize the scale of enterprise data. Finally, the output of this method was used as the SCN algorithm's input to fit and obtain classification results on enterprise investment behavior. To facilitate subsequent discussions, all data objects are represented as matrices. Typically, data is represented as $X_i \in R^{N\times F}$, and features are represented as $F_X \in R^{N\times C}$.

2.1 Dataset construction

The data collected in the data terminal of the CCCC Industry Operational Platform often accompanied with noise. To solve this problem, the commonly used data cleaning techniques in big data are employed to clean the massive enterprise data [4]. The main objectives of data cleaning process were as follows: (1) detecting and removing abnormal data; (2) detecting and

removing nearly duplicate data; (3) cleaning data that does not conform to the final format; (4) cleaning data with relational issues. As shown in **Figure 1**.

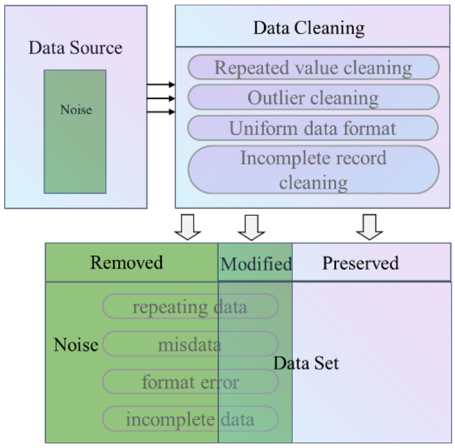

Figure 1. general view of data cleaning from data source

Before data cleaning, selecting scientific research-oriented and information transmission-oriented enterprises as the target of our research to avoid inaccurate prediction results caused by prior knowledge errors due to differences in investment planning concepts in various industries. (1) To deal with incomplete data, we graded fields and utilized computational analysis methods to fill in low-weight fields. If a business record does not contain a high-weight field, we deleted it from the dataset in order to prevent inaccurate information from affecting model fitting. (2) Concerning erroneous data, we utilized deviation analysis in statistical methods to eliminate outliers, followed by filtering rules in the database to exclude values that did not meet requirements. The following formula summarizes this process:

$$\frac{\Sigma_{k=1}^{n}\left|x_i^k - M^k / MAD^k\right|}{n} > 2 \tag{1}$$

In the equation (1), a single field value x_i has k dimensions, n is represents the dimensions of enterprise data, M_i is the median of k dimensions, and MAD is the median absolute deviation value [3]. Records that meet the above equation are removed as outliers to improve the effectiveness of the dataset. (3) To eliminate duplicate data, we utilized predefined duplicate identification rules to remove redundant enterprise records. (4) For inconsistent data formats, we processed each field with a fixed field format.

After the data was cleaned, enterprise information datasets of 10,000 records each were constructed for information transmission-oriented and scientific research-oriented enterprises, and each record contained 39 field information as the initial feature set.

2.2 Feature optimization method

In this study, there is a large difference in the order of magnitude of enterprise feature sets, with features such as registered capital reaching tens of thousands while others such as the number

of insured persons or intellectual property remain in the units or tens. When a feature's variance is significantly larger than that of other features, it can dominate the learning algorithm and hinder its ability to discover patterns elsewhere. To address this issue, feature normalization can be employed to make features between different dimensions comparable in value, thereby greatly enhancing the accuracy of the classifier.

Traditional linear regression models make assumptions about the linear distribution relationship of the different dimensions of the enterprise feature set. However, such an assumption is not valid considering the presence of peaked or heavy-tailed distributions and significant heteroscedasticity. The model presented in this paper utilizes the quantile transformation method that employs a nonlinear mapping technique to transform the original data into a uniform distribution with values ranging between 0 and 1. By applying the quantile transformation, the abnormal distributions can be smoothed out and the absolute relationship of values can be mapped to relative relationships.

