J. COMBIN. MATH. COMBIN. COMPUT. 127a (2025) 9083--9097

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Neural Network-based Market Fluctuation Prediction Model Construction and Corporate Strategy Research in Business Administration

Xin Yuan[1], ✉

[1] *School of Society and Humanities, Jiangxi University of Finance and Economics, Nanchang, Jiangxi, 330013, China*

ABSTRACT
Based on the complexity and nonlinear characteristics of market volatility, this paper proposes a market volatility prediction model that combines MA filtering method, autoregressive moving average (ARMA), and long-short-term memory (LSTM) neural network. And the back-propagation (BP) neural network is utilized to quantitatively solve the problem of corporate strategy formulation, and a corporate strategy formation model is established to determine the corporate strategic choice through the corporate strategic environment and strategic capabilities. In the ablation experiment, the combined model MA-ARMA-LSTM reduces its MSE, RMSE, MAE and MAPE by 0.0007, 0.0131, 0.0074 and 1.57%, respectively, compared to the ARMA model. Compared with common market volatility prediction models, the combined model has the smallest error in each assessment index. The output of BP neural network for corporate strategy selection is consistent with the expert ranking, which is verified to be in line with the actual business situation, indicating that the method in this paper can provide a reasonable corporate strategy.

Keywords: MA filtering method, ARMA, LSTM, BP neural network, market volatility prediction

1. Introduction

Driven by the wave of economic globalization in the 21st century and strongly supported by communication technology, trade and investment among different countries and regions have increased, and cross-border capital flows have accelerated the interconnection of financial markets of different countries, and the correlation among international financial markets has been gradually enhanced, which at the same time has also brought a series of risks.

China has injected new vitality into the manufacturing and other real industries and promoted the development of China's real economy. As the core assets in the financial derivatives market, stock index

✉ Corresponding author.
E-mail address: yianxin99666@163.com (X. Yuan).
Received 15 January 2024; Revised 10 February 2024; Accepted 30 November 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-505

assets are the key assets for investors to carry out asset allocation and risk management, and they are also important risk avoidance tools that cannot be ignored in the financial system [1-4]. China's foreign investment in international stock index belongs to an important part of the international cycle of "going out", in actively guiding the development of foreign countries should also pay attention to the existence of spatial correlation characteristics between the stock index assets of various countries, the need to guard against the risk of fluctuations in the international financial environment [5-8]. Volatility is the core of financial risk measurement and prediction, is an important indicator to measure the volatility of financial asset prices, reflecting the uncertainty and risk level of asset prices, and its changes will be closely related to market sentiment, macroeconomic data and other factors [9-11]. The higher volatility means that the asset price fluctuation is larger and the risk level is higher, and conversely the lower volatility represents the asset price fluctuation is small and the asset price is relatively stable [12-14]. Therefore, the study of volatility is a crucial part in financial risk prediction and research. Finding appropriate models and methods to measure and predict the volatility of stock index assets is the core of volatility research.

Neural network machine learning algorithms, as a major core technology in the field of artificial intelligence, with good performance and flexible structure, have been developing rapidly in natural language processing, image and speech recognition, sequence processing, etc., and have provided a new method for forecasting financial time series. Bucci, A. introduced the advantages of artificial neural network forecasting methods, namely, that it is possible to forecast linear or nonlinear without knowing the data generation process of the data for approximate forecasting, and by comparing neural networks with traditional econometric methods, it was found that the long-range dependence captured by neural network models can improve forecasting accuracy during periods of high market volatility [15]. Ge, W. et al. showed that financial market volatility prediction is closely related to the decision making of market participants, proposed a shared task to evaluate the performance of neural network-based financial volatility prediction models, and based on this, proposed an improvement method [16]. Kristjanpoller, W. et al. proposed the use of hybrid artificial neural network-generalized autoregressive conditional heteroskedasticity (ANN-GARCH) models to forecast price fluctuations of virtual cryptocurrencies, which enables firms to better manage exchange rate risk and leveraged assets as part of their overall financial strategy [17]. Chen, W. J. et al. pointed out that market volatility can reflect the basic information of investors or policy makers to a certain extent and designed a hybrid deep neural network model (HDNN) based on temporal embedding, which utilizes an end-to-end learning paradigm to forecast volatility of a dataset from a time series, which improves predictive ability compared to statistical methods [18]. Wang, Y. et al. emphasized that in addition to historical trends, social factors are also important factors affecting price fluctuations in the stock market, and proposed a hybrid time-series prediction neural network (HTPNN) incorporating news effects, which significantly improves the predictive performance of the model by learning the fusion features of news word vectors and time series to capture the underlying patterns of fluctuations [19]. Liu, Y. compared the performance of Support Vector Machine (SVM), Long and Short Term Memory Recurrent Neural Networks (LSTM RNNs) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models in financial volatility or risk prediction, and experiments showed that the LSTM RNNs are better in coping with the prediction of long series of data, which can help enterprises optimize the investment portfolio to achieve the maximization of profits [20]. Xu, K. et al. examined the application of graph neural networks in financial risk management, which can effectively capture the interaction between nodes in a complex system, so that the prediction accuracy can be improved by learning the interactions between different assets when utilizing graph neural networks for the prediction of volatility and value-at-risk calculations in the financial market [21]. Chen, Q. et al. introduced a graph Transformer network to multivariate the limit order data as well as the related stock market network to predict the short-term volatility of the stock market by learning the temporal and cross-sectional relationships from different sources in the data, and the experimental results showed that the proposed model has better performance performance [22]. Liu, Y. et al. constructed an online stock forum sentiment analysis model using a recurrent neural network to fuse the two collected sentiment indicators into the market data for volatility prediction, and found that the prediction performance of the model was significantly improved after mixing [23].

