INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 20, Issue: 5, Month: October, Year: 2025
Article Number: 6774, https://doi.org/10.15837/ijccc.2025.5.6774

CCC Publications

Predicting Stock Trends using Convolutional Neural Networks and Multi-indicator Feature Engineering

Wenbo Li, Long Zhang, Xiangmin Zhang

Wenbo Li
Faculty of Business
City University of Macau,China
B23092100108@cityu.edu.cn

Long Zhang*
1. School of Economics and Management
China University of Geosciences , Beijing, China
2. Faculty of Business
City University of Macau,China
*Corresponding author: *zhangl@cugb.edu.cn

Xiangmin Zhang
School of Economics and Management
China University of Geosciences , Beijing, China
51718642@qq.com

Abstract

Stock data is a typical class of time series data, which is characterised by complex formation mechanism and rich time granularity, and due to its financial attributes, it has always been a hotspot and a difficult point in time series research. Based on feature engineering. This study proposes a novel approach to predicting stock price movements using convolutional neural networks (CNNs) and feature engineering. We select key technical indicators through random forest modeling and transform multi-indicator time series into composite images using an enhanced Gram angular field method. These images are then used to train CNNs for predicting stock trends. Experiments on different time granularities of the GEM index (399006.SZ) demonstrate prediction accuracies exceeding 85% on both training and validation sets. This approach effectively captures temporal features and multi-dimensional information in financial time series, offering improved predictive performance over traditional methods. However, the model's performance is contingent on sufficient data availability, suggesting areas for future research.

Keywords: stock time series, random forest, feature engineering, convolutional neural network.

1 Introduction

Time series data is widely applied across various domains, with stock data emerging as a hot topic for research due to its standardized nature and availability. However, the complexity and randomness of stock data present challenges for traditional methods to effectively extract features.In recent years, although convolutional neural networks (CNNs) have achieved significant success in multiple fields, they face challenges in predicting stock time series, as they were originally designed for processing image data and are not directly applicable to one-dimensional, highly stochastic stock data. With the growing development of machine learning technology, the financial industry has increasingly relied on it. An increasing number of scholars are exploring the patterns of stock price movements through machine learning, such as by combining machine learning models with traditional models and continuously improving machine learning models. Currently, research on stock trends is also a significant hotspot.

In early studies, stock time series data that included dates and closing prices were widely used for predictions. Basic trading data such as opening prices, highest prices, lowest prices, trading volumes, turnover, and turnover rates have also been commonly used by scholars as auxiliary indicators for predictions. With the continuous development of big data technology, more historical stock trading data has been utilized for predictions, and technical indicators constructed by financial and statistical experts are being increasingly employed. Wang J et al. selected the Dow Jones Index and the NASDAQ Index as representatives of mature stock markets, and the Hang Seng Index of Hong Kong, the Shanghai Composite Index, and the ChiNext Index of China as representatives of emerging stock markets for study, to compare the different performances of their algorithms in mature and emerging markets[19]. Akita, R. employed Paragraph Vector and Long Short-Term Memory (LSTM) networks to analyze financial time series and assist investors in making decisions based on various factors, including the Consumer Price Index, Price-to-Earnings Ratio, and various events reported in newspapers. The aim of this research is to convert news articles into distributed representations and utilize LSTM to simulate the time-sensitive impact of past events on the opening prices of multiple companies[1]. Wang J et al. have proposed a novel multi-scale nonlinear ensemble learning framework for stock price prediction. This framework combines variational mode decomposition (VMD), evolutionary weighted support vector regression (EWSVR), and long short-term memory network (LSTM) [18].Smith, J et al. explores the application of various machine learning algorithms in stock market prediction, including Random Forest, Gradient Boosting Trees, and Deep Neural Networks. Through the analysis of historical data, the authors evaluate the performance of these algorithms in predicting short-term fluctuations in stock prices and discuss the impact of algorithm selection, parameter tuning, and feature engineering on the prediction results.[16].Li, W et al. propose a novel neural network architecture capable of automatically learning useful feature representations from raw data and utilizing these features to construct an efficient stock price prediction model. The experimental results demonstrate that the proposed model achieves higher prediction accuracy than traditional methods across multiple stock market datasets. Furthermore, the study explores the impact of different types of autoencoders on prediction performance and proposes corresponding optimization strategies [12].

In recent years, many scholars have also improved machine learning methods for predicting stock market trends. CUI used a piecewise linear representation method that combines turning points and maximum absolute deviation points in the stock time series to extract sequence features, and conducted a detailed discussion and comparison of different sequence lengths, different thresholds, different methods, and different industrial stock data in various fields[5].Williams, B. et al. empirically demonstrated the advantages of hybrid models over singular forecasting techniques and discussed how to select and combine different forecasting methods to achieve optimal forecasting performance. Additionally, the study considered the robustness and interpretability of the models, as well as how to reduce computational complexity while maintaining forecasting accuracy[21].Yang, H. et al. propose a financial time series data mining method based on trend division. Through trend division and extreme point extraction at different time granularities, they construct a stock time series decision table and use support vector machines for verification. This method effectively solves the overfitting problem that traditional methods encounter when dealing with large datasets and data with different time granularities, providing new ideas and approaches for financial time series research.IDREES proposed a method that combines fuzzy transfer learning to predict stock market time series and found that fuzzy transfer learning with smoothing can effectively improve prediction accuracy when stock market data fluctuates over time[9].LIU H, LONG Z H proposed a new framework for stock closing price prediction with higher prediction accuracy, including data processing, deep learning predictor, and prediction optimization method[13].Tang, L. et al. proposed a novel computational intelligence model, EPAK, for predicting univariate time series, as well as a complex prediction model for stock market indices that utilizes EPAK as its core to integrate predictions across all industry indices. The authors conducted empirical validation of the EPAK model using real historical data from the Chinese stock market, assessing its performance in terms of prediction accuracy, stability, and robustness by comparing it with traditional prediction methods. The results demonstrated that the EPAK model exhibited superiority in multiple aspects.

Because stock prediction has the characteristics of time series, many literature or related research analyses and predictions will use the LSTM model for prediction, and many related studies also optimize LSTM in different ways or make more effective feature selection for it. Li Jinxuan et al. proposed a stock prediction method based on multi-perspective feature data, using multiple independent Long Short-Term memory networks (LSTM) to calculate the stock data of each perspective, and adaptively integrating the contributions of different perspectives to the prediction results through ensemble learning[11]. Wei et al. proposed a deep learning framework for predicting stock prices that combines wavelet transformation, stacked auto-encoders, and Long Short-Term memory (Long short-term memory, LSTM), and the experiment proved that the model improved prediction accuracy and profitability[3]. GUO proposed the concept of data trends and proposed a neural network method for predicting future time series data, extracting two datasets to obtain trend datasets and residual datasets, thereby obtaining two separate learning sets for training[6]. Bandhu, K. C., et al. proposed an improved technique for real-time stock price prediction that combines stream processing and deep learning. They utilized Long Short-Term Memory (LSTM) networks to forecast the stock prices of Apple Inc., and went through multiple stages of processing and modeling including data cleaning, feature selection, feature scaling, model construction, model evaluation, model refinement, and prediction. The experimental results showed that the model exhibited high accuracy and precision in predicting stock prices for the next 30 days[2]. Hou Yani explored and verified a stock price prediction model based on the bidirectional Long Short-Term memory network (Bi-LSTM). This study focused on solving some of the challenges in stock price prediction, such as non-linearity, volatility, and other issues that make the accuracy of traditional statistical methods in predicting stock prices not ideal[8].

