Expert Systems With Applications 205 (2022) 117595

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A stock time series forecasting approach incorporating candlestick patterns and sequence similarity

Mengxia Liang [a], Shaocong Wu [b], Xiaolong Wang [b,*], Qingcai Chen [b]

[a] Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China
[b] College of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, China

ARTICLE INFO

Keywords:
Stock time series forecasting
Sequential pattern mining
Candlestick pattern
Sequence similarity
Pattern matching

ABSTRACT

This article aims to implement trend forecasting of stock time series based on candlestick patterns and sequence similarity. Financial time series forecasting plays a central role in hedging market risks and optimizing investment portfolios. This is a challenging task, as financial engineering requires the proposed approach to be interpretable, robust, and compatible. It is noted that many published research studies are based on multi-modal data, which makes the prediction approaches increasingly complex, difficult to interpret, and does not allow the migration across different data. Given this situation, it is believed that a candlestick data-based approach is promising. It is already recognized by the technical analyses, prevalent in financial markets, more readily available, and has better interpretability. In this paper, the forecasting approach is divided into two steps. In the first step, sequential pattern mining is used to obtain candlestick patterns from multidimensional candlestick data, and the correlation between different patterns and the corresponding future trends are calculated. In the second step, a new sequence similarity is proposed to match the diverse candlestick sequences with the existing patterns. The method is validated on real data from 800 stocks in the Chinese stock market, which are divided into two groups of experiments, and the average accuracy achieved by the proposed method is 56.04% and 55.56%, which is higher than the SVM model (50.83% and 51.32%) and the LSTM model (50.71% and 50.68%) used for comparison, proving that our work is more stable and accurate. This work is instructive for further research around candlestick data to follow.

1. Introduction

With the globalization and ease of investment in international and national markets, many people are looking towards stock markets for gaining higher profits. Financial time series generated in real-time contains a massive amount of confidential information worthy of mining and analysis (Liang, Wang, & Wu, 2021; Wu, Wang, Liang, & Wu, 2021; Zhang, Yan, & Aasma, 2020). Stock trend prediction is the first preference of investors, which assists them in making more reliable decisions in financial markets (Niu, Wang, Lu, Yang, & Du, 2020). However, there is a high degree of uncertainty in the stock prices (Hu & Zheng, 2020; Huang, Chiou, Chiang, & Wu, 2020), making it difficult for the investors to predict price movements.

Existing approaches are based on experience and econometrics, and as the data grows, the need for advanced technical support becomes more pressing. A financial market is influenced by multiple factors, such as investors' psychological expectations, dynamics of economic development, changes in the industries, national policies, exchange rate, etc. These factors are expressed in different information vehicles, such as news, multimedia, social networks, financial reports, etc. Although influenced by various factors, stock volatility is correlated with its history, enabling scholars to analyze and predict the stock market (Pal & Kar, 2022; Shih, Sun, & Lee, 2019).

The most intuitive idea is to consider extensively all factors that are likely to have an impact, and these approaches can be divided into two categories, one being multi-modality approaches (Cheng, Yang, Xiang, & Liu, 2022) and the other being fuzzy systems (Naranjo, Arroyo, & Santos, 2018). In terms of specific implementations, many researchers have favored deep learning in recent years. Explaining how such complex factors affect financial markets is an arduous task, and the use of deep learning and fuzzy rules (Marszałek & Burczyński, 2014) seems to be the second-best choice for researchers, as both methods have a strong

* Corresponding author.
E-mail addresses: liangmengxia@hotmail.com (M. Liang), wushaocong2013@gmail.com (S. Wu), xlwangsz@hit.edu.cn (X. Wang), qingcai.chen@hit.edu.cn (Q. Chen).

https://doi.org/10.1016/j.eswa.2022.117595
Received 15 March 2022; Received in revised form 21 April 2022; Accepted 12 May 2022
Available online 21 May 2022

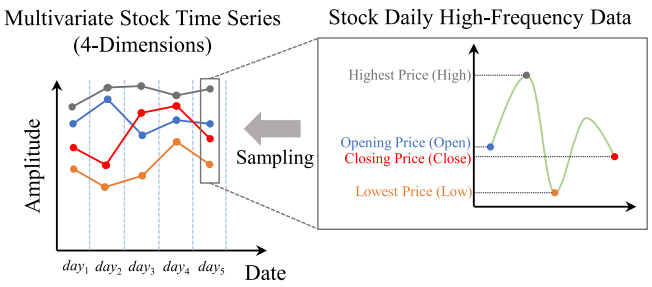


Fig. 1. The high-frequency daily price data of stocks are sampled to obtain multivariate stock time series.

generalization capability and do not rely on artificially summarized features and knowledge. It is generally accepted that as the amount and variety of data used increases, the more accurate the predictions of financial markets will be, and a large body of research has proved this (Meng, Ma, Qiao, & Xie, 2021). However, this also means that the models can be more complex and challenging to interpret, and this reliance on specific data volumes and data types has created technical barriers that make it difficult for the rest of the researchers to replicate these models on their data and research tasks. In short, not all researchers can find enough data to drive a complex model.

With this in mind, a forecasting method that meets the requirements of financial engineering, with interpretability, robustness, and compatibility, is needed to be proposed. It is noted that candlesticks are a type of data that meets these needs. Candlesticks is a technique for plotting past price action of a specific underlying such as a stock, index, or commodity using open, high, low, and close prices. The candlestick is considered as a special representation of a stock's multidimensional price series, and it is essentially a daily sampling of high-frequency price data, as shown in Fig. 1.

Candlesticks are the basis of technical analysis, and in this paper, each candlestick represents price data for a single trading day (which can also be constructed using shorter or longer intervals). Each candlestick shows the four essential prices of the day on its pattern structure through certain mapping rules: the area between the open and the close is called the 'body', the price fluctuations above and below the body are "shadows" (also called wicks), and the shadows indicate the highest and lowest prices at which the stock has traded during the time interval represented (Fig. 2). The advantages of candlesticks are their effectiveness in expressing trend signals and the low difficulty in obtaining data, making daily price data of stocks easily obtained by almost all researchers (Madbouly, Elkholy, Gharib, & Darwish, 2020; Udagawa, 2019; Wang & Wang, 2019).

The popularity of candlesticks has proliferated over the past decade. Because of the density of information and the ability of candlesticks to represent trading patterns over short periods, many investors invest by studying candlestick information. In the empirical analysis, sophisticated investors often predict stock movements based on the candlestick series, confirming the correlation between candlesticks and stock movements.

A candlestick pattern is a particular sequence of candlesticks on a candlestick chart, mainly used to identify trends. Candlestick patterns emerge because human behavior and responses are patterned and repeated throughout history (Chen & Tsai, 2020). The candlestick pattern captures this information and represents it as a single candlestick or a combination of candlesticks of variable length (Udagawa, 2018).

However, mining candlestick patterns from historical data is still a challenging task. Candlestick pattern mining is not equivalent to statistical high-frequency candlestick combinations, and influences such as corresponding trends, noise, pattern deformation, and temporal order also need to be considered. As the number of candlesticks included in a candlestick pattern increases, the number of possible pattern forms increases geometrically, and it is impractical to use the "one by one" comparison alone. In addition, candlestick patterns mined through historical data do not match every new sequence precisely. Over time, stock prices will always present new series that have never been seen before. Another problem solved is mapping the existing "candlestick patterns-trend" correlation knowledge to the latest candlestick sequence.

In summary, the innovations and contributions of our work comprise the following five points:

- A morphology-based candlestick encoding method is proposed to reduce the dimensionality and preserve the integrity of the candlestick data (each candlestick is considered a unit with a unique structure), thus making it possible to mine patterns in multivariate time series.
- Segmenting the series based on price reversal points is proposed to help determine the location, the length, and obtaining the future trend of candlestick patterns.
- The correlations between candlestick patterns and trends were calculated through a combination of subsequence matching and Self-Comparison.
- New sequence similarity is used to match the existing patterns with new sequences to enable the trend forecasting of stocks.
- Eight hundred representative stocks were obtained from the Chinese stock market's the CSI 300 and the CSI 500 indices, and the corresponding candlesticks data was collated. These data were cleaned and calibrated and made public in https://github.com/shaocongWu/Multivariate_Stock_Time_Series_Dataset. Experiments on our approach and some traditional models such as SVM and LSTM were performed on the whole dataset.