Research on market volatility forecasting based on ARMA model. Considering the time series component characteristics, MA filtering method that can decompose the time series into high volatility components and low volatility components is added. Considering the nonlinear characteristics of market volatility, LSTN deep learning model is added. Finally, the MA-ARMA-LSTM model is constructed for market volatility prediction. BP neural network is selected to quantitatively solve the problem of corporate strategy formulation. The fuzzy evaluation method is utilized to solve the metric problem of linguistic description variables, and the model of enterprise strategic environment and capability determining strategic choice is established.

2. Neural network-based market volatility prediction modeling

Market volatility forecasting is an important topic in finance. Traditional market volatility forecasting methods, such as linear autoregressive moving average (ARMA), although effective in some cases, fail to capture the nonlinear characteristics of market volatility [24]. Therefore, this paper proposes the application of ARMA and nonlinear LSTN deep learning models in volatility time series data forecasting. Numerous combinations of ARMA models and neural networks have been developed by first applying an ARMA model on a given time series data, considering the error between the original data and the ARMA forecasted data as a nonlinear component, and modeling it in a different way with artificial neural networks. Although these models give a higher prediction accuracy than the individual models, there is scope to further improve the accuracy if the nature of the given time series is taken into account before applying the models. Therefore, in this paper, the proposed MA filtering method is introduced to preprocess the data before prediction.

2.1. MA filtering method

The basic idea of MA filtering method is that a smooth time series, if it obeys the normal distribution, then the series is strictly smooth, so if an ARMA time series is strictly smooth, then it may obey the normal distribution, to test whether a series $\{y_t\}$ is normally distributed, usually judged from its kurtosis value, the kurtosis value calculation formula:

$$kurtosis = \frac{E\left\{\left(y_t - E\{y_t\}\right)^4\right\}}{\left(E\left\{\left(y_t - E\{y_t\}\right)^2\right\}\right)^2} \tag{1}$$

If the kurtosis value is 3, the sequence follows a normal distribution, i.e., the sequence is low volatility. If the kurtosis value is too large or too small, the sequence is either low or high kurtosis, i.e., the sequence is high volatility.

The first thing that the MA filter separates from the original estimated time series $\{y_t\}$ is the low volatility component, and let this component be denoted by y_{tr}, then there is:

$$y_{tr} = \frac{1}{m}\sum_{i=t-m+1}^{t} y_i \tag{2}$$

The high volatility component is denoted by y_{res}. There:

$$y_{res} = y_t - y_{tr} \tag{3}$$

It is necessary to find a suitable m value to decompose the data, according to the idea of MA filtering method, the m value should be taken as 3. The MA filter decomposition diagram is shown in Fig. 1.

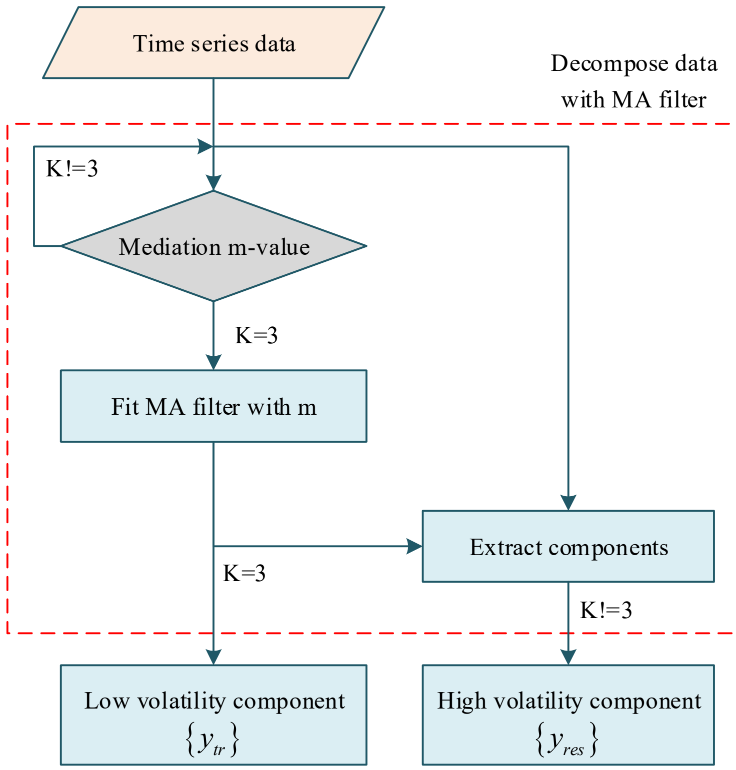

Fig. 1. Ma filtering method decomposition process

2.2. ARMA time series modeling

The autoregressive sliding average model (ARMA) is an important approach to the study of time series, consisting of a "hybrid" of an autoregressive model (AR model) and a sliding average model (MA) as a base.

A model with the following structure is called an autoregressive model of order p, abbreviated as $AR(p)$:

$$\begin{cases} x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + \varepsilon_t \\ \phi_p \neq 0 \\ E(\varepsilon_t) = 0, Var(\varepsilon_t) = \sigma_s^2, E(\varepsilon_t \varepsilon_s) = 0, s \neq t \\ E(\varepsilon_t \varepsilon_s) = 0, \forall s < t \end{cases} \tag{4}$$

In particular, when $\phi_0 = 0$, it is called centered $AR(p)$.