As an important part of deep learning, the CNN algorithm has also been applied to various fields in recent years, among which the CNN model that has performed well in the field of image recognition has attracted attention. More and more research is expanding the CNN model and applying it to various scenarios, such as face recognition, plant and animal classification, natural language processing, etc., and many studies are also used to deal with time series, such as dealing with sound time series, digital signals, etc., and there has also been some development in the financial field. Wμet al. propose an innovative prediction algorithm that combines Convolutional Neural Networks (CNN) with Long Short-Term Memory networks (LSTM) based on a graph structure, named SACLSTM. This method significantly enhances the accuracy of stock price prediction by integrating historical data and its leading indicators, such as options and futures[22]. Tsantekidis, A.et al. explored the feasibility of using CNN to predict stock prices from the limit order book. By extracting feature images from the limit order book as input for the CNN, they demonstrated the ability of CNN to capture market microstructure information and successfully applied it to stock price prediction[17]. Choudhury A proposed a deep learning system that uses various data from a subset of stocks on the NASDAQ exchange to predict stock prices. The system's framework allows the use of Variational autoencoders (VAE) to eliminate noise and uses time series data engineering to extract higher-level features[14]. KIRISCI proposed a new CNN-based prediction model that can be applied to certain time series and can successfully extract their features in the prediction process. The proposed CNN prediction model consists of three convolutional layers and five fully connected layers, and can also determine the nonlinear relationship between inputs and outputs. ReLu and Elu activation functions are also used[10]. Yu Chen et al. combined convolutional neural networks (CNN), Bidirectional

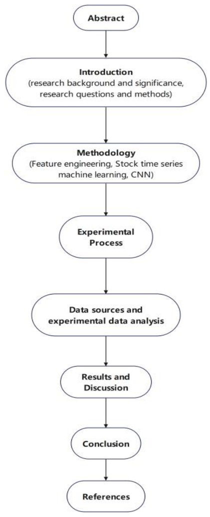
Long Short-Term memory networks (BiLSTM), and at-tension mechanism (AM) to propose a new stock price prediction model-CNN-BiLSTM-ECA. More specifically, CNN is used to extract deep features of stock data to reduce the impact of high noise and nonlinearity. The BiLSTM network is used to predict stock prices based on the extracted deep features[4]. At the same time, a new efficient channel attention (ECA) module is introduced in the network model to further improve the network's sensitivity to important features and key information. Zhang Xiaodong et al. built a convolutional neural network model based on deep decomposition mechanism and attention mechanism (FA-CNN) to improve the prediction accuracy of stock price movements through enhanced feature learning. Unlike most previous studies that only focus on the temporal features of financial time series data, their model also extracts the intraday interactions between input features. In terms of data representation, sub-industry indices are used as supplementary information for the current state of the stock because there is a stock price linkage between individual stocks and their industry indices[23]. Seo, H. et al. propose an innovative high-frequency trading strategy based on Deep Convolutional Neural Networks (CNN). CNNs possess robust capabilities to extract crucial feature information from market microstructure data, such as limit order books and trading volumes, which are vital for high-frequency trading. Leveraging the unique advantages of CNNs in identifying patterns and features within complex data, the authors have constructed an intelligent system capable of automatically recognizing trading opportunities and swiftly adjusting strategies. This system is adept at navigating the rapid fluctuations of high-frequency trading markets, enabling precise and real-time decision-making, ultimately enhancing trading profitability and risk management capabilities[15]. Trang-Thi Ho and Yennun Huang proposed a multi-channel collaborative network that combines candlestick charts and social media data for stock trend prediction. First, emotional characteristics are extracted using natural language toolkits and sentiment analysis to analyze social media. Then, the historical time series data of the stock is transformed into a candlestick chart that illustrates the stock movement pattern, and finally, the emotional characteristics of the stock and its candlestick chart are integrated to predict the stock price movement in 4-day, 6-day, 8-day, and 10-day time periods[7].

Through a comprehensive review of the aforementioned literature, we discern that time series analysis has emerged as an increasingly popular and autonomous research field. Traditional time series studies, however, often rely on a singular or a combination of models to interpret or characterize time series dynamics, leading to predictions that, though effective within a limited timeframe, suffer from poor generalization capabilities (i.e., effective only on trained data but faltering with new data), thus falling short of catering to the demands of the big data era.

Despite convolutional neural networks' (CNNs) exceptional performance in image recognition, their application in financial time series forecasting, particularly for stock index futures, has been constrained by their inability to effectively extract temporal information from raw numerical time series data. In the context of stock index futures price prediction, leveraging raw time series values as inputs has neglected the potential of image-based approaches. While domestic research has explored converting financial time series into images, particularly K-line charts, for CNN models to enhance prediction outcomes, there remains ample room for efficiency optimization.

This study delves further into the realm of deep learning by introducing CNN algorithms into the financial domain, specifically exploring their application in stock market prediction and forecasting stock trends. Drawing on the principles of feature engineering, we meticulously select stock indicators to curate a more precise dataset, theoretically enhancing the predictive accuracy of CNNs. To this end, we refine the Gramian Angular Field (GAF) methodology, transforming multi-indicator stock time series into composite images. This innovative approach harnesses CNN models to evaluate their capability in achieving precise predictions of stock price movements, uptrends, and downtrends.

The structure of the article includes the following parts:



2 Methodology

2.1 Feature engineering

2.1.1 Concept

Feature engineering is a process that involves processing and extracting the properties of the raw data into features that better express the essence of the problem, so as to be better used by the algorithm or model. The definition of feature engineering is when found data set, if the dependent variable Y affected by the independent variable X, the dependent variable Y X can be called the characteristics of the feature engineering is through certain methods to decompose and aggregate the important features of the original data set, through this method, the characteristics can be more easily express problem nature, and can apply these characteristics to different problems of model prediction, improve the prediction accuracy of invisible data, this is the main purpose of feature engineering.

2.1.2 Research steps

Feature engineering is mainly divided into the following steps:

① Data preprocessing: In order to improve the accuracy and effectiveness of the classification algorithm, it is generally necessary to pre-process the data before the classification, including data cleaning, data transformation and correlation analysis.

② Feature selection: select meaningful features for the next machine learning algorithm and model for training. Through different angles, such as whether the feature diverges, the correlation between feature and target, Filter (filtering method), Wrapper (packaging method), Embedded (embedding method), etc.

③ Feature dimension reduction: the process of screening the high-dimensional feature set from the initial data and selecting the low-dimensional feature set, so as to optimize and narrow the feature space according to certain evaluation criteria. For example, principal component analysis method and linear discriminant analysis method can effectively avoid dimensional disasters, avoid the introduction of noise in high-dimensional data, and prevent overfitting.

Stock prices are often influenced by many factors, and the relationship may be non-linear. Feature engineering can help the model to capture these nonlinear relationships. Therefore, feature engineering is very important in stock price prediction, by creating new features or transforming existing features, which can help the model to better understand the data, thus improving the accuracy of prediction. The historical data of stocks is an important basis for the future stock price forecast, and the technical index derived from the historical data is a method to judge when to enter the market. Therefore, this paper uses the method of feature engineering to screen the basic indexes and technical indexes of stock data.

