Stock Price Prediction Based on Trend Characterization

Fangjun Huang*

{*corresponding author: huangfa5@msu.edu}

Computer Science, Michigan State University, East Lansing, Michigan

Abstract: In the current economic landscape, stocks have evolved into an essential vehicle for investment and wealth management, with stock trading constituting a vital facet of modern economic activity. Fluctuations in stock prices directly impact investor returns, and the ability to forecast stock price changes could minimize investment risks and augment returns. Consequently, stock price prediction has emerged as a focal point of research. The evolution of machine learning technology has led to the progressive application of these techniques in stock investment decision-making. This study introduces an enhanced KNN model for predicting stock prices. Grounded in the basic principle of KNN, this model restructures the input feature attributes by linking continuous multi-day trading indicators to create a short-term price change trend. This trend is then used as the KNN model's input to predict subsequent trading days' stock prices. Experimental trials indicate that the enhanced KNN model outperforms the comparative algorithms in predictive performance.

Keywords: Machine learning, KNN model, Stock Price prediction

1 Introduction

Given the rapid technological advancements, fields such as artificial intelligence and machine learning have achieved considerable strides in multiple sectors. In finance, applying these advanced technologies to analyze and predict stock markets has emerged as a compelling investment strategy.

The complexity, nonlinearity, and uncertainty surrounding the stock market have piqued the interest of investors. Classic forecasting methods, such as technical and fundamental analyses, provide some insights into market trends. However, these methods need help to keep pace with the constant market flux and provide consistently accurate predictions. This is where machine learning technology comes into play, given its impressive capabilities in processing complex data, identifying potential patterns, and adapting to changes, thereby introducing new opportunities in the finance sector.

At the heart of machine learning lies the principle of using algorithms to learn from extensive historical data. This involves recognizing potential patterns and correlations and using these insights to predict future trends. When predicting stock prices, machine learning methodologies, which include supervised, unsupervised, and reinforcement learning, can extract potential market trends from a wide range of information sources. These could be historical stock market data, company financial statements, news articles, and social media posts, all of which offer

MSEA 2023, May 26-28, Nanjing, People's Republic of China

DOI 10.4108/eai.26-5-2023.2334468

valuable predictions for investors.

In stock price prediction, various machine learning research has been conducted, employing various technical approaches to predict stock prices and provide investors with more precise forecasts.

Some research has investigated the use of Support Vector Machines (SVM) for predicting financial time series data. Through experimental validation, the effectiveness of SVM in predicting stock prices was shown, and the method was compared with other techniques [1]. Other research has delved into a comprehensive analysis of stock market prediction techniques, emphasizing methods such as neural networks, fuzzy logic, and genetic algorithms [2]. By comparing the advantages and disadvantages of various methods, valuable references for further research were provided. Used deep learning networks, such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), to analyze and predict the stock market, proposing a new data representation method and demonstrating the application of deep learning in stock price prediction through case studies [3]. Proposed a deep learning framework based on Stacked Autoencoder (SAE) and Long Short-Term Memory (LSTM) for financial time series prediction, proving its superiority in prediction accuracy and stability through experiments [4]. Applied deep neural networks, gradient boosting trees, and random forests to statistical arbitrage strategies for the S&P 500 index, with experimental results showing high accuracy in stock price prediction [5]. Used trend determinacy data preprocessing and machine learning techniques to predict stock and stock price index trends, applying methods such as Support Vector Machines, random forests, and artificial neural networks to stock price prediction and comparing their predictive performance [6]. Studied the size, value, and momentum characteristics of international stock returns to model and predict the market, finding that these features contain valuable information for stock price prediction [7]. Used Long Short-Term Memory (LSTM) networks to predict stock returns, taking the Chinese stock market as a case study [8]. Experimental results showed that LSTM has good performance in predicting stock returns. Employed artificial neural network models to predict stock market indices [9]. By comparing various network structures and parameter settings, it was found that artificial neural networks have higher accuracy in stock price prediction. Combined numerical and textual information to predict stocks using deep learning [10]. The authors used Convolutional Neural Networks (CNN) to process textual data and Recurrent Neural Networks (RNN) to process numerical data, demonstrating the application potential of deep learning in stock price prediction through experiments.