A model with the following structure is called an autoregressive model of order q, abbreviated as $MA(q)$:

$$\begin{cases} x_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \\ \theta_q \neq 0 \\ E(\varepsilon_t) = 0, \mathrm{Var}(\varepsilon_t) = \sigma_s^2, E(\varepsilon_t \varepsilon_s) = 0, s \neq t \end{cases} \tag{5}$$

In particular, when $\mu = 0$, it is called a centered $MA(q)$

Therefore, a model with the following structure is called an autoregressive moving average model, abbreviated as $ARMA(p,q)$:

$$\begin{cases} x_t = \phi_0 + \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \\ \phi_p \neq 0, \theta_q \neq 0 \\ E(\varepsilon_t) = 0, Var(\varepsilon_t) = \sigma_s^2, E(\varepsilon_t \varepsilon_s) = 0, s \neq t \\ E(\varepsilon_t \varepsilon_s) = 0, \forall s < t \end{cases} \tag{6}$$

In particular, when $\phi_0 = 0$, it is called a centered $ARMA(p,q)$ model.

ARMA modeling is a linear technique for dealing with time series. When using this method to deal with time series, it is first necessary to test the smoothness of the given time series, if the time series is not smooth, the time series has to be differenced to make it smooth, if the series is not smooth after one differencing, the differencing process is continued until the series is smooth [25]. If the series is smoothed after d differentiations, then the model is denoted as $ARIMA(p,d,q)$. If the series itself is smoothed, there is no need for differentiation, then $d = 0$, then the model is denoted as $ARMA(p,q)$.

Generally it takes three steps to build an ARMA model:

1) Identify the model order, that is, to find out the value of p and q, calculate the value of p and q with the correlation analysis, that is, the use of serial autocorrelation function and partial autocorrelation function of the nature of the series to do the autocorrelation plot and partial autocorrelation plot, in turn, to determine the value of p and q.

2) Estimate the model parameters, estimating the model parameters generally use Box-Jenkins method or Gaussian Maximum Likelihood Estimation (GMLE) method.

3) Use the estimated model to predict the data, the model validity needs to be tested before prediction, the validity of the model is tested using the AIC criterion, the best ARMA model has the smallest AIC value.

2.3. LSTM deep neural network

Long Short-Term Memory (LSTM) networks are a more accurate network structure incorporating gradient learning algorithms.

A step-by-step approach to understanding LSTM:

Step 1: Decide what information we will discard from the cell state. This decision is made through a layer called the "forget gate", which reads the volatility prediction for the previous moment and the inputs for that moment, and outputs a value between 0 and 1 for each number in cell state c_{i-1}. 1 means "fully retained" and 0 means "fully discarded":

$$f_i = sigmoid\left[\left(h_v, H_i\right) \cdot W_f + b_f\right] \tag{7}$$

Step 2: Determine what new information is stored in the cell state. There are two parts to this, first, an $sigmoid$ layer called the "Input Gate Layer" decides what values we are going to update, and then a tanh layer creates a new vector of candidate values that are added to the state:

$$c_i = sigmoid\left[\left(h_l, H_i\right) \cdot W_c + b_c\right] \tag{8}$$

$$\tilde{I}_l = \tanh\left[\left(h_i, H_i\right) \cdot W_{\tilde{I}} + b_{\tilde{I}}\right] \tag{9}$$

Step 3: Determine the new information to update. Now it's time to update the old cell state, c_{i-1} to c_i. The previous steps have determined what will be done, now it's just a matter of actually getting it done, multiplying the old state I_{i-1} with f_i and discarding the information we've determined needs to be discarded. Then add $c_i \cdot \tilde{I}_t$, which is the new candidate value, after which we decide how much to update each state for the change:

$$I_i = I_{i-1} \cdot f_i + c_i \cdot \tilde{I}_t \tag{10}$$

Step 4: Determine what value to output. This output will be based on the cell state, but also a filtered version. First run an $sigmoid$ -layer to determine what part of the cell state will be output, then run the cell state through tanh (to get a value between -1 and 1) and multiply it by the output of the $sigmoid$ -gate, which will end up just outputting the portion we determined to output:

$$\hat{h}_{i+1} = \alpha + \beta \cdot o_i \tanh\left[I_i\right] \tag{11}$$

$$o_i = sigmoid\left[\left(h_l, H_i\right) \cdot W_0 + b_0\right] \tag{12}$$

2.4. Combined ARMA-LSTM model based on MA filtering method

In practice, time series are usually characterized by composite and high volatility components, which may not be well portrayed by a single ARMA model or RNN model. After obtaining the stock trading data, we first calculate the volatility time series, which is denoted as $\{y_t\}$, and then decompose $\{y_t\}$ into a low volatility series $\{l_t\}$ and a high volatility series $\{h_t\}$ based on the MA filtering method mentioned above. Since the ARMA model is a typical linear model, which can portray the characteristics of linear series, i.e., the low volatility series, well, we use the ARMA model to do the prediction for $\{l_t\}$. And the LSTM model happens to be able to portray the nonlinear series relatively well, so we use the LSTM deep neural network model to do the prediction of $\{h_t\}$, and construct the combined model MA-ARMA-LSTM.

The steps to build the MA-ARMA-LSTM combined model are as follows:

1) Decompose the volatility time series $\{y_t\}$ into two parts using MA filtering:

$$y_t = l_t + h_t \tag{13}$$

where l_t denotes the low volatility component and h_t denotes the high volatility component.