In this paper, x represents the enterprise feature set, denoted as $x = \{x_1, x_2, x_3 \dots, x_n\}$, where n=39. Here, x_1 represents the number of years since establishment, x_2 represents the amount of registered capital, and so on. The formula for the quantile transformation method is as follows:

$$P_k = 1 + (n-1) * p_k \tag{2}$$

$$Q_k = x_{\lfloor P_k \rfloor} + (x_{\lfloor P_k \rfloor + 1} - x_{\lfloor P_k \rfloor}) \times (P_k - \lfloor P_k \rfloor) \tag{3}$$

$$X_{Opt} = G^{-1}(X) = {}^{X - Q_0}/_{Q_k - Q_0} \tag{4}$$

In the equation, $p_k \in p_1, p_2, p_3 \dots, p_k$ are pre-defined quantiles, such as quartiles $(0.25, 0.5, 0.75)$. In equation (1), P_k is the position of the k-th quantile in the sorted feature set. In equation (2), the corresponding quantile Q_k is calculated based on P_k. In equation (3), the dataset X is mapped to $U(0,1)$. The processed enterprise feature set X_{Opt} is inputted as the optimized feature into the SCN algorithm model for analysis.

2.3 Basic of Stochastic Configuration Networks

Stochastic Configuration Networks (SCN) [7] is a type of supervised neural network with random weights. It differs from conventional BP networks in that it can begin with a small network that requires minimal human input. With randomly selected input weights and thresholds, the SCN can gradually increase the number of hidden layer nodes while updating the weights and thresholds until the termination condition of training accuracy is met. This approach enables more automatic and flexible network configuration compared to traditional methods.

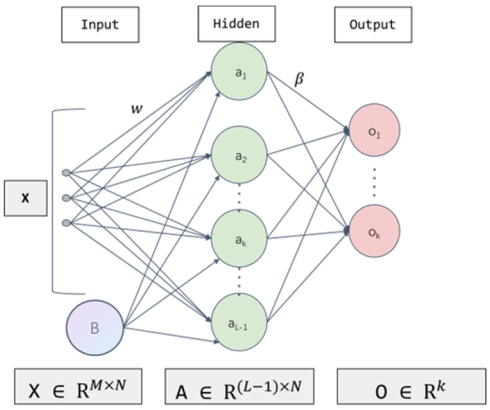

Figure 2. illustration of the SCN structure

As shown in the **Figure 2**, the SCN model has $L-1$ hidden layer nodes, M input layer neurons, K output layer neurons. X represents the input sample matrix of the network and is obtained by normalizing the features.

$$X = \{X_1, X_2, \cdots, X_N\} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{bmatrix} \tag{4}$$

In equation (4), M and N represent the number and dimension of input samples, respectively. In this study, M represents the number of enterprises in the feature set, and N represents the number of enterprises feature types, with $N = 39$. $X_n = (x_{1n}, x_{2n}, \cdots, x_{Mn})^T$ is the n-th input sample, and $T_n = (t_{1n}, t_{2n}, \cdots, t_{Kn})^T$ is the expected output of the n-th sample. As in equation (5), the weight matrix between the input layer and the hidden layer is denoted by W. It is important to note that all numerical data have been normalized before computing.

$$W = \{W_1, \cdots, W_N\} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1L-1} \\ w_{21} & w_{22} & \cdots & w_{2L-1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{ML-1} \end{bmatrix} \tag{5}$$

The threshold matrix B with hidden layer nodes is defined as $B = [b_1, b_2, \cdots b_{L-1}]^T$, where b_j represents the threshold of the j-th hidden neuron. $g(\bullet)$ is the activation function of the hidden layer neurons, and $g_j = [g_{1j}, g_{2j}, \cdots g_{jN}]^T$ represents the output of the j-th hidden layer neuron. Specifically,

$$\mathrm{g}_j = \mathrm{g}\left(w_j^T X + b_j\right) = \frac{1}{1+exp\left(-w_j^T X + b_j\right)} \tag{6}$$

Where in equation (6), w_j is the connection weight between the input node and the j −th hidden node. The output matrix of the hidden layer nodes is denoted as equation (7).