2 Methodology

2.1 Introduction to KNN Algorithm

The K-nearest neighbor algorithm (KNN) is a widely adopted supervised learning algorithm in machine learning. KNN has no model training process; an unlabeled sample is compared with each sample in the training set during prediction. The algorithm identifies the 'k' points closest to the unlabeled sample and predicts the category of this sample based on the categories of these selected points. The effective application of KNN depends on a careful selection of the 'k' value, the measure of distance between samples, and the rules for deciding the classification. These three factors crucially influence the outcome of the algorithm. KNN's advantages include

simplicity, the absence of a need for training, and a lack of sensitivity to outliers.

When it comes to stock price prediction employing the classic KNN algorithm, the closing price in the stock index is chosen as a feature attribute, and the closing price of the trading day following the sample data of trading day is selected as a label attribute. That means each sample data consists of two attributes: first one is the closing price for the current day, and the second one is the closing price for the next trading day.

Assuming there are 'n+1' trading day indicators, with record numbers from 0 to n. The first 'm+1' trading day data is selected as the training set. Given that the trading day with the record number 0 lacks previous trading day data, the training set comprises 'm' samples, i.e.,$\{(X_{train}, Y_{train})\} = \{(x'_0, x'_1), (x'_1, x'_2), \dots, (x'_{m-1}, x'_m)\}$. And the remaining data forms the test set. Since the trading day with the record number n has no data for the next trading day, the test set contains n-m samples, i.e.$X_{test} = \{x'_m, x'_{m+1}, \dots, x'_n\}$. In the process of applying the KNN method, the gap between samples is computed as the absolute difference in the concluding prices of the dual attributes of the sample, i.e.,

$$d_{ij} = \left|x'_{i-1} - x'_{j-1}\right| \tag{1}$$

The formula (1) mentioned, d_{ij} is the distance from one point to the other one that mapped in the feature space by the trading day data with record numbers i and j, where $i, j \in N, 1 \le i \le n, 1 \le j \le N$. While making predictions, we suppose that the sample to be forecasted is $x'_{\mathrm{t}}(\mathrm{t} \in \mathrm{N}, \mathrm{m} \le \mathrm{t} \le \mathrm{n})$. Based on the distance calculation formula, we evaluate the distance metric from the point symbolized by the sample to every other point in the testing set. These points within the testing set are sorted in order of increasing distance. The first k points in this sequence represent the k points nearest to the test sample point and are denoted as the set T^*. However, predicting stock prices differs from conventional classification problems as it involves an indefinite continuous value rather than a distinct class. Therefore, the decision rule could be the average of the mapped values of the k nearest points. In other words, the predicted value of a stock's closing price on a specific trading day is the mean of the closing prices of the trading days most proximate to the preceding day's closing price, among the chosen k trading days, i.e.

$$y_t = \frac{1}{k} \sum_{x'_j \in T^*} x'_{j+1} \tag{2}$$

A complete depiction of the algorithm can be found in Table 1.

Table 1: Stock price prediction process based on classic KNN algorithm.

Input: Dataset $\{(X_{train}, Y_{train})\} = \{(x'_0, x'_1), (x'_1, x'_2), \dots, (x'_{n-1}, x'_n)\}$,Test Sample x_t	
1：	For each x'_t， $x'_t \in X_{test}$：
2：	Find the k nearest points to x'_t in X according to formula (2), forming the set T^*.
3：	$y'_t = \frac{1}{k} \sum_{x'_j \in T^*} x'_{j+1}$
Output： y'_t， $m < t \le n+1$	

Classic KNN algorithms in stock price prediction indeed have some shortcomings: the algorithm only uses a single day's stock index as a feature attribute and selects samples with the closest features for prediction. This method ignores the differences in subsequent trends of the same stock index. When the feature attributes of the test sample and the comparison sample are similar, their trends may be entirely opposite. For example, even though a test sample and a comparison sample may have the same current closing price, one might be on an upward trajectory with the stock price likely to continue rising on the next trading day, while the other might be on a downward path with the stock price expected to continue falling on the next trading day. This disparity in trends can lead to significant inaccuracies in prediction results that rely on the nearest neighbor principle.