The paper is organized as follows. Given that our work involves several related research tasks, in Section 2, an analysis of related work

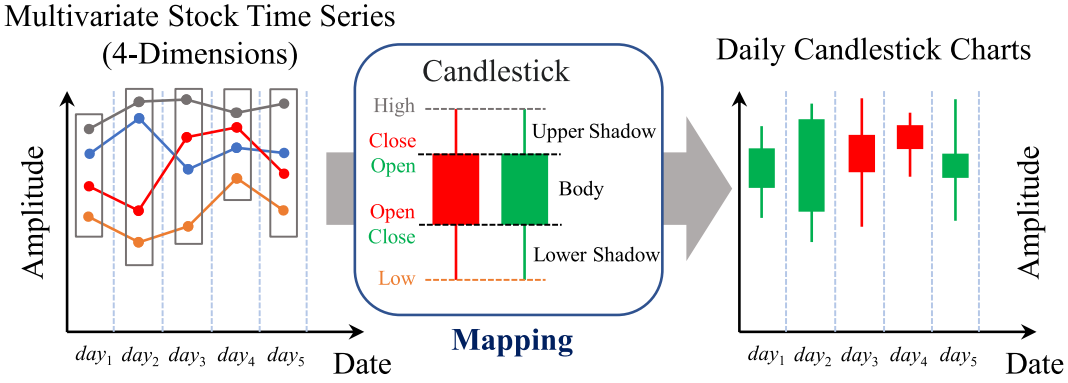


Fig. 2. Through mapping rules, multivariate stock time series are converted into daily candlesticks time series.

and a summary of the similarities and differences between our work and existing works are presented. In Section 3, the methods for encoding candlesticks, the subsequence matching algorithms, and the trend correlation are explored, and the relevant definitions are listed in detail. In Section 4, the calculation of the similarity between the new sequences and the existing patterns and how to use the similarity to predict the trend of target stock is described. Finally, a detailed experimental validation and analysis are presented in Section 5.

2. Related work and gap analysis

The research environment of stock forecasting tasks that integrate candlestick patterns and sequence similarity is multifaceted, then it is divided into three cases: (1) sequential pattern mining and its applications. (2) candlesticks and its applications. and (3) stock time series forecasting. These will help readers quickly understand the significance of our work and the current research environment.

2.1. Sequential pattern mining and its applications

Mining candlestick patterns from candlestick sequences has some similarities to traditional sequential pattern mining, and these similarities have inspired our work. Frequent item mining was presented initially by Agrawal (Agrawal, 1994), which is the prototype of sequential pattern mining. It was proposed to obtain sequences of frequent items to develop patterns for customers' purchasing behaviors in a certain period (Agrawal & Srikant, 1995).

In the traditional sequential pattern mining, $I = \{i_1, i_2, \ldots, i_m\}$ represents a set of items, including m distinct items, and an itemset is a subset of items, besides, $s = <s_1, s_2, \ldots, s_i, \ldots, s_n>$ denotes an ordered list of itemsets, where $s_i \subseteqq I$, s_i is an itemset and an element of sequence s. A sequence $< a_1, a_2, \ldots, a_m >$ is contained in another sequence $< b_1, b_2, \ldots, b_n >$ when there exist integers $1 \leq i_1 < i_2 < \cdots < i_m \leq n$, such that $a_1 \subseteqq b_{i_1}$, $a_2 \subseteqq b_{i_2}$, …, $a_m \subseteqq b_{i_m}$. A support of sequence s is defined as the proportion of all sequences that contain sequence s. For instance, if there are five sequences and three sequences contain sequence s, then the support of sequence s is 3/5. Sequential pattern mining aims to find maximum sequences among all sequences that have certain minimum support. Each maximum sequence represents a sequential pattern.

Concerning the limitations of sequential pattern mining, a broad range of algorithms have been proposed to resolve those limitations, including the AprioriSome, AprioriAll (Agrawal & Srikant, 1995), Generalized Sequential Patterns(GSP) (Srikant & Agrawal, 1996), FreeSpan (Han et al., 2000), PrefixSpan (Pei et al., 2001), Sequential Pattern Discovery using Equivalence classes (SPADE) (Zaki, 2001), etc.

In the original sequential pattern mining, two algorithms (AprioriSome and AprioriAll) were proposed to extract sequential patterns from a database of customer transactions (Agrawal, 1994). Then, it was widely used in predicting a plausible sequence continuation by exploring a rule underlying the generation of a given sequence (Agrawal & Srikant, 1995). Aiming to an accurate destination forecasting given its first partial trip trajectory as input, Iqbal and Pao (2021) considered it a multi-class prediction problem. It proposed an efficient, Non-redundant Contrast Sequence Miner algorithm to distinguish mine patterns, which can only be seen in one class. Owing to recent technological advances, sequential data are generated promptly and frequently to process massive and streaming data. Li, Li, Peng, Li, and Tungom (2020) presented a one-pass algorithm, with a popularly used frequency, as output, from a given sequence and two advanced models to further improve the processing efficiency of the extremely long sequences and streaming data. To alleviate cold-start and data-sparsity problems, Tarus, Niu, and Yousif (2017) introduced a hybrid knowledge-based recommender system using ontology and sequential pattern mining for the recommendation of e-learning to learners. Owing to the disadvantages of traditional sequential pattern mining algorithms, Wang et al. (2021) proposed a timeliness variable threshold and an increment Prefixspan algorithm, namely TVI-Prefixspan, and verified their effectiveness algorithm. Le, Shrestha, Jeong, and Damnjanovic (2021) presented a data-driven scheme, leveraging the readily available daily work report data of past projects to develop a knowledge base for constructing sequence patterns.

The sequential pattern mining algorithms have mainly concentrated on discovering rules, generating sequences, combining resources (Li, Liang, & He, 2017), and predicting sequences. The empirical prediction of financial markets via analyzing historical stock data and interpreting the future trends requires a robust sequential pattern mining algorithm. The investors can access candlestick sequences in historical data and presume the morphology, which may appear at the next stage. A single candlestick could itself stand for a definite trend of a stock. Thus, sequential pattern mining could be used to predict trends of stocks.

Table 1
Abbreviations (Abbr) of data formations with corresponding details.

Abbr	Detailed Descriptions	Abbr	Detailed Descriptions
O	the Opening price	*MI*	Macroeconomic Indicator
H	the Highest price	*MFTS*	Multivariate Financial Time Series
L	the Lowest price	*A*	the trading Amount
C	the Closing price	*IP*	Investor Profiles
V	the trading Volume	*CTFD*	Client Transaction Flow Data
TI	Technical Indicators		

2.2. Candlesticks and its applications

Although candlestick data is commonplace for financial researchers, the combination of computer technology and candlestick data for financial market research has not produced as many results as one might expect. This is a fascinating phenomenon and can be verified simply. For example, a joint search in DBLP (Ley, 2002) (A computer science bibliography website started in 1993 and all important journals/conferences on computer science are tracked.) using "candlestick" as a keyword yielded only 60 matches in the entire database, half of which were recorded in the last five years, with the earliest recorded research published in 2005[1]. A closer comparison reveals that some of these papers have been duplicated for searches, and others have been in preprint, meaning that there may be less persuasive research than expected. We have sorted through the 17 papers published in the last three years based on the search results, summarized in Table 3 below.

In order to present information from multiple papers clearly in one table, we have abbreviated some terms, with detailed descriptions of financial data sources and machine learning models in Appendix A (Table A1 and Table A2), and descriptions of data formations and metrics in Table 1 and Table 2. In particular, "*OHLC*" means that the input data includes the opening(O), highest(H), lowest(L), and closing (C) prices.

The analysis of the 17 papers led us to the following conclusions.

(1) Studies considering candlestick patterns are rare (Chen & Tsai, 2020; Hu, Si, Fong, & Lau, 2019; Naranjo & Santos, 2019; Udagawa, 2019) and most focus on individual candlesticks or inter-day price movement (Lin, Liu, Yang, & Wu, 2021; Udagawa, 2019; Wang & Wang, 2019). In particular, the literature (Chen & Tsai, 2020) points out that the candlestick patterns used in most studies are the summaries of experienced investors and that such artificially derived "pattern-trend" correlations are often ineffective in complex financial markets (Wang & Wang, 2019). The 103 candlestick patterns studied in the literature (Hu et al., 2019) can be considered off-the-shelf expert knowledge, and the

[1] Online result: https://dblp.dagstuhl.de/search?q=candlestick. The search date: December 24, 2021.

Table 2
Abbreviations (Abbr) of metrics with corresponding details (In alphabetical order).

Abbr	Detailed Descriptions	Abbr	Detailed Descriptions
ACC	Accuracy	NMSE	Normalized Mean Square Error
A-ERR	Average Error Percentage	Pro	Periodic Profit
ERR	The Error Percentage	RMSE	Root Mean Square Error
F1	F1 Score	R^2_{os}	Out-of-sample R-square
MAE	Mean Absolute Error	SDBC	Standard Deviation with the Bessel Correction
MAPE	Mean Absolute Percentage Error	TR	Total Relative
MSE	Mean Squared Error		

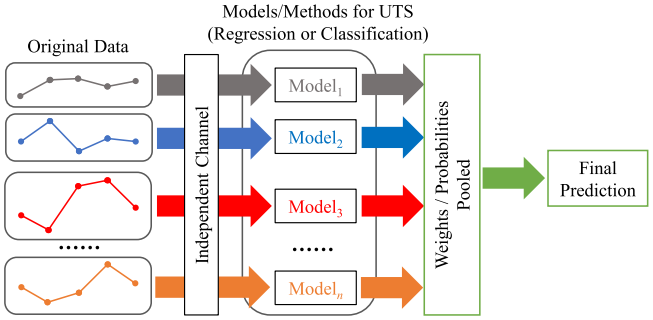


Fig. 3. The candlesticks are broken down into multivariate time series consisting of multiple price data. These multivariate time series are used as original data, with each dimension representing a feature. An independent model learns these feature time series, and the final predictions are obtained by pooling weights/probabilities. Models and methods suitable for dealing with univariate time series (UTS) are widely used.

researcher went through various machine learning methods to verify the accuracy of this expert knowledge. In fact, existing "candlestick patterns" are usually described by natural language or fuzzy rules (Naranjo & Santos, 2019; Naranjo et al., 2018) and are not suitable for direct analysis using computer technology.