2.2 Stock time series machine learning

At present, machine learning models for stock time series have been widely used, and scholars can use different types of models to conduct different aspects of the analysis of stock time series. The most concerned and most intuitive way to predict the stock price is to use the stock price information in the past to predict the future stock price of a certain period of time, and this demand is difficult to achieve in the traditional time series analysis, so we focus on the algorithm of neural network architecture in machine learning. The complex structure of deep learning can effectively organize the input data, and compared with other machine learning methods, it can better avoid overfitting strategies, so it can deal with financial problems more efficiently. Neural networks, however, have many different models according to the scope of work. The following list is simple description and classification.

Table 1: **Common neural network model architectures and its features**

Model name	Model features
Multilayer Perceptron,MLP	The basis of neural networks, used for simple classification or regression problems.
Convolutional neural network,CNN	Feature extraction by using the convolution operation. It is mainly used in the field of computer vision, such as image classification, object detection and face recognition.
Recurrent neural network,RNN	Add the sequence data model based on the convolutional neural network, which is mainly applied to the language model and processing the video data.
Long Short Term memory network,LSTM	At the same time, the model with long time series memory is mainly used for text generation, machine translation, video marking, etc.

2.3 Convolutional neural network

2.3.1 Concept

Convolutional neural network (CNN) is a model based on supervised learning. In the case of supervised learning, each data should have a corresponding label. Similar data, such as video data, multi-volume label image data, etc., may correspond to a label. By extracting the features of these data and expressing them as a data type, the information content can be increased to make learning more efficient. The performance of CNN model can be improved by deepening the number of layers, and this method uses the complex function to extract features in a nonlinear way, which improves the model performance while making the model with complex feature representation capability. However, due to the deepening of layers, not only overfitting (Overfitting) occurs due to the gradient disappearance problem, but also degradation problems (Degradation), in which training losses increase despite the deepening of layers.

2.3.2 Architecture

Number and Types of Layers:

Input Layer: The entry point of the network, usually corresponding to the pixel values of an image or other types of input data.

Convolutional Layer: Responsible for extracting image features. Each convolutional layer contains multiple convolutional kernels (or filters) that slide over the input image, computing dot products within local regions to generate feature maps.

Activation Function Layer: Often uses ReLU (Rectified Linear Unit) as the activation function, which accelerates training and mitigates the vanishing gradient problem. Other popular activation functions include Sigmoid and Tanh, but they may cause the vanishing gradient issue in deep networks.

Pooling Layer: Performs downsampling on feature maps. Common methods include max pooling (Max Pooling) and average pooling (Average Pooling). Pooling layers help reduce the number of parameters, prevent overfitting, and improve the model's generalization ability.

Fully Connected Layer: Located at the end of the network, used for classification tasks. Each neuron in a fully connected layer is connected to all neurons in the previous layer, ultimately outputting a fixed-length vector for classification or other tasks.

Hyperparameters: Kernel Size: Determines the dimension of each convolutional kernel in the convolutional layer. Common sizes include 3× 3 and 5×5.

Stride: Determines the distance the convolutional kernel slides over the input image. A larger stride results in a smaller output feature map.

Padding: Adds extra zero pixels around the input image to maintain the same size of the output feature map as the input. Common padding modes include 'SAME' and 'VALID'.

Pooling Size:Determines the size of the window for pooling operations, with 2×2 being a common choice.

Network Depth: The number of layers in the network. Deeper networks can capture more complex features but are also prone to overfitting and increased computational costs.

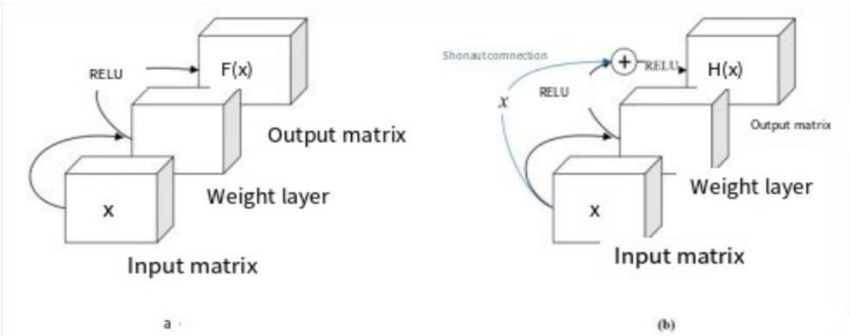


Figure 1: (a) Convolutional neural network model without residual learning
(b) Convolutional neural network model using residual learning

Figure 1 (a) is a convolutional neural network model without residual learning. Compared to figure 1 (a), figure 1 (b) linearly adds the first layer with the shortcut connection (shortcut connection) to obtain the output matrix of the convolutional neural network model. Residual learning In the case of shortcut connection, input x to the feature $F(x)$ through the startup function, as follows:

$$H(X) = F(X) + x \tag{1}$$

(1) It is regarded as an optimization mapping, where x is the input matrix, $F(x)$ is the output matrix of the weight layer, and $H(x)$ is the output matrix. This approach can solve the problem of degradation due to the deepening of layers.

Using the bottleneck method in the network, to figure 2 cases, compared to figure 2 (a) present no bottleneck structure CNN model, figure 2 (b) structure design three bottleneck structure of weight layer bottleneck structure, this structure is characterized by reducing the calculation parameters to reduce time, increase the number of layers of about 4 times, can extract many features, through the 3x3 convolution size 64 layers, reconstructed into 1x1 convolution size 64,3x3 convolution size 64 layers and 1x1 convolution size 256 layers. Thus, substantial information can be provided for this structure.

According to different logical strategy designs and the operating principles behind them, many different types of machine learning models are derived to assist us in all the decisions to enter the stock market. Therefore, this study will also explore the CNN algorithm of deep learning into the financial field, continue to explore the application of convolutional neural network (CNN) in stock prediction, and predict the stock trend.

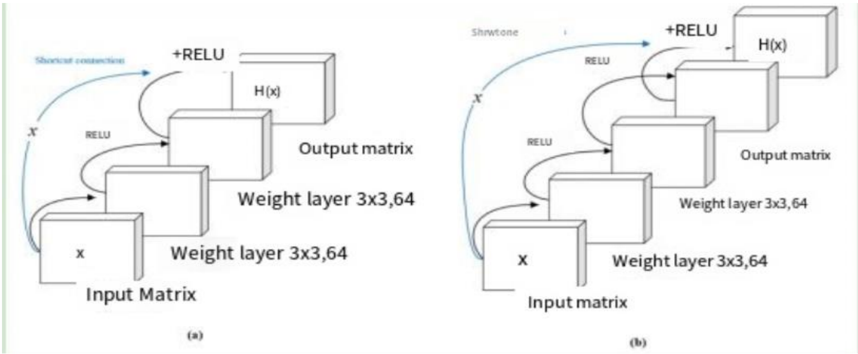


Figure 2: (a) Convolutional neural network model without bottleneck structure
(b) Convolutional neural network model without bottleneck structure

3 Data sources and experimental data analysis

3.1 Data source

In the experiment, the 1 minute line, 15 minute line, 60 minute line and daily line data from January 13,2020 to January 11,2023 of 399006.SZ GEM index were selected for the convolution network test. Experimental data were obtained from the wind database. The time granularity and basic data selected in this paper are shown in the table below.