2) Test the stability of the low volatility series $\{l_t\}$ and build an ARMA model for forecasting:

$$l_t = f\left(l_{t-1}, l_{t-2}, \cdots, l_{t-p}, \varepsilon_t, \varepsilon_{t-1}, \cdots, \varepsilon_{t-q}\right) \tag{14}$$

where $\varepsilon_t, \varepsilon_{t-1}, \cdots, \varepsilon_{t-q}$ is a sequence of random errors and f is a linear ARMA equation, viz:

$$\begin{cases} x_t = \phi_1 l_{t-1} + \cdots + \phi_p l_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \\ \phi_p \neq 0, \theta_q \neq 0 \\ E\left(\varepsilon_t\right) = 0, Var\left(\varepsilon_t\right) = \sigma_s^2, E\left(\varepsilon_t \varepsilon_s\right) = 0, s \neq t \\ E\left(\varepsilon_t \varepsilon_s\right) = 0, \forall s < t \end{cases} \tag{15}$$

3) High volatility sequence $\{h_t\}$ with 3-step forward forecasting using information from $h_{t-1}, h_{t-2}, \cdots, h_{t-N}$:

$$h_t = g\left(h_{t-1}, h_{t-2}, \cdots, h_{t-N}\right) + \varepsilon_t \tag{16}$$

where $g(\cdot)$ is the LSTM neural network.

4) Calculate the final predicted value of the time series $\{y_t\}$ predicted in the first 3 steps:

$$y_t = \hat{l}_t + h_t \tag{17}$$

where $\hat{l}_t = \hat{f}\left(l_{t-1}, l_{t-2}, \cdots, l_{t-p}, \varepsilon_t, \varepsilon_{t-1}, \cdots, \varepsilon_{t-q}\right)$, $h_t = \hat{g}\left(h_{t-1}, h_{t-2}, \cdots, h_{t-N}\right)$.

3. Corporate strategy formation based on BP neural network

3.1. Corporate strategy formation model

In the presence of a complete set of strategy pools, strategy formation is actually the selection of appropriate strategic issues in the set of corporate strategy pools based on the corporate strategic environment and corporate strategic capabilities. Therefore, the corporate strategy formation model can be abstractly represented as a mapping from the space of corporate strategic environment and the space of corporate strategic capabilities to the space of corporate strategies, denoted as:

$$f: E \times C \to S \tag{18}$$

Here S is the set of strategies, E is the set of strategic environments, and C is the set of strategic capabilities. To find this mapping relationship, it is first necessary to identify the elements and measures of each space.

The strategic pool set consists of the strategic measures and their possible combinations in the strategic space. In this paper, it is divided into two dimensions: the development dimension and the operational dimension. The development dimension component consists of 9 types (vertical integration, concentric diversification, horizontal integration, integration, mergers and acquisitions, profitability priority, prudent stabilization, austerity shifts, and abandonment strategies), and the operation dimension consists of 53 types in 13 categories (competition, investment, production, organizational structure, quality, sales promotion, research and development, human resources, and corporate culture-based strategies, etc.). The synthesis of the two-dimensional components can generate numerous combinations of corporate strategies. Assuming that there are N two-dimensional components, $S = \{s_1, s_2, \cdots, s_N\}$, thus constituting a solution set for corporate strategy selection. Obviously corporate strategy choices can be conveniently represented by Boolean variables. Each component of a corporate strategy (n in total) is binary coded so that when the component is selected, its corresponding Boolean variable takes 1, otherwise 0. This allows for the representation of not only a single strategy, but also a composite strategy.

The strategic environment of a firm consists of the general environment (political, economic, technological, socio-cultural, natural and international environments) and the competitive environment (industry trends, suppliers, buyers, industry competition, potential competition, substitutes, and other stakeholder group factors) for a total of 86, assuming a total of M factor, $E = (e_1, e_2, \cdots, e_M)$. Similarly, the strategic capabilities of a firm can be categorized into intangible strategic capabilities (cultural perceptions, social status, etc.) and tangible strategic capabilities (e.g., number of capital turnovers, market share, etc.) for a total of 29 factors. Similarly, corporate strategic capabilities can be categorized into intangible strategic capabilities (cultural perceptions, social status, etc.) and tangible strategic capabilities (e.g., the number of capital turnovers, market share, technological development capabilities, etc.), which are generally recorded as $C = (c_1, c_2, \cdots, c_l)$. The factors of strategic environment and strategic capability are obviously both quantitative and non-quantitative indicators. Quantitative indicators do not have the difficulty of measurement, while a large proportion of non-quantitative indicators are usually described by language through relative comparison. For example, political laws can be described as good, better or very poor. Therefore, for such indicators it is necessary to use fuzzy evaluation methods to determine their metrics. Corporate strategic environment and capabilities are equivalent to inputs in the mapping relationship.

3.2. Input Variables and Metrics

3.2.1. Fuzzy integrated evaluation of linguistic description indicators. Linguistic description indicators can be measured by the results of fuzzy relationship operations defined on fuzzy sets, and the process is actually a fuzzy comprehensive evaluation process. Let $X = (x_1, x_2, \cdots, x_1)$ be the set of linguistic description indicators to be evaluated, called the factor set. $A = (a_1, a_2, \cdots, a_J)$ is the fuzzy evaluation set of the indicators, in this paper $A=$ (very good, better, average, very poor). The operation is noted as $\vec{r}_i=[1,0.7,0.5,0.1,0.0]^T$, and the superscript T denotes the transpose. $U=(u_1,u_2,\cdots u_J)$ is the scale set, u_J indicates the scale corresponding to the evaluation of the j th level, through the scale set can be molded into the subordination vector of the variables can be integrated into a scalar, so that it can be processed together with non-fuzzy variables, this paper adopts a linear scale, $\vec{u}=[0.9,0.7.0.5,0.3,0.1]^T$.