(2) Many researchers have incorporated other technical indicators on top of candlestick data to improve model accuracy (Ananthi & Vijayakumar, 2020; Birogul, Temur, & Kose, 2020; Lin et al., 2021; Zhipeng, 2019), but the added technical indicators vary greatly in number and type, resulting in non-comparable findings.

(3) The candlestick data is composed of four price data. However, instead of considering the candlestick as a whole, many studies have treated the candlestick data by splitting it into multivariate time series containing four price time series (Ahmadi et al., 2018; Ananthi & Vijayakumar, 2020; Lin et al., 2021; Zhipeng, 2019) (as shown in Fig. 3), which has essentially lost the unique structural information of the candlestick data.

(4) The size of the data set varies considerably. In some studies, researchers have studied data from over 3,000 stocks (Lin et al., 2021; Zhipeng, 2019), while others may have focused on data from only one sample from start to finish (Hu et al., 2018; Meng et al., 2021; Udagawa, 2018, 2019). At the same time, the data sources used by researchers differ from each other, with most studies being based on specific data sources and data sizes.

(5) Specific candlesticks recur (Chen & Tsai, 2020), candlestick data is a generalization of price data over time, and property enables de-noising (Fengqian & Chao, 2020; Udagawa, 2018).

(6) A novel idea from the perspective of image processing (Birogul et al., 2020; Guo, Hsu, & Hung, 2018; Hu et al., 2018), these studies encode candlestick data as 2D candlestick charts and learn the morphological features of the candlestick data through deep neural networks.

(7) Although all used candlestick data, the ultimate targets of these studies were not the same, some papers focus on predicting market trends or future prices of stocks (Ananthi & Vijayakumar, 2020; Lin et al., 2021; Madbouly et al., 2020), while others are more concerned with designing trading strategies to make profits (Birogul et al., 2020; Fengqian & Chao, 2020; Hu et al., 2018; Naranjo & Santos, 2019; Naranjo et al., 2018; Udagawa, 2019). Differences in the purpose of the research also lead to differences in the metrics. Measuring the merits of a trading strategy is not an easy task, with some articles considering asset profitability as a criterion (Fengqian & Chao, 2020; Naranjo et al., 2018) and others proposing complex custom rules for evaluation (Hu et al., 2018). The diversity of metrics is one of the reasons why these studies are so difficult to compare with each other.

2.3. Stock time series forecasting

Stock time series forecasting is a much broader research objective that has been explored by many researchers using different data and different methods/models. Similarly, we summarized the information from 12 relevant papers, and the results are shown in Table 4, with abbreviations consistent with Table 3.

(1) Most research about stock prediction is divided into two classes: the stock "price" prediction (Alhnaity & Abbod, 2020; Mohanty, Parida, & Khuntia, 2021; Niu et al., 2020; P; Yu & Yan, 2020; Yu, Qin, Chen, & Parmar, 2020; Zhang et al., 2020) and the other is the stock "trend" prediction (Chen, Jiang, Zhang, & Chen, 2021; Huang, Mao, & Deng, 2021; Jiang, Liu, Zhang, & Liu, 2020; Long, Chen, He, Wu, & Ren, 2020; Thakkar, Patel, & Shah, 2021). Stock price prediction is a regression problem, and stock trend prediction is a classification problem. MAE, MSE, RMSE, MAPE, etc., are often used to measure regression performance, and accuracy is used to measure classification performance.

(2) **Difficult and unfair to compare directly.** Researches of stock trend prediction use different data, different data formats, and different variables, making it difficult and unfair to compare with each other directly. So, incremental comparison methods are widely used to verify the effectiveness of their proposed models. The classic models are also transformed to ensure the inputs and outputs are the same as their proposed model, and then comparative experiments can be implemented.

(3) **The size of the experiment samples is small.** In the current research of the stock trend prediction, only fewer samples in a dataset are used, often the indexes of different financial markets (Mohanty et al., 2021; Thakkar et al., 2021; P; Yu & Yan, 2020; Zhang et al., 2020), which could not reflect the different effects of the different models on different samples. It is known that there are more than hundreds of stocks in a market. It is not sufficient to experiment with a single-digit sample.

(4) **Unable to cover large numbers of samples.** There are many hyperparameters in the hybrid models, which could be tuned for a single sample. That makes it difficult to transform for different samples. However, the stock which deserves to buy is chosen from hundreds of stocks in a market. The hybrid models in current research could not cover so many samples without tuning for each sample. So, the stock trend prediction model, which could

Table 3
A summary of the information in 17 related papers.

Cite	**Data Source** *Data Formation*	Size	Target	Model	Metric
(Lin et al., 2021)	**CSM** *TI / OLHC*	3455	The direction of the closing price (Classification)	LR / SVM / KNN RF / GBDT / LSTM	ACC F1
(Ananthi & Vijayakumar, 2020)	**NSE** *TI / OHLC*	9	Market trend (Classification)	KNN	ACC
(Meng et al., 2021)	**S&P 500** *Candlesticks (Monthly)*	1	US Stock Market Returns (Mock trading)	OLS	R^2_{os}
(Birogul et al., 2020)	**BIST** *TI and 2D Candlestick (one-year)*	112	"Buy-Sell" Trend Decision (Mock trading)	YOLO	Pro
(Fengqian & Chao, 2020)	**RB / M / IF 300** *Candlesticks (As points)*	3	Trading system achieve adaptive control (Mock trading)	DRL	ACC Pro
(Madbouly et al., 2020)	**NYSE** *Standard Candlesticks*	13	The Next Day Price (Future Candlestick) (Regression)	CM	MSE
(Hu et al., 2019)	**Synthetic and Real (U.S.)** *103 Candlestick Patterns*	3 (Real)	The formal specifications for 103 candlestick patterns. (Classification)	Bagging / RC / RS RF / PART / ANN SVM / N.J.	ACC
(Naranjo & Santos, 2019)	**NASDAQ 100** **IBEX 35** *Fuzzy Candlesticks*	2	"Bug-Hold" Fuzzy Decision (Mock trading)	KNN	ERR A-ERR SDBC
(Zhipeng, 2019)	**Wind** *OHLC / TI*	3612	Higher accuracy of classified prediction on the characterized candlestick. (Classification)	CCEA SVM	ACC
(Wang & Wang, 2019)	**U.S.** *Standard Candlesticks*	20	The direction of the next day's price. (Classification)	N/A	ACC p-value
(Udagawa, 2019)	**NASDAQ** *Standard Candlesticks*	1	Finding criteria that trigger reversing trade. (Mock trading)	N/A	Gain R-Square
(Chen & Tsai, 2020)	**Geometric Brownian Motion** **EUR/USD 1-minute price** *OHLC / Standard Candlesticks*	2	Identifying the special types of candlestick patterns. (Classification)	GASF CNN	ACC
(Ahmadi et al., 2018)	**Unknown Data Source** *OHLC*	48	Stock performance that is given in the form of buy, sell or no-action signal. (Classification)	SVM HAICG	ACC
(Naranjo et al., 2018)	**NASDAQ 100** **Eurostoxx** *Fuzzy Candlesticks*	19	Fuzzy Output (called Bullish) which represents the prediction of an upward trend. (Mock trading)	FM	Pro
(Udagawa, 2018)	**N 255** *Standard Candlesticks*	1	Stock Price Trend. (Classification)	Blending Rules	Custom
(Hu et al., 2018)	**FTSE 100** *2D Candlestick*	1	Providing low-risk high-return portfolios. (Mock trading)	CAE	Custom
(Guo et al., 2018)	**TAIEX** *2D Candlestick*	6	Price Movement (Up or Down). (Classification)	DCP (CNN + AE)	ACC

cover more samples without parameter adjusting one by one, is needed for investors.

(5) **The mode information of the candlestick is lost.** Despite the poor interpretability and compatibility, many researchers have adopted deep learning models (Alhnaity & Abbod, 2020; Long et al., 2020; Niu et al., 2020). Deep neural networks have the advantage of learning correlations between different time series (shown in Fig. 4), which is a clear improvement, but still ignores the oneness that candlestick data possesses in itself. Candlestick has its mode information which reflects the shape of the curve and the pattern of volatility within a time unit, and if the data in a candlestick were processed respectively, the mode information would be lost.

3. Morphology-based candlestick pre-processing and pattern mining

This section presents the relevant definitions and a framework for mining candlestick patterns from standard candlestick data.

3.1. Overview of relevant definitions

In our work, all candlestick data used conform to the classic Japanese Candlestick Chart (Nison, 2001). This is to clarify the formation of the data, as there are many variants of candlesticks that have different definitions, such as the Heikin-Ashi candlesticks. If there is no special declaration, all candlesticks used in this paper are daily candlesticks.

For the sake of brevity and clarity in the definition and description of the method, in the following, "candlestick" will be

Table 4
A summary of the information in 12 related papers.