2.2 Data Preprocessing

The unprocessed stock data carries a wealth of information, encompassing elements like the opening price, closing price, highest and lowest stock prices each trading day, along with stock price changes and percentage shifts, among others. By preprocessing the data and reorganizing the data format, the hidden trends in stock prices can be revealed. To capture the track of stock price changes, we concatenate the stock index data of consecutive N days into a single sample, which represents a stock index change curve spanning N days. By using samples processed in this way instead of the original training set data, the KNN algorithm can find the closest short-term trend to the predicted sample in the training set, thus making the final stock price prediction. After completing the data preprocessing, if the data from different dates or historical periods have significant differences in magnitude, normalization is required. The fundamental principle of normalization is to perform linear transformations on the data, confining all observation data within a certain range, mitigating the impact of different dimensions on the algorithm's prediction effect, and promoting the rapid convergence of machine learning models. This study adopts the Min-Max normalization method, which applies the following transformation to all stock indices:

$$x_t' = \frac{x_t - min}{max - min} \tag{3}$$

In the formula (3) mentioned, x_t represents the actual closing price value of the trading day with index t in the raw stock data, and x_t' represents the normalized data value with $x_t' \in [0,1]$. Max symbolizes the highest closing prices, and the min symbolizes the lowest one across all stock information, respectively.

2.3 Refined KNN Algorithm Model

Accroding to the issues discussed earlier, this paper enhances the classic KNN model and introduces an refined KNN algorithm model. The enhanced model posits that the stock price prediction for the current trading day depends not just on the previous trading day's stock price, but also on the trend of the preceding period, as reflected by the stock price change curve of the preceding N trading days. If the K nearest samples identified have not only similar current day stock prices but also comparable recent trends, a more precise stock price prediction for the next trading day can be derived based on these K nearest samples. Based on this premise, the enhanced KNN model in this paper no longer merely uses the previous trading day's data as a sample, but rather follows the preprocessing method described in Section 2.2, merging the stock price data of the preceding N days in sequence to form a new sample and generate a new training

set. The inputs of the classic KNN model and the refined KNN model are illustrated in Figures 1 and 2, respectively.

Fig. 1. Training data for the classic KNN model

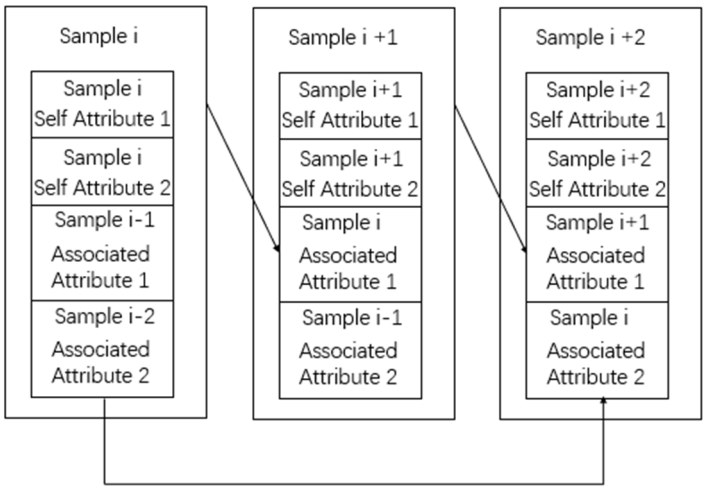

Fig. 2. Training data for the Refined KNN model

As the diagrams illustrate, in the enhanced KNN-based stock price prediction model, the stock price indices of the previous N trading days are appended as an extended "recent stock price change trend" for each trading day's stock price index, and these are utilized for the subsequent trading day's stock price prediction. Following this processing method, assuming the original training set contains m sample data, the new training set X_{train} post-processing includes (M-N) sample data. In X_{train}, the primary component functions as a characteristic, represented as a vector derived from the end-of-day prices of the past N trading days, i.e., $\boldsymbol{x}_i' = [x_{i-N}', x_{i-N+1}', \ldots, x_{i-1}']$, and the subsequent element serves as an identifier, symbolizing the final transaction price of the current trading day. The enhanced KNN model maps a trading day's stock index information into an N-length trend curve, and the measurement of distance between two samples should accordingly be adjusted to the Euclidean distance between a pair of points within a high-dimensional framework, i.e.,

$$d'_{ij} = \sqrt{\sum_{k=1}^{N} \left(x'_{i-1} - x'_{j-1}\right)^2} \tag{4}$$

The decision for deriving the predicted value for a given trading day's stock price remains unaltered from the fundamental KNN model. It is derived from the mean of closing prices of the k trading days that display the closest resemblance to the previous day's closing price, as indicated in equation (2).