Table 2: **Stock data situation**

Data granularity	The number of data
The 1-minute line raw data	174720
The 15-minute line raw data	11651
The 60-minute line raw data	2912
Daily line raw data	728

The selected indicators based on the obtained data are shown in the table below.

Table 3: **Selected indicators**

Basic indicators	Technical indicator
opening price(open)	BIAS
closing price(close)	BOLL
highest price(high)	KDJ
lowest price(low)	MA
trading volume(volume)	MACD
	RSI

3.2 Experimental process

The experiment will adopt the concept of feature engineering, utilizing random forest to filter stock indicators. Subsequently, the gramian angular field (GAF) approach will be employed to transform the filtered multi-indicator time series into an image set. This image set will then be introduced into a convolutional neural network (CNN). By analyzing the results, we aim to determine whether this approach can enhance the prediction accuracy of stock trends.

3.3 Experimental data analysis

3.3.1 Establish a random forest model

First, a random forest model is built to predict the daily rise and fall of the stock (rise is "1", fall is "0"), and output the characteristic important values.

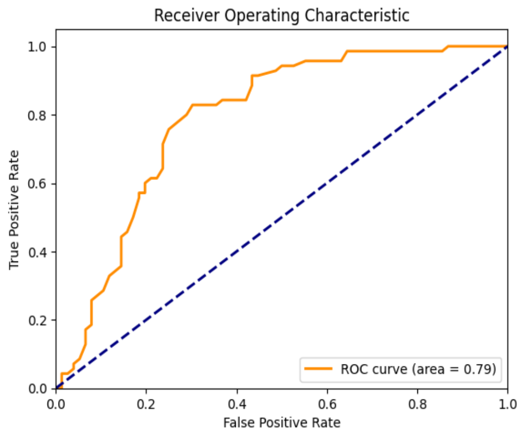


Figure 3: The ROC curves for the random forest

The random forest model predicted stocks with an accuracy of 76.02%. By exporting the ROC plot, where the x-axis is the false positive rate (False Positive Rate) and the y-axis is the true positive rate (True Positive Rate). The area under the curve (AUC) indicates the performance of the model. The closer the AUC is to 1, the better the performance of the model. It can be seen that the accuracy of the model is greater than 50%, indicating that the selected technical indicators can predict the rise and fall of the stock.

Table 4: **The characteristic values of random forest output indicators**

order number	Feature	Importance
1	RSI	0.210968
2	KDJ	0.157933
3	BIAS	0.153103
4	open	0.081858
5	MACD	0.07979
6	volume	0.0783
7	BOLL	0.064089
8	low	0.060568
9	MA	0.05862
10	high	0.054772

Through random forest modeling, it can be found that the volume indicators in the basic indicators and the RSI, KDJ and BIAS indicators in the technical indicators can better reflect the rise and fall of the closing price. Therefore, in the subsequent modeling, combined with the definition of several indicators, the data set of close, volume, RSI and BIAS is selected.

3.3.2 Picture of multiple index time series based on Gram angle field

(1) Time series image visualization method —— gram angle field

Gram angle field (GAF) is a new method to convert time series into images. It was first proposed by Zhiguang Wang and Tim Oates et al. In the paper, the time series was transformed into images by GAF method, and then attributed and analyzed by CNN. The author conducted classification tests

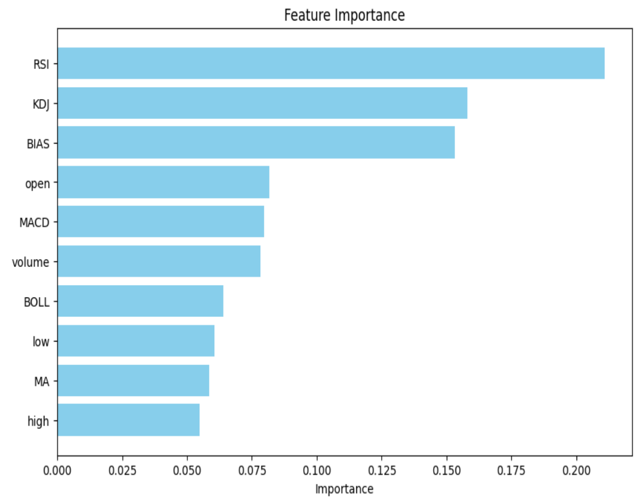


Figure 4: Random forest output indicator feature values

on multiple data sets, and achieved good result[20]. . After this, some foreign scholars introduced it into the financial time series prediction. This study also refers to the coding method of time series proposed by Wang et al.: The practice of GAF matrix and GAF matrix can be divided into three steps: data normalization, coordinate axis transformation and trigonometric function. The details are as follows:

① Data Normalization

After standardizing the given time series N data $x = \left\{x_{(1)}, x_{(2)}, \ldots, x_{(N)}\right\}$ in the following formula, the value is in the $[-1, 1]$interval:

$$\tilde{x}_{(i)} = \frac{\left(x_{(i)} - \max(x)\right) + \left(x_{(i)} - \min(x)\right)}{\max(x) - \min(x)} \tag{2}$$

Example: After the one-dimensional time series [23,56,32,46,6,67,77,25,66,34] (left figure) is data normalization, the value falls within the [-1,1] interval (right figure).

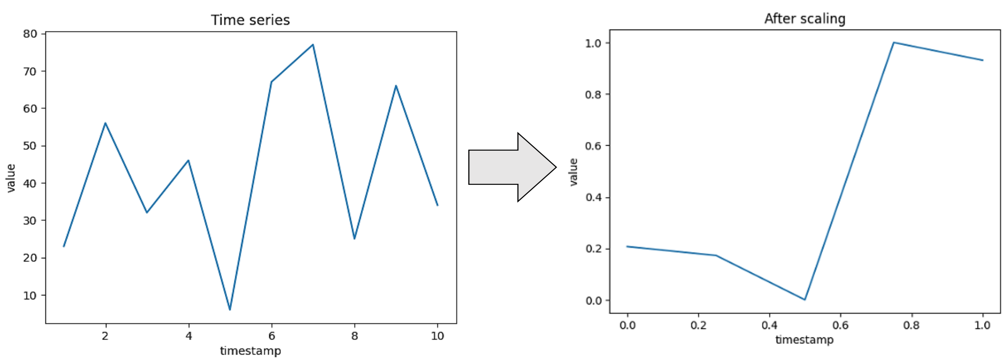


Figure 5: Time series normalization

② The transformation of the coordinate axes

This step transforms the regularized time series data $\tilde{x}(i)$ into a polar coordinate system by the following equation:

$$\begin{cases} \varphi_{(i)} = \arccos(\tilde{x}(i)), -1 \leq \tilde{x}(i) \leq 1 \\ \hline r_{(i)} = \frac{i}{N}, i = 1, 2, ..., N \end{cases} \tag{3}$$

Where $\varphi(i)$,$r(i)$ are the Angle of the polar coordinate system and the radius coordinate of the polar coordinate system, respectively, and the value of $\varphi(i)$ is in the interval of $[0, \pi]$.

For example, Figure 6 presents the normalized time series (left) into a polar system and polar image (right).