Cite	**Data Source** *Data Formation*	Size	Preprocessing	Model	Target	Metric
(Dami & Esterabi, 2021)	**TSE** *OHLCV*	10	AE	AE-LSTM	Stock return(Regression)	MAE TR
(Mohanty et al., 2021)	**YES, SBI, BOI** *OHLC*	5	AE	KELM	Closing Price(Regression)	MAPE
(Yu & Yan, 2020)	**S&P 500, DJIA, N 255, HSI, CSI 300, CNPI** *C*	6	PSR	LSTM	Stock Price(Regression)	ACC
(Jiang et al., 2020)	**S&P 500, DJIA 30, NASDAQ** *OHLC, 8 TIs and 16 MIs*	3	CMDV	Meta-LassoLR	Trend(Classification)	ACC
(Zhang et al., 2020)	**SSCI, SZSE, GEM, HSI., DJIA, S&P 500** *C*	6	CEEMD PCA	LSTM	Closing Price(Regression)	RMSE MAE NMSE
(Niu et al., 2020)	**SSCI, SSFI, DJIA** *MFTS (OHLC + 12 TIs)*	3	TFS C-LSTM	LSTM GRU	Opening Price(Regression)	DA
(Yu et al., 2020)	**1 Stock (Pingtan Development)** *OHLC + 10 TIs*	1	LLE	BPNN	Closing Price(Regression)	RMSE MAE
(Alhnaity & Abbod, 2020)	**FTSE 100, N 225, S&P 500** *C*	3	EEMD	BPNN / RNN SVR / GA-WA	Closing Price(Regression)	MAE MSE RMSE
(Chen et al., 2021)	**CSM** *OHLC + 10 TIs*	6	ICGN Dual-CNN	GC-CNN	Trend (Classification)	ACC
(Thakkar et al., 2021)	**NSE of India** *Historical Dataset + 10 TIs*	4	PCC APCC	VNN	Trend(Classification)	ACC
(Huang et al., 2021)	**EURUSD, SSCI, SZSESMEP, CNPI** *C / OHLCVA*	4	NVG	Hybrid Models(CNN, ResNet and GRU)	Trend(Classification)	ACC
(Long et al., 2020)	**CSC** *IP + CTFD*	3	CNN	Attention-based Bi-LSTM	Trend(Classification)	ACC

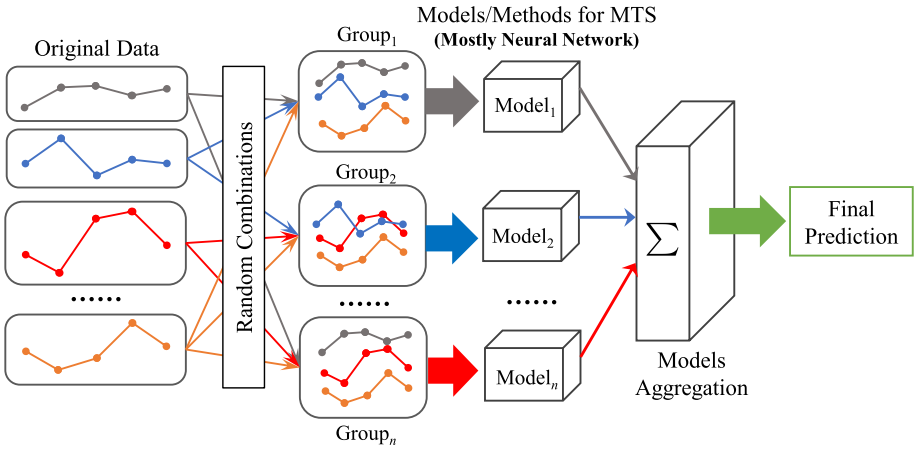


Fig. 4. By performing random combinations of multiple univariate time series, deep neural networks can learn associations between features. In this case, models and methods suitable for multivariate time series (MTS) processing can be applied.

referred simply as "*K*-line", a more commonly used term. Due to a large number of definitions, they are classified, and their abbreviated information is given in different tables. Table 5 lists the data formations used in our work, while Table 6 and Table 7 summarize the features and functions of these data formations.

3.2. Basic definitions of K-line and morphology-based encoding method

The *K-line* data is the basis of our work. During our approach, *K*-line will exist in different formations layered on top of each other. For example, a single *K-line* is a basic unit containing daily stock prices, and *n K-lines* grouped together in chronological order become a *K-line* time series.

Definition 1. (*K-line*, *k*) *A K-line of a stock is defined as a* 5*-tuple*, $k = (date, op, hp, lp, cp)$, *where date is the timestamp of the K-line, for a daily K-line, date is just the date that K-line is recorded, op is the opening price in a day, hp is the highest price in a day, lp is the lowest price in a day, and cp is the closing price in a day. If the opening price is lower than the closing price, it is a "Yang K-line", namely a white candlestick, and if the opening price is higher than the closing price, it is a "Yin K-line" a black candlestick. The color of Yin or Yang K-line is diverse in different countries. In China, Yang K-line is red, and Yin K-line is green. However, an opposite trend is dominant in the United States. A K-line* (*k*) *is a fundamental element of a K-line time series, and its morphology can be viewed in* Fig. 2.

Table 5
Abbreviations and descriptions of concepts (data formations) and the corresponding definitions.

Concept	Abbr	Description	Definition
K-line	*k*	Basic candlestick unit with *date*, *open*, *close*, *high* and *low* prices. The shape is shown in Fig. 2.	Definition 1
K-line Time Series	*KTS*	A sequence of several *K-lines* arranged in *date* order.	Definition 2
K-line Pattern	*KtsP*	A data pair consisting of a specific *KTS* and a corresponding *Trend*.	Definition 3
Subsequence	*Sub*	When each element of a shorter sequence appears in the order in another longer sequence, the shorter one is called a *subsequence* of the longer one.	Definition 4
Pattern Set	*PSet*	A set obtained by integrating multiple *K-line* Patterns (*KtsP*).	Definition 5
Pattern Record	*pr*	Multiple features of a particular *KTS* are mined by self-comparison of *PSet*. These features are combined with the *KTS* and the corresponding *Trend* into one record.	Definition 6
Pattern Record Set	*PRSet*	A set obtained by integrating multiple Pattern Records (*pr*).	Definition 7

Table 6
Descriptions of features and the corresponding definitions.

Feature	Description	Definition
Trend	The direction of price movement for each *KTS* in the next period.	Definition 3 and Section 3.3
oNum	The sum of times that a particular *KTS* appears as a *subsequence* of each *KTS* in a *PSet*.	Definition 6 and Eq. (1)
sameTrendNum	The number of times that the *Trend* is the same based on *oNum*.	Definition 6 and Eq. (2)
Pattern Accuracy (*PACC*)	A statistic describing the magnitude of the correlation between the *KTS* and the corresponding *Trend*.	Definition 6 and Eq. (3)

Table 7
Descriptions of functions and the corresponding definitions.

Function	Description	Definition
whetherSub(*X*, *Y*)	Determine if sequence *X* is a *subsequence* of sequence *Y*.	Function 1 and Algorithm 1
E-KTS(*KtsP*)	Extracts and returns the *KTS* in a specific *KtsP*.	Function 2
E-Trend(*KtsP*)	Extracts and returns the *Trend* in a specific *KtsP*.	Function 3
sameTrend($KtsP_i$, $KtsP_j$)	Determine whether the trends of the two *KtsP*s are the same.	Function 4
whetherExist(*KtsP*, *PRSet*)	Determine whether a *PRSet* contains a *pr* for a particular *KtsP*.	Function 5

Definition 2. (*K-line Time Series*, *KTS*) *A K-line time series KTS*=$\{k_1, k_2, \ldots, k_i, \ldots, k_n\}$ *is a sequence of K-line that is ordered chronologically, where n denotes the length of the K-line time series* (*i.e.*, Fig. 2 *shows a K-line time series with length* $n = 5$ *and* k_i *is the K-line of* day_i). *The K-line time series* (*also called the K-line sequence*) *in this paper is a typical instance of multivariate financial time series, and it is also a part of the K-line pattern that will be defined later.*

The essence of mining *K*-line patterns lies in the analysis of 5-tuple combinations. This is a complex task because there are many variables to consider. On the other hand, as the relationship between the four prices that make up a *K*-line is limited, this leads to limited morphologies for the *K*-line. An encoding method based on the type of morphology of the *K*-lines is proposed, which can downscale a multivariate time series to a single-dimensional morphology series while preserving the temporal order. The detailed encoding rules are shown in Table 8, and the abbreviations are consistent with Definition 1. In particular, it should be noted that there is an extreme case where the four prices are identical (corresponding to a straight line "—"), which is not considered.

Fig. 5 shows the one-to-one correspondence between the *K*-line morphologies and the codes. With the morphology-based encoding method, a one-dimensional representation of the *K*-line time series can be obtained. Fig. 6 shows a simple example of a *K*-line time series with length = 9 encoded to a one-dimensional character sequence with the same length.

Table 8
Morphology-based encoding rules of a *K*-line.