Table 2: Refined KNN Algorithm Model for Forecasting Stock Prices

Input: Training set $\{(X_{train}, Y_{train})\} = \{(x'_N, x'_N), (x'_{N+1}, x'_{N+1}), \dots, (x'_m, x'_m)\}$,test sample $X_{test} = \{x'_{m+1}, x'_{m+2}, \dots, x'_{n+1}\}$	
1:	For each x'_t, $x'_t \in X_{test}$:
2:	Find the k points in X_{train} that are closet to x'_t according to Equation (4), forming the setT^*;
3:	$y'_t = \frac{1}{k}\sum_{x'_j \in T^*} x'_{j+1}$
Output: y'_t, $m < t \leq n+1$	

This model considers the trend of stock price changes over the prior period (N trading days) instead of a single point. Given it integrates the trend of price change when searching for similar samples, it theoretically provides enhanced predictive performance. The best value of N will depend on the specific characteristics of different stock datasets. When N=1, the refined KNN algorithm model reverts to the classic KNN model (Table 2).

2.4 Evaluating the Model

Once the model is trained, its performance is assessed using certain evaluation metrics. In this study, the Root Mean Square Error (RMSE) is utilized as the evaluation metric.

Assuming the model's predicted values= $[y'_{m+1}, y'_{m+2}, \dots, y'_{n+1}]$, and the actual values of the test set are$Y_{test} = [y_{m+1}, y_{m+2}, \dots, y_{n+1}]$, The root mean square error can be computed as per the formula:

$$RMSE = \sqrt{\frac{1}{n-m}\sum_{k=m+1}^{n+1} (y_k - y'_k)^2} \tag{5}$$

RMSE signifies the divergence between the forecasted and the actual values. A larger RMSE represents a larger divergence, signifying that the model is less precise. On the other hand, a smaller RMSE indicates less divergence, implying a more precise model.

3 Experiments and Results Analysis

3.1 Experiment Design

An experiment is designed for algorithm testing to substantiate the effectiveness of the enhanced KNN model presented in this paper. The experiment utilizes historical stock data with code 600967, from May 18, 2004, to June 3, 2020, comprising 3900 trading days. After a thorough data cleansing process and eliminating invalid data, 3700 valid data points were procured and normalized. These data points were then sequenced chronologically and were associated with trading days to create a new sample set. To underline the superiority of the enhanced KNN, it is juxtaposed with the logistic regression algorithm and the traditional KNN algorithm. For each comparison model, the first three thousand samples were assigned to the training set, with the remaining samples constituting the test set. RMSE is the chosen evaluation metric for predictive performance.

Throughout the KNN learning process, a grid search method was implemented to optimize the hyperparameter defining the number of nearest neighbors in KNN, thereby enhancing the KNN model's performance. For this experiment, N=30 was selected, postulating that the current trading day's stock price is influenced by the preceding 30 trading days' price change trend.

3.2 Experimental Results and Analysis

The predictive outcomes from the logistic regression algorithm model, the conventional KNN algorithm model, and the refined KNN algorithm model proposed in this study are depicted in Figures 3 through 5.

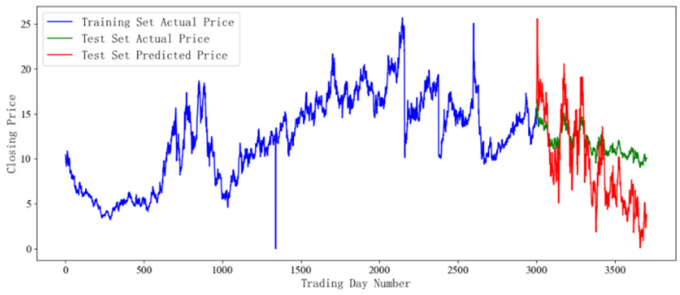

Fig. 3. Prediction of stock data employing the logistic regression algorithm model.