$$H = [g_1, g_2, \cdots, g_{L-1}]^{\mathrm{T}} = \begin{bmatrix} g_{1,1} & g_{1.2} & \cdots & g_{1,N} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ g_{L-1,1} & g_{L-1,2} & \cdots & g_{L-1,N} \end{bmatrix} \tag{7}$$

The connection weight between the $j-$th hidden node and the $k-$th output node is denoted as β_{jk}, and the complete matrix is denoted as equation (8).

$$\beta = [\beta_1, \beta_2, \cdots, \beta_{L-1}]^T = \begin{bmatrix} \beta_{1,1} & \beta_{1.2} & \cdots & \beta_{1,K} \\ \beta_{2,1} & \beta_{2,2} & \cdots & \beta_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{L-1,1} & \beta_{L-1,2} & \cdots & \beta_{L-1,K} \end{bmatrix} \tag{8}$$

The actual output of the network is represented by $O = [o_1, o_2, \cdots, o_K]^T$. Thus, the output of the randomly configured network with the structure of $M-(L-1)-K$ can be denoted as equation (9).

$$O = \beta^T H \tag{9}$$

Given the objective function $f: \mathbb{R}^M \to \mathbb{R}^K$, the current output of the network is

$$f_{L-1}(X) = \beta^T H = \sum_{j=1}^{L-1} \beta_j g_j\left(w_j^T X + b_j\right)(L = 1,2,\cdots; f_0 = 0) \tag{10}$$

The current residual of the network is

$$e_{L-1} = f - f_{L-1} = \left[e_{L-1,\ 1},\ e_{L-1,\ 2}, \cdots e_{L-1,\ K}\right] \tag{11}$$

In equation (10) and (11), if the output residue $\|e_L - 1\|$ of the current network fails to meet the preset error requirements, the network will select new hidden layer neuron nodes based on inequality constraints to obtain the parameters g_L (w_L and b_L) of the $L-$th hidden layer node. Inequality constraints are used to select hidden layer node parameters in the SCN, and they are represented as:

$$\xi_{L,K} = \left(\frac{(e_{L-1,k}(X)^T \cdot g_L(X))^2}{g_L(X)^T \cdot g_L(X)} - (1 - r - \mu_L) e_{L-1,k}(X)^T \cdot e_{L-1,k}(X)\right) \geq 0 \tag{12}$$

In equation (12), $g_L = g_L(w_L^T X + b_L)$ denotes the output of the $L-$th hidden node, e_{L-1} represents the output residue when constructing $L-1$ hidden layer nodes, $0 < r < 1$ can change during the parameter selection process, and $\mu_L = \frac{1-r}{L+1}$ is a non-negative real number sequence. From this constraint, it can be seen that the selection of network parameter weights and thresholds is related to the distribution of the given training samples. The output weight of the $L-$th hidden layer node of the network can be obtained based on formula (14), which is denoted as follows:

$$\beta_{L,k} = \frac{\langle e_{L-1,k}, g_L \rangle}{\|g_L\|^2}, k = 1,2, \ldots K \tag{13}$$

After calculating the L −th hidden layer node, the output of the network is obtained as follows:

$$f_L(X) = f_{L-1}(X) + \beta_L g_L(w_L^T X + b_L) \tag{14}$$

The next step is to verify the output error e_L of the network and evaluate whether it complies with the predetermined error criteria. If it satisfies the requirement, the construction of the SCN is considered finalized. Otherwise, the network incorporates additional hidden layer neuron node parameters according to inequality constraints, as shown in equation (13), in order to minimize the output error, until the stopping condition is reached.

3 EXPERIMENTS

This section presents experiments conducted on the enterprise feature set created in section 2.1, aiming to evaluate the efficiency of the feature optimization method and backbone network structure for machine learning. Quantitative analysis was employed, and various models were compared based on evaluation metrics that included training accuracy, testing accuracy, and testing time. The hyperparameters used in the SCN model for this section were set as follows: the maximum number of generated hidden nodes is 300, and the maximum number of random configurations is 100 times. In the table presented in this chapter, A represents normalization [0,1], B represents normalization [-1,1], C represents Z-score normalization, D represents equidistribution mapping, E represents PCA (20Dim), F represents PCA (40Dim), G represents SVM, H represents TextCNN, I represents MLP and J represents SCN.