No.	Relationship between four prices	Code	Type
1	$hp > op > cp > lp$	*a*	Yin *K*-lines ($op > cp$)
2	$hp = op > cp > lp$	*b*	
3	$hp = op > cp = lp$	*c*	
4	$hp > op > cp = lp$	*d*	
5	$hp > cp > op > lp$	*e*	Yang *K*-lines ($cp > op$)
6	$hp = cp > op > lp$	*f*	
7	$hp = cp > op = lp$	*g*	
8	$hp > cp > op = lp$	*h*	
9	$hp > op = cp > lp$	*i*	Doji ($op = cp$)
10	$hp = op = cp > lp$	*j*	
11	$hp = op = cp > lp$	*k*	
12	$hp > op = cp = lp$	*l*	

3.3. Segmentation and trend tagging

K-line pattern mining is performed after converting the *K*-line to a string representation. The traditional sequential pattern mining method aims to find frequent itemset and screen sequences by the sequence support, which is utilized to represent the frequency of each sequence. The majority of the extracted sequence patterns in the traditional sequential pattern mining are in the form of a one-dimension sequence

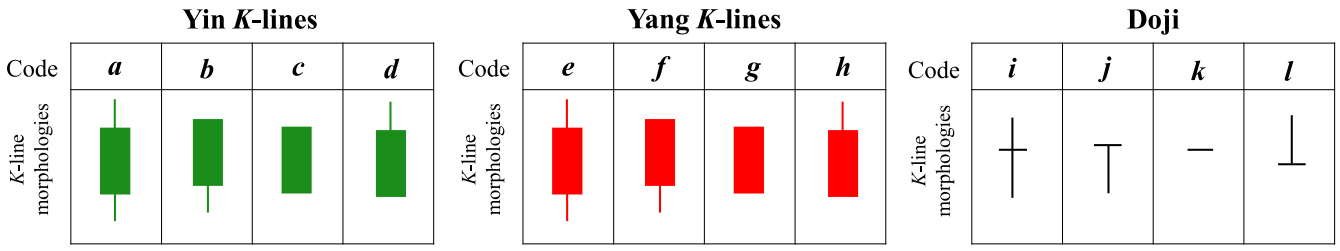


Fig. 5. The one-to-one correspondence between the *K*-line morphologies and the codes.

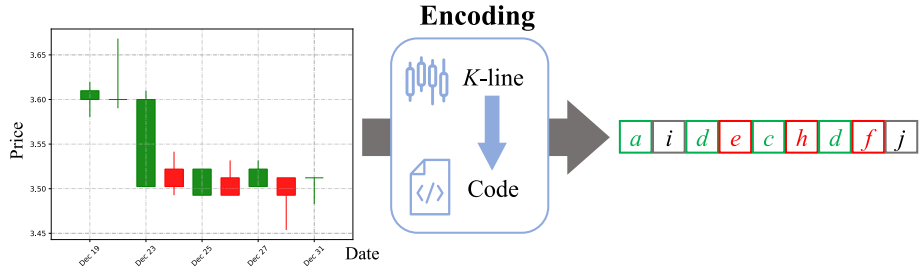


Fig. 6. A simple example of encoding *K*-line time series with length = 9.

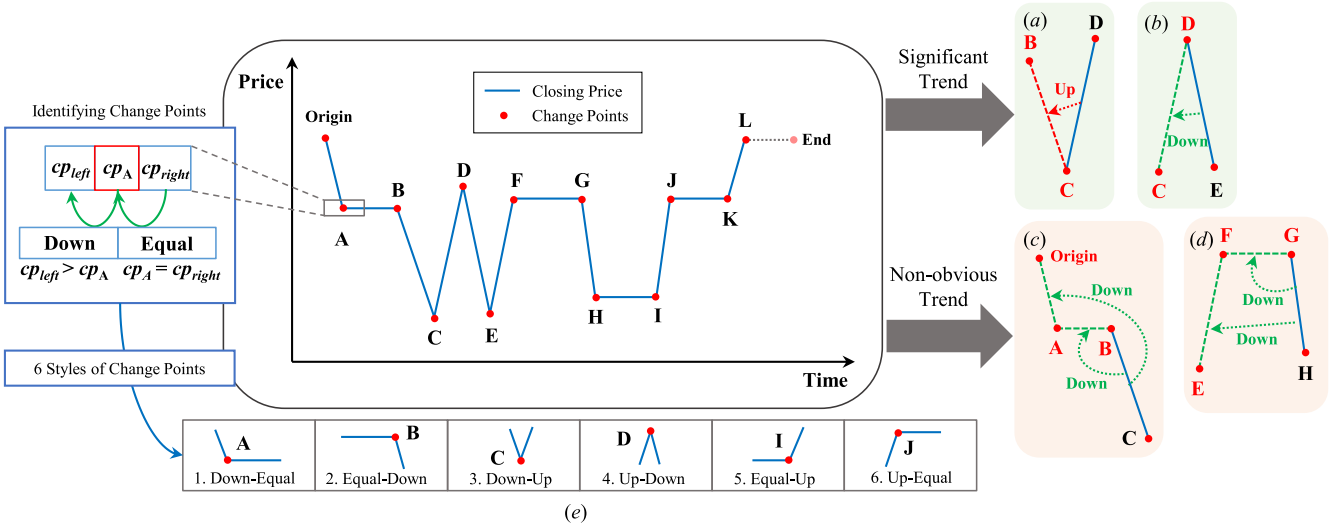


Fig. 7. Illustration of segmentation and trend tagging.

without a tag, and the sequence is screened only by the least sequence support. The *K*-line pattern consists of a *K*-lines sequence and a *Trend* tag in our approach. The relevant definitions are given below. Rather than traversing each combination of elements to obtain the initial pattern, the long sequence is sliced into a number of short sequences in chronological order based on price change points (shown in Fig. 7), with the later sequence providing the basis for trend tagging of the sequence before them. The encoded sequence does not need to be used in this process, and only the closing price sequence will be considered. As the time-stamps are strictly consistent, the segments and trends can be mapped one-to-one onto the encoded sequence.

Definition 3. (*K-line Pattern, KtsP*) *A K-line pattern KtsP={KTS, Trend} is composed of two components, KTS and Trend. KTS is a K-line time series, and Trend is a trending tag, representing the price trend after the K-line time series in history data are harvested. If the price goes up, Trend = "Up", if the price goes down, Trend = "Down", if the price neither goes Up or Down, Trend = "Equal".*

How the price change points can be used to achieve segmentation and trend tagging are illustrated with the example shown in Fig. 7, where the blue line is the closing price curve of a stock, and the red points are price change points. This example combines several artificially constructed sequences and is intended to show as closely as possible the situations encountered during actual data processing. The example contains 12 common price change points (from "A" to "L") and two boundary points ("Origin" and "End").

The sequences should firmly stand for the price trend because it is necessary to achieve the *K*-line pattern with a prediction ability. The price change points in the closing price curves are also called the "price reversal points", indicating that the price may show an upward or downward trend. Price change points are the basis for segmentation, and they can be positioned by a change in price direction. A sliding window with size = 3 is used to determine whether our work's price direction has changed. As shown in Fig. 7, the sliding window contains three closing prices (*cp*) from point "A" and the points to the left and right of it, in chronological order, the direction from the left point to point "A" is

Table 9
Time series segments and their trend tags.

No.	Segment (Begin-End)	Trend
1	Origin-A	Down
2	A-B	Down
3	B-C	Up
4	C-D	Down
5	D-E	Up
6	E-F	Down
7	F-G	Down
8	G-H	Up
9	H-I	Up
10	I-J	Up
11	J-K	Up
12	K-L	Equal
13	L-End	/

"Down" as $cp_{left} > cp_{A}$, while the direction from point "A" to the right point is "Equal" as $cp_{A} = cp_{right}$. As the direction of prices to the left and right of point "A" clearly changes, it can be marked as a price change point. Similarly, there are six styles of price change points, and their shapes and directions are illustrated in Fig. 7 (*e*).

The sequence is split at the price change points to obtain some subsequences of varying length, with the direction in each subsequence being consistent. These subsequences are the prototypes of the *K*-line pattern and need to be labeled with their corresponding future trends further. Specifically, there are two types of trend tagging, as shown below.

- **Significant Trend** (Fig. 7 (*a*) and (*b*)). Any sequence has a significant trend when another sequence immediately follows it with a clear direction of price change ("Up" or "Down"). The significant trend is determined by the price direction of the latter sequence, e.g., the trend of sequence BC corresponds to the direction of the latter sequence CD ("Up"), and the direction of sequence DE marks the trend of sequence CD as "Down".
- **Non-obvious Trend** (Fig. 7 (*c*) and (*d*)). For stock prices, not all trends are significant, and in fact, there are cases where the closing prices of multiple trading days are consistent, which makes the stock price present a period of non-obvious direction ("Equal"). In the face of this situation, the longer term needs to be considered. The sequence trend will be marked by the nearest subsequent "Non-Equal" sequence. For example, the trends of the sequences Origin-A and AB are determined by the sequence BC ("Down"), while the trends of the sequences EF and FG likewise require consideration of the direction of price changes in GH ("Down").

Boundary points are not the same as price change points, and they are only responsible for marking the beginning and end of the data. The

Algorithm 1 Determine whether sequence X is a subsequence of sequence Y.