Evaluating each of the linguistic descriptors of the strategy and strategic capabilities of an enterprise (without loss of generality, denoted as k enterprise), I vectors $\vec{r}_i^{\,k}=\left[r_{i1}^k,r_{i2}^k,\cdots,r_i^k\right]^T$ can be obtained, which can form the evaluation matrix of linguistic descriptors of the strategic environment and strategic capabilities of the enterprise:

$$R_k=\left[\vec{r}_1^{\,k},\vec{r}_2^{\,k},\cdots,\vec{r}_I^{\,k}\right]^T \tag{19}$$

Using the scale vector $\vec{u}$ the evaluable matrix is converted to a fuzzy evaluation value vector:

$$\vec{V}_k=R_k\vec{u}=\left[v_1^k,v_2^k,\cdots,v_I^k\right]^T \tag{20}$$

Here v_i^k denotes the model evaluation value of the i rd linguistic description indicator of k enterprises. Through the above fuzzy operation, the fuzzy representation of linguistic description indicators is converted into fuzzy evaluation value (variables between 0 and 1), which can be handled in the same way as quantitative indicators.

3.2.2. Standardization of quantitative indicators. There is a small variety of quantitative indicators in the strategic environment and capabilities of the enterprise, with different units, and all quantitative indicators are standardized using the following transformations in order to facilitate the processing and reduce the impact on the next step of the neurological plan:

$$f(x)=\frac{x-x_{\min}}{x_{\max}-x_{\min}} \tag{21}$$

After this transformation, x becomes a standardized quantity that is dimensionless and lies between 0 and 1.

3.2.3. Synthesis of enterprise strategic environment and capability indicators. Since the numerous indicators of the strategic environment and capabilities of enterprises bring many difficulties for further processing and need to be dealt with in a hierarchical manner, they can be merged into several major categories of indicators as needed, after the above processing. Both the linguistic description indicators and the mathematical and physical indicators have been measured as scalars between 0 and 1, so they can be easily merged into several major indicators using the weighting method. Let $\vec{W}=\left[w_1,w_2\cdots,w_p\right]^T$ be the weight vector $0<w_i<1$, $\sum W_i=1$. w_i reflects the importance given to the i reflects the importance given to the i indicators in the merging. The merged indicator $Y=\vec{W}^T\cdot\vec{X}$, here $\vec{X}$ is the vector composed of the indicators to be merged, and the weights can be obtained by expert experience or other methods, here the hierarchical analysis method is used to obtain them.

3.3. Neural network modeling of strategic choices

Backpropagation model (BP) and its training algorithm are the more mature and mathematically based networks and algorithms in current neural network research. The previous research on the formation of corporate strategy has actually transformed the problem into a problem of the influence of strategic decision forces on strategic choices similar to the classification problem, so the BP network and training algorithm are selected [26].

BP network is composed of an input layer, an output layer and several hidden layers. A network with one hidden layer can form an arbitrary convex domain in the space tensored by the input variables, and two hidden layers can form an arbitrarily complex judgment region. Here a BP network with one hidden layer is used and the network is trained using the standard BP algorithm. Where the losers of the BP network are the strategic decision forces and the outputs are the strategic components, the number of hidden nodes has not yet been decided in a general way and needs to be determined empirically using the trial and error method.

4. Empirical analysis

4.1. Analysis of experimental results of market volatility forecasting models

4.1.1. Introduction to the data set. In this paper, we use the daily trading data of 3 markets, SSE 50 index, SSE 50 ETF fund and SSE 50 ETF option, in the WIND database, with the time span from February 9, 2017 to December 22, 2023 period. The prediction target is the volatility of the SSE 50 ETF fund, and 11 financial characteristics data are selected as the experimental indicators, including option implied volatility, fund AD, stock closing price, option maximum price, fund volume-weighted average price, fund closing price, option closing price, option degree of in-value, fund average price, fund maximum price, and option risk-free rate.

4.1.2. Evaluation indicators. The volatility prediction task of the SSE 50 ETF fund studied in this paper is essentially a regression task. Therefore, four common evaluation metrics in regression tasks are used, i.e., mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), as shown in Eqs. (22) to (25):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i^{pred}\right)^2 \tag{22}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i^{pred}\right)^2} \tag{23}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i^{pred} - y_i\right| \tag{24}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i^{pred} - y_i}{y}\right| \tag{25}$$

4.1.3. Experimental results. In order to verify the effectiveness of the model proposed in this paper (MA-ARMA-LSTM) for the volatility prediction task of the SSE 50 ETF fund, comparative analyses are carried out through model ablation experiments and comprehensive experiments, so as to better assess the performance of MA-ARMA-LSTM in the volatility prediction task.