3.1 Feature optimization method

In this section, SCN was used as the backbone network for all models, based on the dataset of scientific research-oriented enterprises. Unreasonable feature distribution can lead to problems such as local optima and gradient disappearance in prediction results, making feature optimization an essential part of this model. Regarding data normalization, we compared and studied the min-max normalization [2], z-score normalization [1], and quantile transformation. **Table 1** shows the optimization results of all methods on the scientific research enterprise dataset. Without feature optimization, the accuracy reached 79.7% after training the dataset using SCN, which is 3.7 percentage points lower than the training accuracy, indicating overfitting. By introducing feature optimization methods, the prediction accuracy improved by 7.1 to 13.2 percentage points. The scaling ratio has little effect on the final prediction result. When using non-linear mapping quantile transform, it is less affected by outliers and has better results in training time and training accuracy. Unlike linear mapping, non-linear mapping data can fit to an optimal solution more quickly, reducing the overall training time by one-third.

Table 1: Results of different feature optimization methods based on SCN

Method	Train Acc (%)	Eval Acc (%)	Total Time (s)
None	83.43	79.7	73.8
Normalization [0,1]	88.8	86.8	86.66

Normalization [-1,1]	88.6	87.35	86.63
Z-Score Normalization	88.85	86.2	84.06
Equidistribution Mapping	**93.6**	**92.9**	**55.04**

Moreover, we used PCA for optimizing feature aggregation. We aimed to assess the effect of manually collected data dimensions on the prediction outcomes of our models. The evaluation results are detailed in **Table 2**. When using PCA alone to minimize the dimension of features, there is no discernible impact on the accuracy of the model. However, excessive dimension reduction carries the risk of losing key information and deteriorating the performance of our model. Subsequently, the study demonstrated that that applying PCA to optimize dimensions after already improving the features in terms of magnitude can elevate prediction accuracy for the four normalization methods by 1-2 percentage points. These results endorse the dependability of extracting and aggregating the principal components of the feature set. In addition, ignoring feature coordinates with near-zero variance as a feature optimization tactic is deemed sound.

Table 2: Results of introducing PCA.

Optimized Method						**Train Acc**	**Eval Acc**
A	B	C	D	E	F	%	%
						81.43	78.7
				√		80.4	77.9
					√	80.9	78.9
√						88.8	85.8
	√					86.8	86.35
		√				87.85	86.1
			√			93.4	92.9
√				√		89.5	86.7
	√			√		88.3	87.3
		√		√		89.93	87.75
			√	√		93.4	92.75
√					√	90	87.25
	√				√	89.89	87.8
		√			√	90.59	88.36
			√		√	**94.97**	**93.95**

3.2 Experiments and Analysis

As enterprise data is characterized by rapid growth and change over time, selecting a backbone model requires careful consideration of robust generalization performance while striking a balance between generalization and accuracy. In this section, we compared and evaluated our model, which was implemented based on the feature optimization method discussed in section 3.1, with SVM, TextCNN and MLP. The evaluation dataset was divided into three categories, including scientific research enterprises (10,000), information transmission enterprises (10,000), and a combination of scientific research and information transmission enterprises (20,000). The evaluation criteria were training accuracy and prediction accuracy.

3.2.1 Original Model Effect

To assess the performance of the model that combines feature optimization with SCN, we trained each subsequent model in this section on the combination dataset without any optimization technique. **Table 3** shows that we used this dataset as a reference point for subsequent model optimization and comparison. The specific model hyperparameters for SVM, TextCNN, and MLP will be described in subsequent chapters.

Table 3: Results of different native models.