Input: sequence X, sequence Y
Parameter: m // The length of sequence X
n // The length of sequence Y
Output: **1** or **0** // If 1, X is a subsequence of Y

```
if (m > n) then
    return 0
end if
i, k ← 0
while (i < m) do
    if (k = n) then
        return 0
    end if
    for (j = k to n) do
        if (X[i] = Y[j]) then        // When the same element is found in X and Y.
            i ← i + 1
            k ← j + 1
            break
        end if
        else if (j = n - 1) do
          // when traverses from k^th element of Y sequence and does not find the same element
            return 0
        end else if
    end for
end while
return 1
```

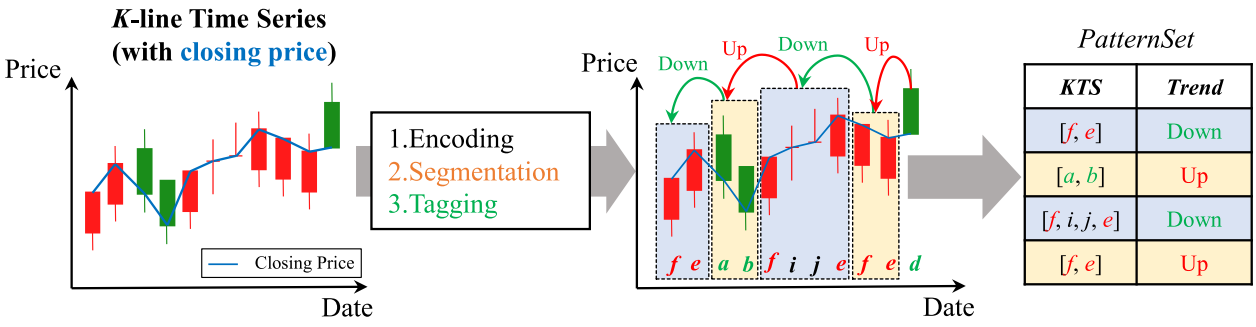


Fig. 8. An example based on real data shows the process of obtaining *K*-line patterns by pre-processing the *K*-line time series.

"Origin" can generally be used as the first price change point, while the sequence between the last price change point (the point "L" in the example) and the "End" are not processed in order to prevent leakage of future information (Although the trend will not be flagged, it still has a direction of a price change that can inform the previous sequence.). Table 9 summarizes all the segments and corresponding trends in the sample shown in Fig. 7. Fig. 8 shows a sample of actual data that combines the pre-processing of encoding, segmentation, and trend tagging.

3.4. K-line Pattern mining

After completing the pre-processing, the prototype of a *K*-line pattern containing the possible *K*-lines sequence that makes up the pattern and their corresponding trends is obtained. These preliminary works are the basis for k-line pattern mining, and the work in this section is to delve into revealing robust correlations between *K*-lines sequences and trends.

The first concern is the definition and judgment of subsequence. Generally, a particular *K*-line pattern takes on various shapes in different financial market environments. Despite the situation's complexity, they result from the same pattern deformation. The subsequence algorithm is proposed to indicate whether a sequence is the deformation of a *K*-line's sequence. The corresponding definition and algorithm are shown below.

Definition 4. (*Subsequence, Sub*) *If every element of an m-length time series* $X = \{x_1, x_2, \ldots, x_i, \ldots, x_m\}$ *shows up in another n-length* ($m \le n$) *time series,* $Y = \{y_1, y_2, \ldots, y_j, \ldots, y_n\}$ *follows the order in X, and the temporal relationship between each pair of elements in X is the same as that in Y, X is the subsequence of Y regardless of the existence of an element that does not exist in X between the pair of elements. For instance, sequence {a, b, c, d} is the subsequence of sequence {**a**, f, **b**, r, e, **c**, j, **d**}. Especially, sequence X is also its subsequence.*

Function 1 *whetherSub*(*X*, *Y*) returns to 1 when sequence *X* is the subsequence of sequence *Y*, or it returns to 0. Algorithm 1 describes the process of subsequence determination.

Definition 5. (*Pattern Set, PSet*) *A Pattern Set* $PSet = \{KtsP_1, KtsP_2, \ldots, KtsP_n\}$ *is composed of n K-line patterns. It is obtained by integrating multiple K-line patterns and represented as a matrix of size n*2, which is the basis for further statistical trend correlations.*

Definition 6. (*Pattern Record, pr*) *A Pattern Record pr = (KTS, Trend, oNum, sameTrendNum, Pattern Accuracy) is defined as a 5-tuple, a supplement of KtsP. For each* $KtsP_i$ *in a PSet* ($length = n$, $1 \le i, j \le n$), *there is a statistical record* $pr_i = (KTS_i, Trend_i, oNum_i, sameTrendNum_i, Pattern\ Accuracy_i)$ *can be obtained by self-comparison, where* KTS_i *and* $Trend_i$ *of* $KtsP_i$ *remain unchanged,* $oNum_i$ *denotes the sum of times that a particular KTS appears as a subsequence of each KtsP in a PSet and defined by Eq. (1),* $sameTrendNum_i$ *is the number of cases where* KTS_i *is a subsequence of* KTS_j *and their Trend are the same direction, a further calculation based on* $oNum_i$ *and defined by Eq. (2). Pattern* $Accuracy_i$ (*abbreviated* $PACC_i$) *calculates the magnitude of the correlation between* KTS_i *and* $Trend_i$ *based on* $oNum_i$ *and* $sameTrendNum_i$ (*described by Eq. (3)*).

$$oNum_i = \sum_{j=1}^{n} whetherSub\big(E\text{-}KTS(KtsP_i),\ E\text{-}KTS\big(KtsP_j\big)\big) \tag{1}$$

$$sameTrendNum_i = \sum_{j=1}^{n} sameTrend\big(KtsP_i,\ KtsP_j\big) \tag{2}$$

$$PACC_i = \frac{sameTrendNum_i}{oNum_i} \tag{3}$$

Definition 7. (*Pattern Record Set, PRSet*) *A Pattern Record Set* $PRSet = \{pr_1, pr_2, \ldots, pr_n\}$ *is a set of n Patter Records. This is the final state of the K-line data in the pattern mining process, which summarises all the information into a matrix of size n*5.*

Function 2 *E-KTS*($KtsP_i$) extracts and returns the KTS_i of the *K*-line pattern $KtsP_i$.

Function 3 *E-Trend*($KtsP_i$) extracts and returns the $Trend_i$ of the *K*-line pattern $KtsP_i$.

Function 4 *sameTrend*($KtsP_i$, $KtsP_j$) determines if the *Trend* is the same for two *KtsP* and it returns to 1 only when *whetherSub*(*E-KTS*($KtsP_i$), *E-KTS*($KtsP_j$)) = 1 and *E-Trend*($KtsP_i$) is the same as *E-Trend* ($KtsP_j$). In the rest of the cases, the return is 0.

Function 5 *whetherExist*(*KtsP*, *PRSet*) returns to 1 when the *K*-line pattern *KtsP* has a pattern recorded in *PRSet*, otherwise, it returns to 0.

K-line pattern mining can be described in four steps based on the above definitions and functions.

- **Step 1**: **Data preparation**. Several *K*-line patterns are obtained by pre-processing (encoding, segmentation, and trend tagging) the raw data, and these *KtsP* are integrated into a *PSet*. Each *KtsP* in *PSet* has a *KTS*, and a corresponding *Trend* tag, namely *PSet*[*i*] (or $KtsP_i$), is

Algorithm 2 *K*-line Pattern Mining.

Input: *PSet*

Parameter: n // The size of *PSet*

Output: *PRSet*

1: **for** (i = 1 to n) **do**		
2: | *sameTrendNum*, *oNum* ← 0 // Parameters zeroing 3: | KTS_1 ← *E-KTS*(*PSet*[*i*]) 4: | $Trend_1$← *E-Trend*(*PSet*[*i*])	Step 1	
5: | **for** (j = 0 to n) **do** 6: | | KTS_2 ← *E-KTS*(*PSet*[*j*]) 7: | | $Trend_2$ ← *E-Trend*(*PSet*[*j*]) 8: | | **if** (*whetherSub*(KTS_1, KTS_2) = 1) **then** 9: | | | *oNum* ← *oNum* + 1 10: | | | **if** ($Trend_1$ = $Trend_2$) **then** 11: | | | | *sameTrendNum* ← *sameTrendNum* + 1 12: | | | **end if** 13: | | **end if** 14: | **end for**	Step 2	Step 4
15: | **if** (*whetherExist* (*PSet*[*i*], *PRSet*) = 0) **then** //When the current *K*-line pattern has not been recorded into the *PRSet*. 16: | | *PACC* ← *oNum*/*sameTrendNum* 17: | | Add *pr*(KTS_1, $Trend_1$, *oNum*, *sameTrendNum*,*PACC*) to *PRSet* 18: | **end if**	Step 3	
19: **end for**		
20: **return** *PRSet*		

taken as a base unit for self-comparison. With a *PSet*[*i*] as the object, zero out all parameters in preparation for the next step.

- **Step 2: Self-comparison.** All *KtsP* in *PSet* is traversed and checked. Including itself, a $KtsP_i$ is to be compared with every *KtsP* in the *PSet* to calculate three essential features (*oNum*, *sameTrendNum* and *PACC*). This process can be regarded as making a copy of *PSet*, and each *KtsP* in *PSet* is compared with the *KtsP* in the copy to determine whether there is a *subsequence* relationship between the two *KTS*s of the two *KtsP*s, and also whether the two *Trend*s of the two are the same, as shown in Fig. 9.
- **Step 3: Check and prevent duplicate operations.** After comparing with the *PSet*[*i*], all data can be integrated as pr_i. *KtsP* with the same *KTS* and *Trend* may appear several times in the *PSet*. The pr_i is searched to verify whether the current $KtsP_i$ has been recorded, if not, the pr_i is added to *PRSet*.
- **Step 4: Loop iterations.** Steps 1–3 are repeated until all *KtsP* in *PSet* can be compared and recorded at *PRSet*.