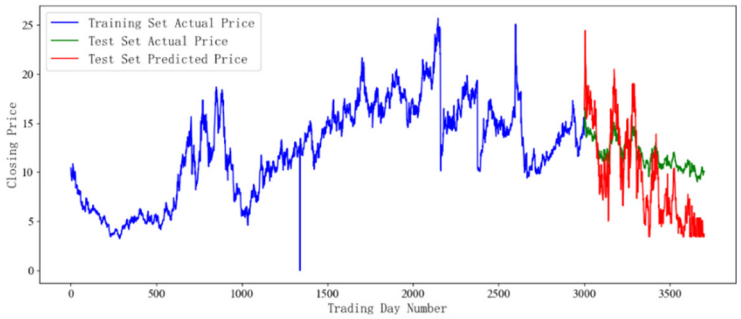

Fig. 4: Prediction of stock data employing the traditional KNN model.

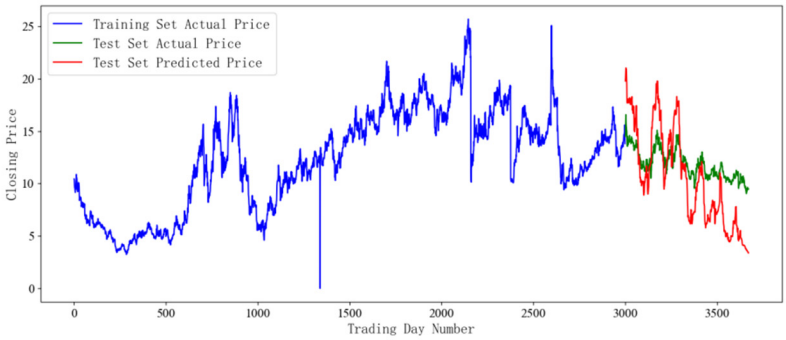

Fig. 5. Prediction of stock data employing the refined KNN model.

In the figures above (Fig.3-5), the blue line illustrates the actual price of the stock samples in the training set, the green line represents the actual price of the stock samples in the test set, while the red line corresponds to the stock prices predicted by each model. Upon comparing the three figures, we find that these algorithms are capable of predicting the trend of stock price changes to a certain extent. The RMSE of each algorithm is calculated individually, with the results displayed in Table 3.

Table 3: Values of Root Mean Square Error (RMSE) for Three Models

Logistic Regression Model	Classic KNN Model	Refined KNN Model
4.221	3.969	3.659

As shown in Table 3, he RMSE value of the logistic regression model is the highest, followed by the classic KNN model, while the refined KNN model exhibits the lowest value. This signifies that the stock price predictions of the refined KNN model are nearer to the actual stock prices in contrast to the forecasts from the remaining models. Therefore, employing the refined KNN model for stock price prediction results in improved predictive accuracy.

4 Conclusion

In addressing the limitations of the traditional KNN model in predicting stock prices, this paper puts forth a refined KNN model. This model remodels the training set data, incorporating the stock price data of the preceding N trading days as an extended feature of a sample for learning and prediction. The experimental outcomes demonstrate that the proposed refined KNN model delivers superior prediction performance compared to the logistic regression and traditional KNN models.

References

[1]Kim, K. J. (2003). Financial time series forecasting using support vector machines. Neurocomputing, 55(1-2), 307-319.

[2]Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques–Part II: Soft computing methods. Expert Systems with Applications, 36(3), 5932-5941.

[3]Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. Expert Systems with Applications, 83, 187-205.

[4]Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS one, 12(7), e0180944.

[5]Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. European Journal of Operational Research, 259(2), 689-702.

[6]Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. Expert Systems with Applications, 42(1), 259-268.

[7]Fama, E. F., & French, K. R. (2012). Size, value, and momentum in international stock returns. Journal of Financial Economics, 105(3), 457-472.

[8]Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. In 2015 IEEE International Conference on Big Data (Big Data) (pp. 2823-2824). IEEE.

[9]Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. Expert Systems with Applications, 38(8), 10389-10397.

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