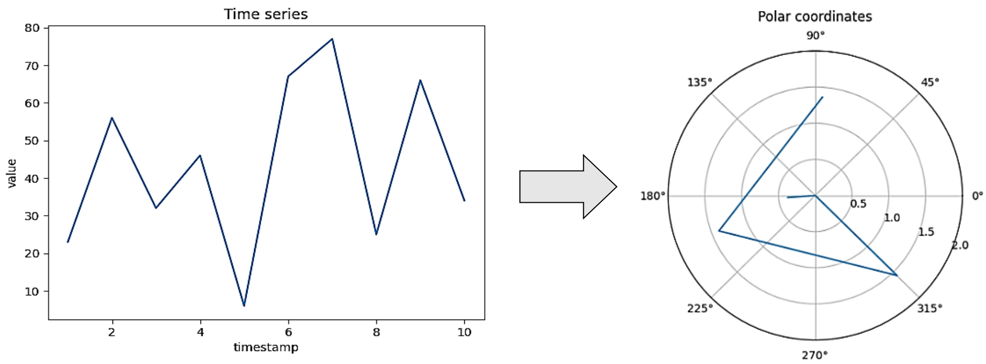


Figure 6: The normalized time series is converted to a polar coordinate system

③ Trigonometric function

Wang et al. (2015) In order to maintain the time dependence (Temporal Dependency), a method is proposed: use the two angular difference sinusoid function to obtain gramian difference angular field (GADF),$DF$is specifically expressed as:

$$GADF = \begin{bmatrix} \sin\left(\varphi_{(1)} - \varphi_{(1)}\right) & \dots & \sin\left(\varphi_{(1)} - \varphi_{(N)}\right) \\ \sin\left(\varphi_{(2)} - \varphi_{(1)}\right) & \dots & \sin\left(\varphi_{(2)} - \varphi_{(N)}\right) \\ \dots & \dots & \dots \\ \sin\left(\varphi_{(N)} - \varphi_{(1)}\right) & \dots & \sin\left(\varphi_{(N)} - \varphi_{(N)}\right) \end{bmatrix} \tag{4}$$

$GADF$ Extract the association between time i and time j, i, j = 1,2,..., N

For example, figure 7 shows that after the angle value in the polar coordinate system is converted by the $GADF$ subtype, the heat map of the matrix is [-1,1], and the color changes from purple to yellow.

(2) Transformation and processing of multi-index stock time series image

In this paper, the method proposed by Zhiguang Wang and Tim Oates (2015) is set to generate a combined picture of multiple indicators. The principle is as follows.

Time series data of 128 daily stocks were selected as calculation examples. The data include four technical indicators, namely, closing price (close), trading volume (volume),bias Rate (BIAS), and relative strength index (RSI).

After converting different index data into pictures and combining, there are two methods for gram angle field, GADF and GASF. In this paper selects the picture generation method of GADF according to the subsequent experimental results.

In this paper, through the method of Gram angle field, the data of different indicators are set as appropriate pixels to generate pictures, and then stitched together into a complete picture of 2 * 2.

(3) Multi-index stock time series image

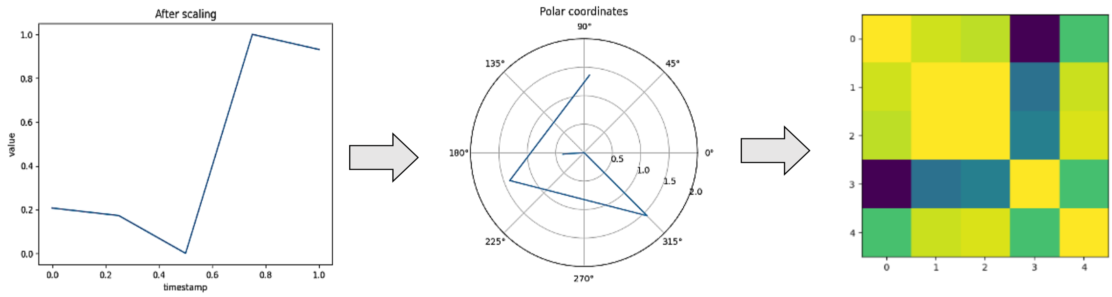


Figure 7: Time series polar coordinates of the scale system is transformed into $GADF$ images

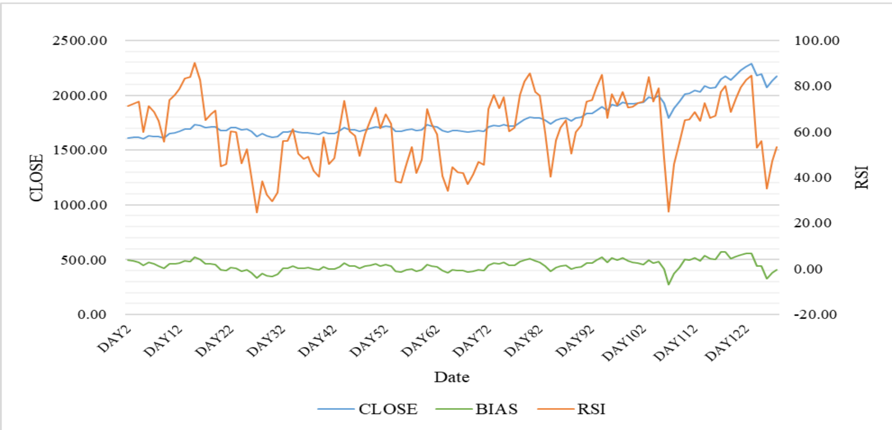


Figure 8: Closing price, convergence and divergence ratio, and relative strength index data

In this paper, the time series trend prediction problem is designed as a classification problem, that is, to predict the trend of stocks, and the corresponding labels are set for each combined picture. If the stock price rises in the forecast range, it will be called "1", and if the price falls, it will be called "0".

Considering the change frequency of the rise and fall of different time granularity, different sliding windows are set for the picture of the stock time series with different granularity, while also considering generating enough pictures to train the convolutional neural network. After generating the picture set, the training set and the test set are divided in a ratio of 7:3.

The number of pictures obtained after the extreme point extraction of the time series with different granularity are as follows:

3.3.3 Multi-index stock time-series convolutional neural network prediction

Based on the above, the stock time series convolutional neural network method based on trend recognition uses the convolutional neural network to carry out picture classification, parameter setting and training. This paper first uses the one-minute raw data and the trend point of 1-minute data to conduct the convolutional neural network experiment. The specific process is as follows.