The whole *K*-line pattern mining process aims to obtain all the pattern's features, assisting in predicting the future trend using the history data. Unlike traditional sequential pattern mining, patterns are not screened during the mining process, and all the patterns are reserved for the prediction model. In our research, *K*-line patterns are screened by predictability.

Algorithm 2 is the detailed description of the process of *K*-line pattern mining. Fig. 9 shows the stages from the *KtsP* to the *PRSet* through a sample example, which uses the same data as Fig. 8 and can be seen as a continuation.

4. A stock trend prediction model based on K-line pattern mining

Although the sequential pattern mining methods and the predictable *K*-line patterns are efficient, how to use these patterns to improve the stock trend prediction is still a challenge. The subsequence method was used to fit the current K-line sequences strictly, and then the trend of the most accurate subsequence pattern was used to predict the trend of the target stock. However, it did not work well in our experiments. Thus, sequence similarity was proposed to achieve the current sequences better matching *K*-line patterns and to predict the stock trend more precisely. Hence, the prediction model is established via a combination of sequence similarity and patterns mined from history data.

4.1. Sequence similarity

When there is a consistency between the current sequence and the sequence of a *K*-line pattern, the standard matching method can find elements the same as the sequence of the *K*-line pattern one by one ("the current sequence" is a sequence at the end of the *K*-line pattern's sequence to be forecast). The matching process is summarized as follows:

- **Step 1**: First, a matching direction is determined (the example shown in Fig. 10 is matched in chronological order). The elements in the current sequence are searched to explore the element's position compatible with the first element in the pattern sequence (*KTS* in *PRSet*), and the current positions of both sequences are taken as the initial point of matching for each sequence. The pointer marks the initial points, as shown in Fig. 10 (1).
- **Step 2**: Compare the elements of two sequences at the position marked by the pointers. If they are the same, both pointers should be

moved forward. Otherwise, only the pointer of the current sequence should be moved forward, and the pointer of the pattern sequence is fixed (Fig. 10 (2)-(7)).

- **Step 3**: Repeat Step 2 until all elements of the pattern sequence can be found identical in the current sequence, or all elements in the current sequence have been compared, but no element compatible with the element in the current position of the pattern sequence can be identified (Fig. 10 (8)).

The example shown in Fig. 10 represents a "compatible" situation where the pattern sequence is a strict subsequence of the current sequence (every element can be matched). Normally, multiple conforming pattern sequences can be found by strict subsequence matching, and these pattern sequences are sorted according to the *PACC* of the *pr* in which they are located. Finally, the pattern sequence with the highest *PACC* can predict the future trend of the current sequence. However, the problems below are worthy of investigation:

(1) **The contradictory matching**. As an example, the current sequence {*b*, *a*, *e*, *e*} can be matched to the different pattern records $pr_1 = \{[a, e, e]$, "Up", …, $PACC = 0.8\}$ and $pr_2 = \{[b, e, e]$, "Down", …, $PACC = 0.8\}$, and both pattern sequences are the strict subsequence of the current sequence with the same *PACC*, while their *Trend* tags are opposite. It is important to indicate which one should be adopted.

(2) **The invalid matching**. An extreme case, meaning that after traversing all *pr* in the *PRSet*, none of the pattern sequences can be identified as a subsequence of the current sequence.

(3) **The similar matching**. Similar matching means that only some elements in the pattern sequence can be matched. This situation is fraught with uncertainty. For instance, the current sequence {*a*, *a*, *e*, *e*, *a*} is similar to pattern records $pr_1 = \{[a, e, e, a, e]$, "Up", …, $PACC = 0.6\}$ and $pr_2 = \{[a, a, e, a, a]$, "Down", …, $PACC = 0.6\}$. The difference in matching direction leads to various matching situations like Fig. 11. Which matching situation yields the most accurate prediction?

To resolve these problems, each element in the sequence is weighed. For a *K*-line sequence $X=\{x_1, x_2, \ldots, x_i, \ldots, x_m\}$ whose length is *m*, the weights of elements set as $w=\{w_1, w_2, \ldots, w_i, \ldots, w_m\}$, and the value of w_i follows the rules presented in Eq. (4).

$$w_i = \begin{cases} \theta, & \theta \in R \text{ and } i = 1, \\ \dfrac{1}{i}, & 1 < i \le \dfrac{m}{2}, \\ \dfrac{1}{(m-i+1)}, & \dfrac{m}{2} < i \le m. \end{cases} \tag{4}$$

The starting point of a *K*-line sequence (x_1) is crucial because it determines the starting timestamp and the location of the one-dimensional sequence. Thus, it is necessary to accurately identify the value of θ to pinpoint the starting point of the sequence and achieve a higher degree of matching. The parameter θ is obtained by training with a large amount of data and described in Section 5.2. For the sake of subsequent calculations, θ is set to 1.1 provisionally. Except for the initial element of a sequence, the closer the element to the current moment is, the more critical the element of the sequence is. The correlation decreases as the time difference increases (Pedrycz & Chen, 2013), and Eq. (4) is formulated according to this rule. In the following, we further define similar sequences.

Definition 9. (*Similar Sequence*). *If some elements of a K-line sequence* $X=\{x_1, x_2, \ldots, x_i, \ldots, x_m\}$ $(m > 1)$ *appear in another sequence* $Y=\{y_1, y_2, \ldots, y_j, \ldots, y_n\}$ $(n > 1)$, *each pair of two elements same both in X and Y follows a strictly consistent order, then X and Y are the similar sequences of each other regardless of the existence of other elements in X. For instance, as displayed in* Fig. 11, *the current sequence* {*a*, *a*, *e*, *e*, *a*} *is a similar sequence of the pattern sequence* {*a*, *e*, *e*, *a*, *e*}. *Especially, a* "*subsequence*" (Definition 4) *is also a* "*similar sequence*", *as the latter is more general.*

For two similar sequences $X = \{x_1, x_2, \ldots, x_i, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_j, \ldots, y_n\}$, the weights of elements are $p = \{p_1, p_2, \ldots, p_i, \ldots, p_m\}$ and $q = \{q_1, q_2, \ldots, q_j, \ldots, q_n\}$, and each pair of matched elements $\{x_i, y_j\}$ have a corresponding weight $\{p_i, q_j\}$, in which the composite similarity of this matched pair is defined as p_i/q_j $(p_i < q_j)$ or q_j/p_i $(p_i > q_j)$. As an element at different positions has different weights, the similarity may differ in diverse matching situations (Fig. 11). Therefore, when calculating the similarity of any two sequences, two different matching directions are considered, including the time increasing (from the past to the future) and time decreasing (from the future to the past). For all pairs of the matched elements of two sequences, the similarity can be obtained as follows:

- **Step 1**: Find and record all the matching situations between *X* and *Y*.
- **Step 2**: For each matching situation, the component similarity of each pair of matched elements is computed, and all the component similarities are added to achieve a similarity of each matching situation, as displayed in Fig. 12.
- **Step 3**: The situation with maximum similarity is selected as the best match, representing the sequence's similarity to the current pattern. Function 6 specifies the relevant calculations.

Function 6 *Similarity*(*X*, *Y*) returns the maximum similarity between sequences *X* and *K*-line pattern *Y*, as defined in Eq. (5).

$$Similarity(X, Y) = \begin{cases} Max(Sim_1, Sim_2, \cdots, Sim_k), & m \ge n \\ \dfrac{Max(Sim_1, Sim_2, \cdots, Sim_k)}{|m-n|+1}, & m < n \end{cases} \tag{5}$$

where [Sim_1, …, Sim_k] is a set of similarities of all matching situations between *X* and *Y*, *Max* () returns the maximum value of the set, *m* is the length of *X*, and *n* is the length of *Y*. If $m < n$, it means the sequence could not include all the elements in the *K*-line pattern, the similarity needs to be decreased to avoid "short sequence" matching with a "long pattern", thus the difference in the sequence length, $|m - n| + 1$, is chosen as the denominator of the similarity to achieve a constraint. Otherwise, the maximum similarity of all the matching situations is chosen between *X* and *Y*.