1) Comparison of sliding window number. The experiments in this section use a sliding window for data prediction, as an example of the volatility prediction task for the SEC 50 ETF fund.Sliding window is a common method in time series forecasting tasks, which is usually used to predict future trends using past data. Therefore, the window size has an important impact on the accuracy of predicting high-value information. If the window is too large, the model will be more inclined to use earlier data and ignore more recent information, and if the window is too small, it will not be able to fully utilize the earlier information. In order to better utilize the information at each time point, this subsection tries to find the best matching model using different sizes of windows by conducting experiments on the MA-ARMA-LSTM model. The experimental results are shown in Table 1. The experimental results

show that using a sliding window of 15 gives lower prediction error metrics and is more effective than other sliding window values in the time series forecasting task. This indicates that when the window size is 15, the model can better utilize the historical data for forecasting and improve the robustness and stability of the model, which leads to more accurate forecasting results. Therefore, the MA-ARMA-LSTM model will choose the sliding window value of T=15 in the next experiments, which provides a reference for the subsequent experiments.

Table 1. Results of Sliding window experiment

T	MSE	RMSE	MAE	MAPE/%
5	0.0008	0.0309	0.0345	23.45%
10	0.0007	0.0277	0.0216	22.58%
15	0.0005	0.0217	0.0167	20.88%
20	0.0006	0.0321	0.0221	22.07%
25	0.0009	0.0254	0.0214	21.45%
30	0.0006	0.0284	0.0247	23.14%

2) Ablation experiment. In order to verify the impact of the addition of MA and LSTM on the performance of the ARMA model, ablation experiments are conducted in this paper with the aim of assessing the extent and importance of the different modules on the model prediction. By gradually removing or replacing these modules, analyzing their contribution in the model performance and further understanding the key role of these modules on the overall model performance, it helps to gain a deeper understanding of the structure and design of the model to optimize and improve the performance of volatility prediction tasks. The experimental results are shown in Table 2. The ARMA model with the addition of both MA and LSTM moderately reduces the error metrics of the model. The MA-ARMA-LSTM model that combines them performs the best and has the smallest error among all the models, with MSE, RMSE, MAE and MAPE reduced by 0.0007, 0.0131, 0.0074 and 1.57%, respectively, compared to the ARMA model.

Table 2. Results of ablation experiment

Model	MSE	RMSE	MAE	MAPE/%
ARMA	0.0013	0.0368	0.0246	21.93
MA+ARMA	0.0009	0.0299	0.0234	21.65
ARMA+LSTM	0.0011	0.0281	0.0229	21.45
MA+ARMA+LSTM	0.0006	0.0237	0.0172	20.36

3) Comprehensive experiments. In order to verify the effectiveness of the model proposed in this paper for the volatility prediction task of the SSE 50 ETF fund, common market volatility prediction models are selected: TCN (Temporal Convolutional Neural Network), SVR (Support Vector Regression Model), RNN (Recurrent Neural Network Model), LSTM-attention, LSTNet, TCN, Transformer with the proposed MA-ARMA-LSTM model to achieve the task of volatility prediction of the SSE 50 ETF fund. The proposed MA-ARMA-LSTM model realizes the volatility prediction task of SSE 50 ETF fund, and the experimental results are shown in Table 3. The experimental results show that the MA-ARMA-LSTM model outperforms other models in the volatility prediction task, proving its reliability and validity in the application of financial data, and the model has the smallest error in each evaluation index, with MSE, RMSE, MAE and MAPE being 0.0003, 0.0297, 0.0117 and 20.29%, respectively, which demonstrates its superior The performance of this method is summarized as follows.

Table 3. Results of comprehensive experiment

Model	MSE	RMSE	MAE	MAPE/%
SVR	0.0049	0.0866	0.0586	37.77
RNN	0.0029	0.0561	0.0398	26.98
LSTM-attention	0.0025	0.0444	0.0213	22.77
LSTNet	0.0013	0.0421	0.0404	22.8
TCN	0.0024	0.0351	0.0287	22.74
Transformer	0.018	0.0447	0.0235	22.32
MA-ARMA-LSTM	0.0003	0.0297	0.0117	20.29

4.2. Validation of corporate strategy formation methodology

In order to verify the operability and practicability of the BP neural network method, the home appliance industry in China is selected as the background for the case study, and the research objects are all well-known enterprises in China. Due to the limitation of the source, the information is organized with 8 valid data, from which the data of 22 businesses of 7 enterprises (2022) are selected as study samples, and the data of 4 businesses of 1 enterprise (2022) are selected as test samples. The following 7 industries were selected: air conditioners (I1), washing machines (I2), refrigerators (I3), TVs (I4), cell phones (I5), microcomputers (I6), and freezers (I7). In the evaluation of multiple indicators, the scale of each indicator is not exactly the same, and it is necessary to use a dimensionless method to normalize each indicator. In this paper, we choose to use the linear normalization method to eliminate the influence of the quantitative outline in the indicators, and obtain the normalized data of each indicator of the industry demand growth rate (IDGR), the intensity of competitors in the same industry (CS), the supplier's ability to kanban (SA), the customer's ability to kanban (CA), the substitutes (S), the potential entrant (PE), the profitability (PM), the cost advantage (CA), and the degree of association (CD), and the sample The original data after normalization is shown in Table 4. Applying the BP artificial neural network method, there are 9 nodes in the input layer, corresponding to each index in the enterprise strategic choice determinants index system. The output node is 1, which is the score of enterprise business. The data of the first 7 enterprises in the table are used as training samples, and the data of enterprise 8 are used as test samples.