Table 5: **Example data of multi-index combination pictures (part)**

DATE	CLOSE	BIAS	RSI	VOLUME
DAY1	1609.83	3.82	71.29	6762149100.00
DAY2	1613.46	3.46	72.42	6881003500.00
DAY3	1615.56	2.96	73.15	7119982900.00
DAY4	1600.80	1.60	59.77	6666591800.00
DAY5	1628.12	2.68	71.39	8085836800.00
DAY6	1625.19	2.04	68.83	7556157500.00
DAY7	1620.86	1.26	64.72	7711826600.00
DAY8	1610.90	0.25	55.56	8976472400.00
DAY9	1652.29	2.31	73.95	8592027900.00
DAY10	1660.43	2.33	76.27	8887671100.00
DAY11	1669.41	2.64	78.77	8885206100.00
DAY12	1689.05	3.43	83.37	11598255200.00
DAY13	1692.24	3.19	84.04	9428686800.00
DAY14	1733.23	5.05	90.18	11239712000.00
DAY15	1725.31	4.00	82.79	10976119700.00

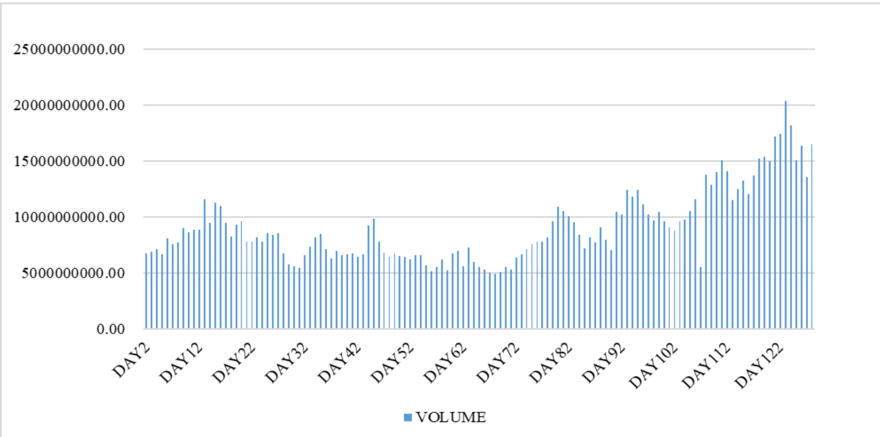


Figure 9: Volume index data

Table 6: **Number of images extracted from the extreme points of granularity at different times**

Data granularit	Image pixel gliding window	The number of data	Number of rising trend pictures	Number of downtrend pictures	Generate the total number of images
1 Minute line data	240,240	174720	383	345	728
The 15-min line data	32,16	11651	396	331	727
The 60-min line data	32,4	2912	416	305	721
Daily line data	8,2	728	252	189	441

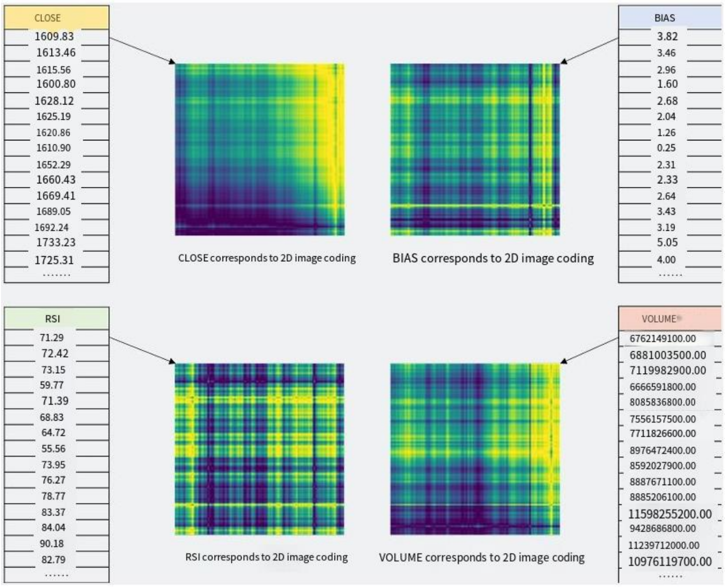


Figure 10: The 4 technical indicators generate pictures with 128 * 128 pixels respectively

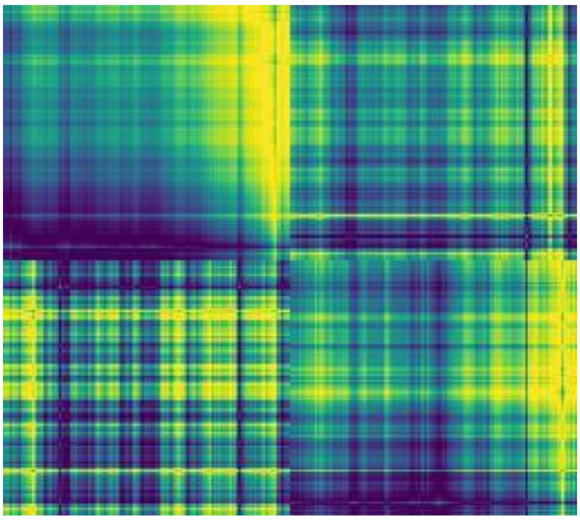


Figure 11: Combined pictures generated by encoding of 4 technical indicators

Convolutional neural network is used to classify the pictures according to the trading strategy. The specific parameters are set as follows:

① Set 3 convolutional layers and 3 maximum pooling layers, activate the function using ReLU, convolution and pool after input full connected layer, and create a convolutional check of input data using Conv2D for convolution calculation.

② Use MaxPooling2D to maximally pool the output of the convolutional layer, level the data at that level using Flatten, and Dense provides a fully connected standard neural network.

③ The compile is used to compile the neural network models, generate models that can be trained, and specify the loss function (loss), optimizer (optimizer) and model evaluation criteria (metrics).

In this paper, convolutional neural network experiments will be conducted for different time granularity (1 minute, 15 minutes, 60 minute line and daily line). The specific results are as follows.

(1) Experimental results of convolutional neural network for 1-minute line data:

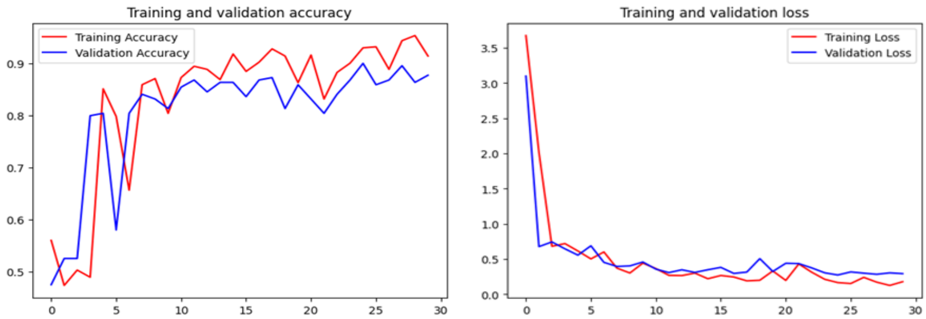


Figure 12: 1-minute line accuracy curve and loss value curves

As can be seen from the accuracy curve of the 1-minute line data, the accuracy of the training set fluctuates greatly, indicating that the model is learning how to classify the data. After the initial fluctuation, the accuracy curves both increase rapidly, which indicates that the model starts to learn useful features from the data in the data set and performs better in the training set; after the first 10 epochs, the accuracy increase begins to slow down and the two curves begin to plateau, indicating that the accuracy rate at this time can reflect the best performance of the model. The training accuracy was consistently slightly higher than the validation accuracy throughout the training gap, suggesting that the model may be slightly overfitting but not seriously.

As can be seen from the loss value curve of the 1-minute line data, the training set loss value and the validation loss value are both decreasing rapidly, in the first few epochs, which indicates that the model is learning rapidly at this stage and making significant progress in reducing prediction error; after a rapid decline, the loss curve begins to plateau, indicating that the model is converging. Both the training loss and validation loss gradually approached the lower stability value; the training loss and validation loss were very close throughout the training process, indicating that the model has similar performance in both the training and validation sets with no obvious signs of overfitting.