Method	Train Acc (%)	Eval Acc (%)	Total Time (s)
SVM	88	74	428.7
TextCNN	91	84	185
MLP	74	73.3	4.8
SCN	79.6	77.6	72

3.2.2 SVM vs OURS

SVM is a commonly used supervised classification algorithm in machine learning. In this experiment, we set the following hyperparameters for the SVM: a penalty coefficient of 100, L2 loss as the loss function, the RBF Gaussian Kernel function as the kernel function, and a kernel function coefficient of 0.001. Based on the evaluation results presented in **Table 4**, the accuracy of the model proposed in this paper outperformed SVM in all three datasets. Although SVM had similar generalization performance to our proposed model structure, introducing PCA dimensionality reduction did not enhance the effectiveness of the SVM algorithm.

Table 4: Comparison results with SVM under datasets.

DataSet	Method		Backbone	Train Acc (%)	Eval Acc (%)
Scientific Research（10000）	D	×	SVM	93.4	92.8
			SCN	**93.6**	**92.9**
		E	G	93.14	**93**
			J	**93.4**	92.75
		F	G	93.45	93.38
			J	**94.97**	**93.95**
Scientific research + Information transmission（20000）	D	×	G	91.1	**90**
			J	**92.1**	89.65
		E	G	90.7	89.7
			J	**91.89**	**90**
		F	G	91.1	90.04
			J	**92.55**	**90.45**

3.2.3 TextCNN vs OURS

The TextCNN employs the CONV1D layer for effective feature extraction from input samples of dimension 1. This paper introduces TextCNN as an alternative network solution for evaluation purposes. Regarding the experiments, the network hyperparameters were set as

follows: cross-entropy loss as the loss function, Adam algorithm as the optimizer, the activation function is RELU, and the training epochs being set to 40.

Table 5: Comparison results with TextCNN under datasets.

DataSet	Method		Backbone	Train Acc (%)	Eval Acc (%)
Scientific Research（10000）	D	×	TextCNN	93.29	92.45
			SCN	**93.6**	**92.9**
		E	H	92.56	92.57
			J	**93.4**	**92.75**
		F	H	93.2	93
			J	**94.97**	**93.95**
Scientific research + Information transmission（20000）	D	×	H	92.27	**92.33**
			J	**92.36**	92.15
		E	H	91.62	91.72
			J	**92.32**	**91.79**
		F	H	92.27	92.41
			J	**92.94**	**92.7**

According to the results in **Table 3**, TextCNN is a well-established deep neural network, exhibiting performance superiority by a margin of 10 percentage points compared to other models. Nonetheless, it has been observed to suffer from overfitting issues on enterprise datasets. **Table 5** compares the TextCNN's performance after feature optimization, revealing that TextCNN is relatively insensitive to such efforts. It can be inferred that, owing to the high dimensionality of deep models, small-size training data cannot suffice to constrain their numerous parameters, and overly elongating the training process may not necessarily enhance the performance, but rather exacerbate the overfitting issue. By contrast, SCN network features a simpler architecture and can adapt to enterprise datasets more quickly, and yields superior results to TextCNN.

3.2.4 MLP vs OURS

Multilayer Perceptron (MLP) is often used as the final classification layer in deep learning due to its simple structure and small parameter count. In this experiment, the hyperparameters of MLP are set as follows: the lbfgs optimizer of the quasi-Newton method is used as the optimizer, the sigmoid function is used as the activation function, and the number of nodes in the hidden layer is set to 128.

Table 6: Comparison results with MLP under datasets.