Table 4. The data is normalized after the sample data is normalized

Enterprise	Business	IDGR	CS	SA	CA	S	PE	PM	CA	CD
1	I1	0	0.2354	0.3366	0.3616	0.6619	0.0444	0.796	0.5204	0.6037
	I2	0.3404	0.4043	0.4118	0.4466	0.0997	0.7366	1	0.0889	0.6308
	I3	0.2281	0.4129	0.4176	0.4746	0.0673	0.9941	0.093	1	0.9928
	I5	0.9235	0	0.3105	0.367	0.2432	0.2833	0.1904	0.1072	0.2938
	I7	0.3265	0.5807	0.422	0.5571	0.0441	0	0.0296	0.3086	1
2	I1	0	0.0072	1	0.6316	0.6619	0.1005	0.7068	0.281	0.9144
	I3	0.267	0.2996	0.479	0.4633	0.1008	0.9823	0.2588	0.3896	0.9847
	I7	0.3201	0.1253	0.382	0.3919	0.0437	0	0.0379	0.4967	0.2958
3	I1	0.0255	0.1161	0.7094	0.3725	0.6601	0.1269	0.3868	0.3043	0.3962
	I3	0.2442	0	0.3132	0.4411	0.0916	1.0027	0.1139	0	0.5996
	I4	0.5744	0.169	0.3151	0.4885	0	0.4892	0.0812	0.1792	0.9878
	I6	1	0	0.2679	0.0638	0.7496	0.035	0.0011	0.168	0.2687
4	I4	0.5614	0.2113	0.954	0.3499	0.0067	0.598	0.1081	0.2109	0.5137
	I5	0.9409	0.0329	0.737	0.3441	0.2163	0.2713	0.1855	0.101	0.1636
5	I4	0.5816	0.0564	0.2893	0.299	0.4036	0.1473	0.4805	0.2986	0.5227
	I5	0.8984	0.079	0.0015	0.2198	0.3003	0	0.5509	0.4143	0.604
6	I1	0	0.2567	0.5788	0.5943	0.6665	0.1193	0.1573	0.1757	0.9391
	I3	0.231	0.0125	1	0.3807	0	0.1035	0.3862	0.292	1
	I7	0.3534	1	0.9911	0.5201	0.9699	0.2016	0.7053	1	0.809
7	I1	0.0062	0.2237	0	0.9827	1	0	0.7531	0.6047	0.2171
	I2	0.3487	0.082	0.3514	0	1	0.3992	0	0.5097	0.6037
	I3	0.2383	0.0028	0.4082	0	0.1043	1	0.2926	0.3839	1
8	I1	0	0.012	0.3704	0.1937	0.6483	0.0436	0.3023	0.3225	0.2459
	I4	0.5659	0.2229	0.6987	0.3208	0.0025	0.4218	0.1058	0.0945	0.8791
	I5	0.9337	0.1631	0.3344	0.8131	0.0978	0.4006	0.4152	0.6812	1
	I6	1	0.0288	0.3117	0.5925	0.9059	0.402	0.0412	1	0.5115

Based on the experience, 19 hidden nodes are taken in this paper and trained 100,000 times using BP algorithm. The learning results are shown in Table 5. 22 test samples have the same expected and simulated values with 0% error.

Table 5. Learning results

Serial number	Expected value	Analog value	Error
1	4	4	0%
2	1	1	0%
3	5	5	0%
4	4	4	0%
5	9	9	0%
6	7	7	0%
7	6	6	0%
8	5	5	0%
9	4	4	0%
10	4	4	0%
11	6	6	0%
12	2	2	0%
13	3	3	0%
14	7	7	0%
15	2	2	0%
16	5	5	0%
17	6	6	0%
18	6	6	0%
19	7	7	0%
20	5	5	0%
21	5	5	0%
22	1	1	0%

The training results are shown in Table 6. The maximum error between the output value obtained using the BP neural network and the expected value is 18.3%, which is related to the small number of learning samples, but does not affect the total ranking, and the business score ranking of enterprises is consistent with the expert ranking. The business score size ranking of enterprise 8 is TV (9.2621), cell phone (8.5321), air conditioner (3.2154), microcomputer (2.3651), the key development business should be cell phone, TV, the second development business is air conditioner, and the business that should be discarded is microcomputer business. From the development of the enterprise in 2023, the enterprise prioritizes the development of TV and cell phone business, air-conditioning is the second, and the enterprise's microcomputer business is operating miserably, which also verifies that the diversification strategy selection method based on BP neural network proposed in this paper is effective.

Table 6. Training results

Serial number	Expected value	Analog value	Error
23	3	3.2154	7.2%
24	9	9.2621	2.9%
25	8	8.5321	6.7%
26	2	2.3651	18.3%

5. Conclusion

Study of constructing MA-ARMA-LSTM model for market volatility prediction. Select BP neural network to quantitatively solve the problem of corporate strategy formulation.

1) Sliding window value of T=15 is selected for the ablation and comparison experiments of MA-ARMA-LSTM model.The MSE, RMSE, MAE and MAPE of MA-ARMA-LSTM model are reduced by 0.0007, 0.0131, 0.0074, and 1.57%, respectively, compared with ARMA model. Compared with common market volatility prediction models, the combined model has the optimal performance.

2) The study selects 22 sample data for corporate strategy formation model training and 4 sample data for corporate strategy formation model testing.The corporate strategy selection output from the BP neural network is consistent with the expert ranking and in line with the actual business conditions, which verifies that the corporate strategy selection method based on BP neural network proposed in this paper is effective.

References

[1] Hu, Y., Ni, J., & Wen, L. (2020). A hybrid deep learning approach by integrating LSTM-ANN networks with GARCH model for copper price volatility prediction. Physica A: Statistical Mechanics and its Applications, 557, 124907.

[2] Pradeepkumar, D., & Ravi, V. (2017). Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network. Applied Soft Computing, 58, 35-52.