(2) Experimental results of convolutional neural network for 15-minute line data:

As can be seen from the accuracy curve of the 15-minute line, the accuracy of the training set fluctuates greatly, indicating that the model is learning how to classify the data. After the initial fluctuation, the accuracy curves both increase rapidly, which indicates that the model starts to learn useful features from the set in the data set and performs better in the training set; after the first 10 epochs, the accuracy improvement begins to slow down and the two curves begin to plateau, indicating that the accuracy rate at this time can reflect the best performance of the model. During the training process, the training accuracy and the validation accuracy intersect several times, but the difference between the two curves increases after 20 epochs, indicating that the model has some overfitting,

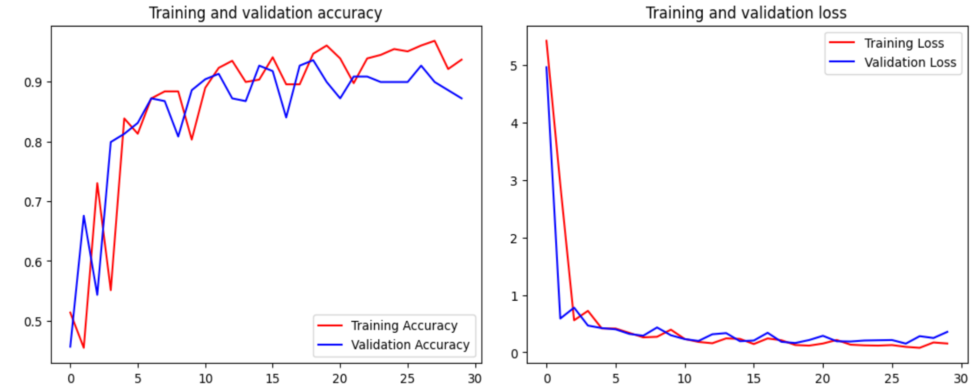


Figure 13: Accuracy curves and loss value curves of the 15-minute line

considering reducing the fitting times to find a more appropriate fitting times.

As can be seen from the loss value curve of the 15-minute line data, both the training set loss value and the validation loss value decrease rapidly, which indicates that the model is learning rapidly at this stage and making significant progress in reducing prediction error; after a rapid decline, the loss curve begins to plateau, indicating that the model is converging. Both the training loss and the validation loss gradually approached the lower stability value; during the whole training process, indicating that the model has similar performance in both the training and validation sets, with no obvious signs of overfitting.

(3) Experimental results of convolutional neural network of 60-min line data:

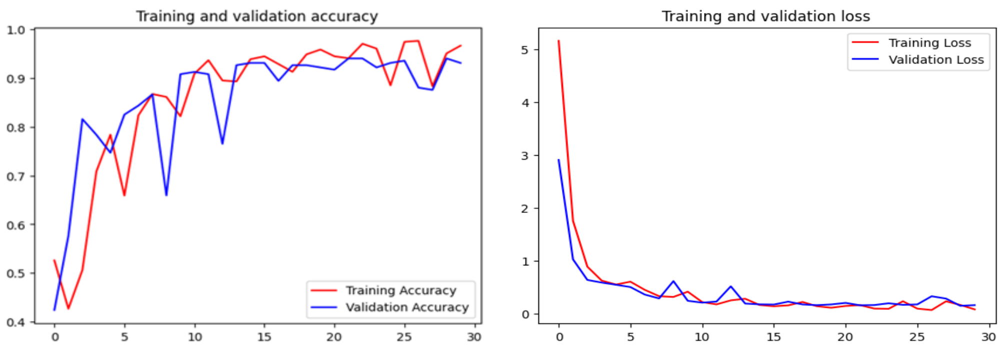


Figure 14: Accuracy curves and loss value curves of the 15-minute line

It can be seen from the accuracy curve of the 60-minute line, since the accuracy of the first 15 epochs, the training set fluctuates greatly. Especially, the accuracy of the validation set varies by 0.25, indicating that the learning ability of the model is not strong enough; however, after the initial fluctuation, the two accuracy curves begin to plateau, indicating that the accuracy at this time can reflect the best performance of the model. During the whole training process, the accuracy of the early training was always slightly higher than the verification accuracy, but the gap was not large.

The two curves in the late period basically overlapped, indicating that the model was not overfit.

As can be seen from the loss value curve of the 60-minute line data, both the training set loss value and the validation loss value decrease rapidly in the first few epochs, which indicates that the model is learning rapidly at this stage and making significant progress in reducing the prediction error; after a rapid decline, the loss curve begins to plateau, indicating that the model is converging. Both the training loss and validation loss gradually approached the lower stable value; and the training loss were very close during the whole training process, indicating that the model has similar performance in both the training and validation sets, with no signs of overfitting.

(4) Experimental results of convolutional neural network of daily line data:

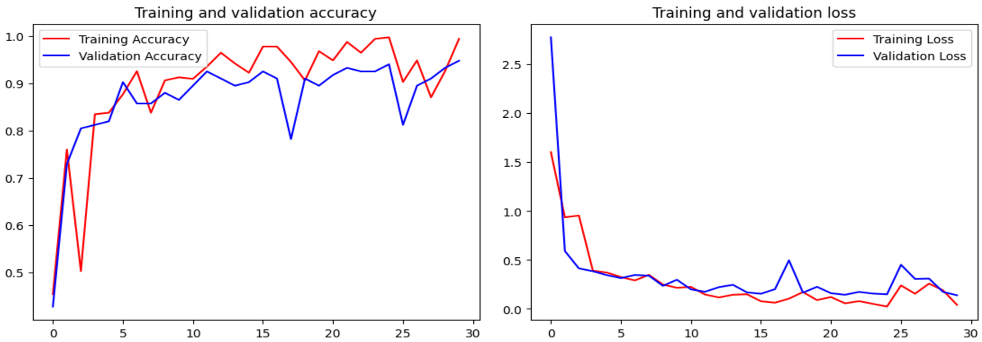


Figure 15: Accuracy curves and loss value curves of the 15-minute line

As can be seen from the accuracy curve of the daily line, the training accuracy was always slightly higher than the validation accuracy, and the gap between 10 and 25 epochs was more obvious, indicating that the model may have overfitting. But as you can see, after five epochs, two accuracy curve always above 75%, the description of the model performance is better, combined with the characteristics of the convolutional neural network, can be analyzed is due to the amount of data, makes the model is not enough data to learn, not the best performance, subsequent consider to increase the amount of data experiments.

As can be seen from the loss value curve of the daily data, the training set loss and validation loss decrease rapidly during the initial epochs, which indicates that the model is learning rapidly at this stage and making significant progress in reducing prediction error; after a rapid decline, the loss curve begins to plateau, indicating that the model is converging. The loss values in the validation set produce fluctuations between 15 and 20 epochs and 25 and 30 epochs, which with the accuracy curve, indicating that the overall model performance is better but needs to be adjusted again.

4 Results and Discussion

In this paper, the data set generated by combining different indices is utilized to classify convolutional neural networks with varying time granularity. As shown in Table 7, the model achieves a high accuracy rate above 85%, accompanied by low loss values on both the training and validation sets. This indicates that the overall performance of the model is excellent. However, due to limited image availability, the model exhibits significant fluctuations during the learning process. To enhance its performance, future studies can expand the time range or adjust sliding window values to increase image quantity.