DataSet	Method		Backbone	Train Acc (%)	Eval Acc (%)
Scientific Research（10000）	D	×	MLP	95.5	91.8
			SCN	**93.6**	**92.9**
		E	I	93.2	92.2
			J	**93.4**	**92.75**
		F	I	93.4	92.8
			J	**94.97**	**93.95**

Scientific research + Information transmission (20000)	D	×	I	92.2	91.5
			J	**92.36**	**92.15**
		E	I	91.5	91.35
			J	**92.32**	**91.79**
		F	I	92.29	92
			J	**92.94**	**92.7**

According to **Table 3**, MLP achieved a classification accuracy of 73.3% in only 5 seconds. While its accuracy may not be as high as other models, MLP still holds potential. In this section, we introduce feature optimization to evaluate the overall effect of MLP. The results are shown in **Table 6**, indicate that MLP has good generalization performance on the three datasets. However, the inability to fit a large volume of data with parameters limits its overall accuracy compared to the SCN model. Given that enterprise data will continue to change and expand in the future, the MLP structure cannot sustain a robust classification performance when the dataset expands significantly.

4 CONCLUSIONS

This paper proposes a method to analyze enterprise investment behavior by using feature optimization and joint SCN on a dataset, allowing investment promotion personnel to make objective and informed judgments when evaluating investment targets. This approach combines the fields of artificial intelligence and industry research, using big data processing methods and machine learning algorithms to efficiently analyze massive amounts of enterprise data. The goal is to assist business personnel in making efficient and data-driven investment decisions.

In comparative experiments, the method realizes the balance between generalization and accuracy, especially the optimization of overall training time. At the same time, the SCN model combines the fast-fitting advantages of the MLP network with incremental learning to address the former shortcomings in prediction accuracy. Adoption of the feature optimization methods for the enterprise feature set has significantly improved most models, particularly the SCN model, which saw a 15.2 percentage point increase in accuracy and a 30% acceleration in fitting time. Finally, the joint SCN model, along with the feature optimization, has resulted in an overall prediction accuracy of 93% for investment behavior in enterprises.

Under the same conditions of data usage, the method can be applied to numerous application scenarios within the context of big data. In the future expansion and improvement of enterprise datasets are expected to enhance the final model performance further.

REFERENCES

[1] Fei N, Gao Y, Lu Z, et al. Z-score normalization, hubness, and few-shot learning[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 142-151. https://doi.org/10.1109/ICCV48922.2021.00021

[2] Gökhan A, Güzeller C O, Eser M T. The effect of the normalization method used in different sample sizes on the success of artificial neural network model[J]. International Journal of Assessment Tools in Education, 2019, 6(2): 170-192. https://doi.org/10.21449/ijate.479404

[3] Howell D C. Median absolute deviation[J]. Encyclopedia of statistics in behavioral science, 2005. https://doi.org/10.1002/0470013192.bsa384
[4] Rahm E, Do H H. Data cleaning: Problems and current approaches[J]. IEEE Data Eng. Bull., 2000, 23(4): 3-13. https://cs.brown.edu/courses/cs227/archives/2017/papers/data-cleaning-IEEE.pdf
[5] Sun C. Research on investment decision-making model from the perspective of “Internet of Things+ Big data”[J]. Future generation computer systems, 2020, 107: 286-292. https://doi.org/10.1016/j.future.2020.02.003
[6] Shlens J. A tutorial on principal component analysis[J]. arXiv preprint arXiv:1404.1100, 2014.https://doi.org/10.48550/arXiv.1404.1100
[7] Wang D, Li M. Stochastic configuration networks: Fundamentals and algorithms[J]. IEEE transactions on cybernetics, 2017, 47(10): 3466-3479. https://doi.org/10.1109/TCYB.2017.2734043
[9] Wu J M T, Li Z, Herencsar N, et al. A graph-based CNN-LSTM stock price prediction algorithm with leading indicators[J]. Multimedia Systems, 2021: 1-20. https://doi.org/10.1007/s00530-021-00758-w
[10] Yang X. The prediction of gold price using ARIMA model[C]//2nd International Conference on Social Science, Public Health and Education (SSPHE 2018). Atlantis Press, 2019: 273-276. https://doi.org/10.2991/ssphe-18.2019.66
[11] Yu K, Lu Z, Stander J. Quantile regression: applications and current research areas[J]. Journal of the Royal Statistical Society: Series D (The Statistician), 2003, 52(3): 331-350. https://doi.org/10.1111/1467-9884.00363