[3] Ramos-Pérez, E., Alonso-González, P. J., & Núñez-Velázquez, J. J. (2021). Multi-transformer: A new neural network-based architecture for forecasting S&P volatility. Mathematics, 9(15), 1794.

[4] Vortelinos, D. I. (2017). Forecasting realized volatility: HAR against Principal Components Combining, neural networks and GARCH. Research in international business and finance, 39, 824-839.

[5] Miura, R., Pichl, L., & Kaizoji, T. (2019). Artificial neural networks for realized volatility prediction in cryptocurrency time series. In Advances in Neural Networks–ISNN 2019: 16th International Symposium on Neural Networks, ISNN 2019, Moscow, Russia, July 10–12, 2019, Proceedings, Part I 16 (pp. 165-172). Springer International Publishing.

[6] Seo, M., & Kim, G. (2020). Hybrid forecasting models based on the neural networks for the volatility of bitcoin. Applied Sciences, 10(14), 4768.

[7] Liu, F., Pantelous, A. A., & von Mettenheim, H. J. (2018). Forecasting and trading high frequency volatility on large indices. Quantitative Finance, 18(5), 737-748.

[8] Yu, S., & Li, Z. (2018). Forecasting stock price index volatility with LSTM deep neural network. In Recent Developments in Data Science and Business Analytics: Proceedings of the International Conference on Data Science and Business Analytics (ICDSBA-2017) (pp. 265-272). Springer International Publishing.

[9] Baffour, A. A., Feng, J., & Taylor, E. K. (2019). A hybrid artificial neural network-GJR modeling approach to forecasting currency exchange rate volatility. Neurocomputing, 365, 285-301.

[10] Othman, A. H. A., Kassim, S., Rosman, R. B., & Redzuan, N. H. B. (2020). Prediction accuracy improvement for Bitcoin market prices based on symmetric volatility information using artificial neural network approach. Journal of Revenue and Pricing Management, 19, 314-330.

[11] Christensen, K., Siggaard, M., & Veliyev, B. (2023). A machine learning approach to volatility forecasting. Journal of Financial Econometrics, 21(5), 1680-1727.

[12] Song, Y., Tang, X., Wang, H., & Ma, Z. (2023). Volatility forecasting for stock market incorporating macroeconomic variables based on GARCH-MIDAS and deep learning models. Journal of Forecasting, 42(1), 51-59.

[13] Xu, K., Wu, Y., Jiang, M., Sun, W., & Yang, Z. (2024). Hybrid LSTM-GARCH Framework for Financial Market Volatility Risk Prediction. Journal of Computer Science and Software Applications, 4(5), 22-29.

[14] Doering, J., Fairbank, M., & Markose, S. (2017, September). Convolutional neural networks applied to high-frequency market microstructure forecasting. In 2017 9th computer science and electronic engineering (ceec) (pp. 31-36). IEEE.

[15] Bucci, A. (2020). Realized volatility forecasting with neural networks. Journal of Financial Econometrics, 18(3), 502-531.

[16] Ge, W., Lalbakhsh, P., Isai, L., Lenskiy, A., & Suominen, H. (2022). Neural network–based financial volatility forecasting: A systematic review. ACM Computing Surveys (CSUR), 55(1), 1-30.

[17] Kristjanpoller, W., & Minutolo, M. C. (2018). A hybrid volatility forecasting framework integrating GARCH, artificial neural network, technical analysis and principal components analysis. Expert Systems with Applications, 109, 1-11.

[18] Chen, W. J., Yao, J. J., & Shao, Y. H. (2023). Volatility forecasting using deep neural network with time-series feature embedding. Economic research-Ekonomska istraživanja, 36(1), 1377-1401.

[19] Wang, Y., Liu, H., Guo, Q., Xie, S., & Zhang, X. (2019). Stock volatility prediction by hybrid neural network. IEEE Access, 7, 154524-154534.

[20] Liu, Y. (2019). Novel volatility forecasting using deep learning–long short term memory recurrent neural networks. Expert Systems with Applications, 132, 99-109.

[21] Xu, K., Wu, Y., Xia, H., Sang, N., & Wang, B. (2022). Graph Neural Networks in Financial Markets: Modeling Volatility and Assessing Value-at-Risk. Journal of Computer Technology and Software, 1(2).

[22] Chen, Q., & Robert, C. Y. (2022, November). Multivariate realized volatility forecasting with graph neural network. In Proceedings of the third acm international conference on ai in finance (pp. 156-164).

[23] Liu, Y., Qin, Z., Li, P., & Wan, T. (2017). Stock volatility prediction using recurrent neural networks with sentiment analysis. In Advances in Artificial Intelligence: From Theory to Practice: 30th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2017, Arras, France, June 27-30, 2017, Proceedings, Part I 30 (pp. 192-201). Springer International Publishing.

[24] Amar Rao,Dhairya Dev,Aeshna Kharbanda,Jaya Singh Parihar & Dariusz Sala. (2024). Mineral policy and sustainable development goals: Volatility forecasting in the Global South's minerals market. Resources Policy105337-105337.

[25] Gholamreza Hesamian,Arne Johannssen & Nataliya Chukhrova. (2024). A neural network-based ARMA model for fuzzy time series data. Computational and Applied Mathematics(8),445-445.

[26] Wei Wang,Yan Wei Fan & Xiu Hui Qi. (2013). Control and Monitor Model of the Critical State of Corporate Strategic Decision Making Based on the Chaotic Neural Networks. Key Engineering Materials(567-567),101-111.