Convolutional neural networks have demonstrated good predictive capabilities in various domains; however, their accuracy in time series analysis, particularly financial and stock data prediction, has been relatively low in previous research. The reason behind this lies in the "random walk" char-

Table 7: **Training results of raw data and extreme point data for different time granularity**

Data type	Training set loss value Loss	Training set accuracy Acc	Validate the set loss value Val_Loss	Validate set accuracy Val_Acc
1 Minutes	0.1792	0.9136	0.2937	0.8767
15 Minutes	0.1544	0.9370	0.3571	0.8721
60 Minutes line	0.9663	0.1598	0.2937	0.9309
Day line	0.0413	0.9935	0.1393	0.9474

acteristics commonly observed in stock time series data which make it challenging for convolutional neural networks to identify relevant features during their learning process. Consequently, these models achieve only a 50% prediction accuracy on original data sets. In this study, multiple indicators are combined to generate a more accurate and recognizable dataset that effectively improves prediction accuracy of convolutional neural networks. Furthermore, it was discovered that a certain number of images are necessary for achieving satisfactory prediction accuracy with convolutional neural networks; thus emphasizing that an adequate amount of data serves as a prerequisite for attaining better predictive outcomes.

5 Conclusion

This study demonstrates the effectiveness of combining feature engineering and convolutional neural networks for predicting stock trends. By transforming multi-indicator stock time series into composite images, our approach captures complex temporal and multi-dimensional features of financial data. The model achieves prediction accuracies exceeding 85% across different time granularities, showing promise for practical applications. However, the model's performance is dependent on data volume, highlighting the need for further research into optimal dataset sizes and compositions. Future work should focus on refining the feature selection process, exploring alternative image transformation techniques, and testing the model's generalizability across different markets and economic conditions.

The integration of Convolutional Neural Networks (CNNs) and sophisticated multi-indicator feature engineering, as presented in this article, offers investors and financial analysts a powerful tool to meticulously discern the intricate dynamics and impending trends within the stock market. This refined approach empowers them to make informed investment decisions, mitigating risks and minimizing potential losses. Accurate stock trend forecasting stands as a cornerstone for effective risk management and strategic asset allocation, enabling investors to construct robust portfolios tailored for long-term growth and asset appreciation.

References

[1] Akita, R., Yoshihara, A., Matsubara, T., & Uehara, K. (2016). Deep learning for stock prediction using numerical and textual information, *IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, 1–6,2016.

[2] Bandhu, K. C., Litoriya, R., Jain, A., Shukla, A. V., & Vaidya, S. (2023). An improved technique for stock price prediction on real-time exploiting stream processing and deep learning, *Multimedia Tools and Applications*, 83, 57269-57289.

[3] Chen, W., Jiang, M., Zhang, W. G., & et al. (2021). A novel graph convolutional feature based convolutional neural network for stock trend prediction, *Information Sciences*, 556, 67-94.

[4] Chen, Y., Fang, R., Liang, T.,& et al.(2021). Stock price forecast based on CNN-BiLSTM-ECA Model, *Scientific Programming*, 2021, 1-20.

[5] Cui, X., Hu, J., Ma, Y., & et al. (2021). Investigation of stock price network based on time series analysis and complex network, *International Journal of Modern Physics B*, 35(13), 2150171.

[6] Guo, H., Wang, L., Liu, X.,& et al. (2021). Trend-based granular representation of time series and its application in clustering, *IEEE Transactions on Cybernetics*, 52(9), 9101-9110.

[7] Ho, T. T., & Huang, Y. (2021). Stock price movement prediction using sentiment analysis and CandleStick chart representation, *Sensors*, 21(23), 7957.

[8] Hou, Y. (2021). Stock price prediction based on Bi-LSTM deep learning, *Statistics and Application*, 10(3), 538-546.

[9] Idrees, S. M., Alam, M. A., & Agarwal, P. (2019). A prediction approach for stock market volatility based on time series data, *IEEE Access*, 7, 17287-17298.

[10] Kirisci, M.,& Cagcag Yolcu, O. (2022). A New CNN-Based Model for Financial Time Series: TAIEX and FTSE Stocks Forecasting, *Neural Processing Letters*, 54(4), 3357-3374.

[11] Li, Jinxuan, Du, Junping, & Xue, Zhe. (2021). Research on stock forecasting based on multi-view stock features, *Journal of Nanjing University*, 57(1), 68-74.

[12] Li, W., Wang, S., & Chen, Y. (2022). Deep autoencoder-based feature extraction for stock market prediction, *Applied Soft Computing*, 119, 108035.

[13] Liu, H., & Long, Z. (2020). An improved deep learning model for predicting stock market price time series, *Digital Signal Processing*, 102, 102741.

[14] Roy Choudhury, A., Abrishami, S., Turek, M., & et al. (2020). Enhancing profit from stock transactions using neural networks, *AI Communications*, 33(2), 75-92.

[15] Seo, H., Kim, H. S., & Kim, Y. (2018). High-frequency trading strategy based on deep convolutional neural networks, *Applied Sciences*, 8(20), 3453.

[16] Smith, J., & Johnson, M. (2020). Stock Market Prediction Using Machine Learning Algorithms, *Journal of Financial Technology*, 10(2), 123-145.

[17] Tsantekidis, A., Passalis, N., Tefas, A., Kose, C.,& Polykarpou, I. (2017). Forecasting stock prices from the limit order book using Convolutional Neural Networks, *In 2017 IEEE 19th Conference on Business Informatics (CBI)* , (pp. 7-12).

[18] Wang, J., Zhuang, Z., Gao, D., Li, Y., & Feng, L. (2023). Stock price prediction using multi-scale nonlinear ensemble of deep learning and evolutionary weighted support vector regression, *Studies in Nonlinear Dynamics & Econometrics*, 27(3), 397-421.

[19] Wang, J., & Wang, J. (2015). Forecasting stock market indexes using principle component analysis and stochastic time effective neural networks, *Neurocomputing*, 156, 68-78.

[20] Wang, Z., & Oates, T. (2015). Imaging Time-Series to Improve Classification and Imputation, *In IJCAI'15: Proceedings of the 24th International Conference on Artificial Intelligence* , (pp. 3939-3945).

[21] Williams, B., & Brown, T. (2022). Hybrid models for enhanced stock market forecasting, *International Journal of Forecasting*, 38(4), 1198-1214.

[22] Wu, J. M. T., Li, Z., Herencsar, N., Vo, B., & Lin, J. C. W. (2021). A Graph-based CNN-LSTM Stock Price Prediction Algorithm with Leading Indicators, *Multimedia Systems*, 29(3), 1751-1770.

[23] Zhang, X., Liu, S., & Zheng, X. (2021). Stock price movement prediction based on a deep factorization machine and the attention mechanism, *Mathematics*, 9(8), 800.

Copyright ©2025 by the authors. Licensee Agora University, Oradea, Romania.
This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License.
Journal's webpage: http://univagora.ro/jour/index.php/ijccc/

Member since 2012
JM08090

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:

Li, W.; Zhang, L.; Zhang, X. (2025). Predicting Stock Trends using Convolutional Neural Networks and Multi-indicator Feature Engineering, *International Journal of Computers Communications & Control*, 20(5), 6774, 2025.
https://doi.org/10.15837/ijccc.2025.5.6774