Direction is a key factor in the matching situation, with some exceptional cases occurring with unidirectional matching. An example is used to illustrate this scenario. When a sequence $X = \{a, d, e, e, e\}$ is matching with a pattern $Y = \{a, e, e, e, e\}$, considering forward or reverse in isolation will give two different unidirectional matching. When both matching directions are considered (bidirectional matching), the order between the directions needs to be discussed, as this determines which of the two identical elements is matched in preference (Fig. 13). A bidirectional matching algorithm is proposed to cover these matching situations, and the matching process is summarized as follows:

- **Step 1**: Traverse *X* and *Y* in chronological order (Forward Matching). If the same elements are found in both sequences, their locations are

Algorithm 3 Bidirectional Matching Algorithm

Input: Sequence X and Pattern Y
Output: Matching results *F-RM* and *R-FM*

```
m ← The length of X
n ← The length of Y
x_start1, x_start2 ← 0 // The location (index) of the start in sequence X
x_end1, x_end2 ← m - 1 // The location (index) of the end in sequence X
y_start1, y_start2 ← 0 // The location (index) of the start in Pattern Y
y_end1, y_end2 ← n - 1 // The location (index) of the end in Pattern Y
F-RM, R-FM ← []
for i = x_start1 to x_end1 do                                          | Step 3
    for j = y_start1 to y_end1 do                                      | Step 1
        if X[i] == Y[j] then
            Add (i, j) to F-RM
            x_start1 ← i + 1
            y_start1 ← j + 1
            break
        end if
    end for
    if (m-1-i) > i and y_end1 > y_start1 then                          | Step 2
        for k = y_end1 to y_start1 do
            if X[m-1-i] == Y[k] then
                Add (m-1-i, k) to F-RM
                x_end1 ← (m-1-i) - 1
                y_end1 ← k - 1
                break
            end if
        end for
    else:
        break
    end if
end for
for i = x_end2 to x_start2 do                                          | Step 6
    for j = y_end2 to y_start2 do                                      | Step 4
        if X[i] == Y[j] then
            Add (i,   j) to R-FM
            x_end2 ← i - 1
            y_end2 ← j - 1
            break
        end if
    end for
    if (m-1-i) < i and y_end2 > y_start2 then                          | Step 5
        for k = y_start2 to y_end2 do
            if X[m-1-i] == Y[k] do
                Add (m-1-i, k) to R-FM
                x_start2 ← (m-1-i) + 1
                y_start2 ← k + 1
                break
            end if
        end for
    else:
        break
    end if
end for
return F-RM, R-FM
```

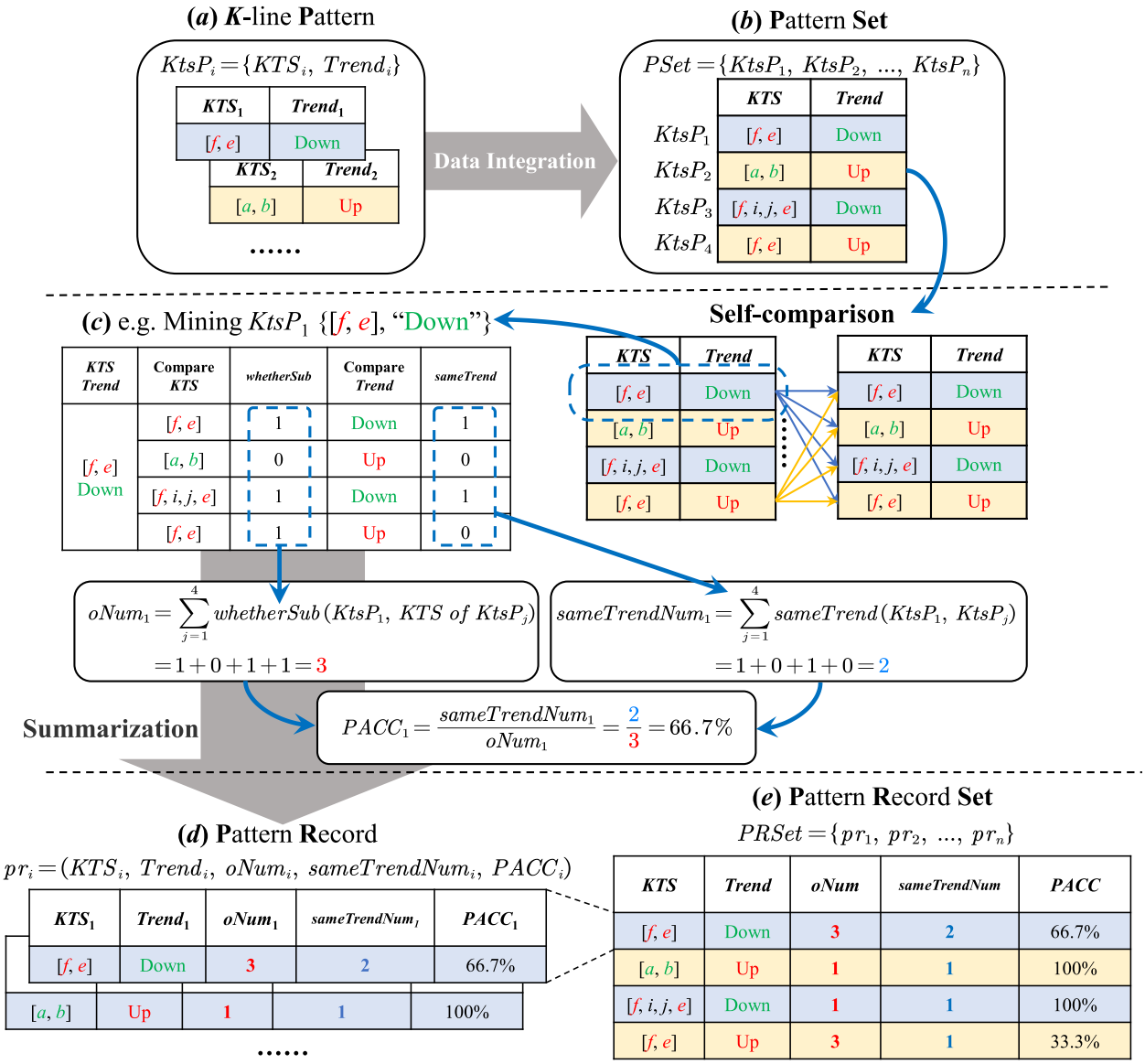


Fig. 9. A complete example contains each stage in *K*-line pattern mining.

recorded, and the next element in each sequence is taken as a new starting point.

- **Step 2**: Traverse *X* and *Y* in reverse chronological order (Reverse Matching). If the same elements are found in both sequences, their locations are recorded, and the next element in each sequence is taken as a new end point.
- **Step 3**: Repeat Step 1 and Step 2 until all elements of *X* and *Y* are checked, and all matched elements are recorded. Up to this point, the bidirectional matching result (abbreviated ***F-RM***) where forward matching is used in preference is obtained.
- **Step 4**: Traverse *X* and *Y* in reverse chronological order (Reverse Matching). If the same elements are found in both sequences, their locations are recorded, and the next element in each sequence is taken as a new end point.
- **Step 5**: Traverse *X* and *Y* in chronological order (Forward Matching). If the same elements are found in both sequences, their locations are recorded, and the next element in each sequence is taken as a new starting point.
- **Step 6**: Repeat Step 4 and Step 5 until all elements of *X* and *Y* are checked, and all matched elements are recorded. Up to this point, the bidirectional matching result (abbreviated ***R-FM***) where reverse matching is used in preference is obtained.

The bidirectional matching algorithm comprises two separate loops, prioritizing forward and reverse matching, respectively, and Fig. 14 illustrates the exact process. A more precise description is given in Algorithm 3.

4.2. *K*-line Pattern based prediction model

Two types of *K*-line pattern matching ("subsequence" and "similar sequence") were described previously in detail and based on these matching methods. Two prediction models were proposed, called subsequence-driven *K*-line prediction model (abbreviated as "Sub-KP") and a similar sequence-driven *K*-line prediction model (abbreviated as "Sim-KP").

In the Sub-KP model, all the *K*-line patterns are mined from the

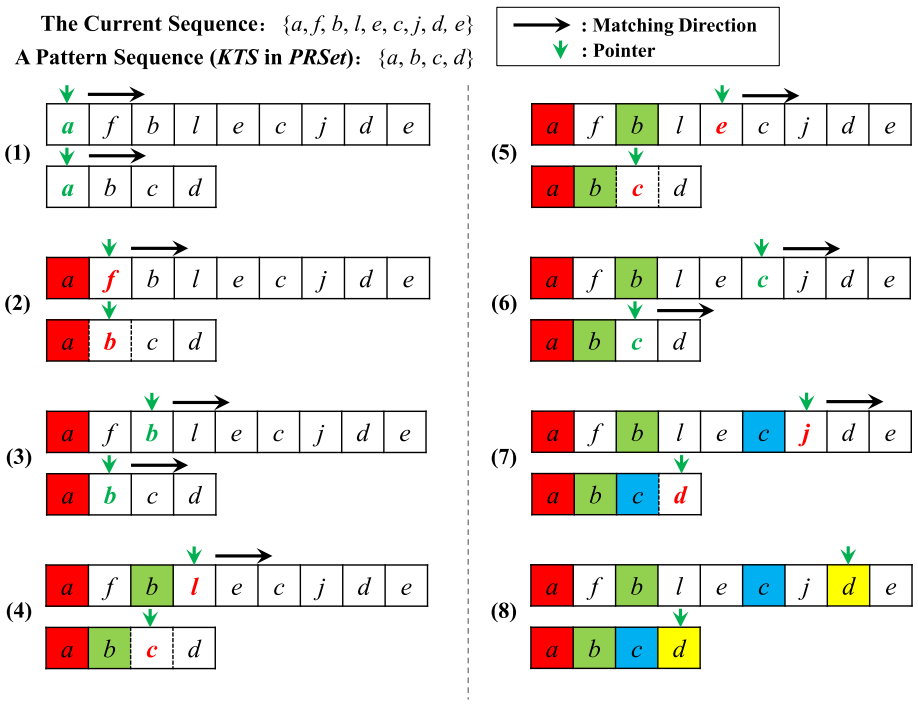


Fig. 10. Illustration of the standard matching.

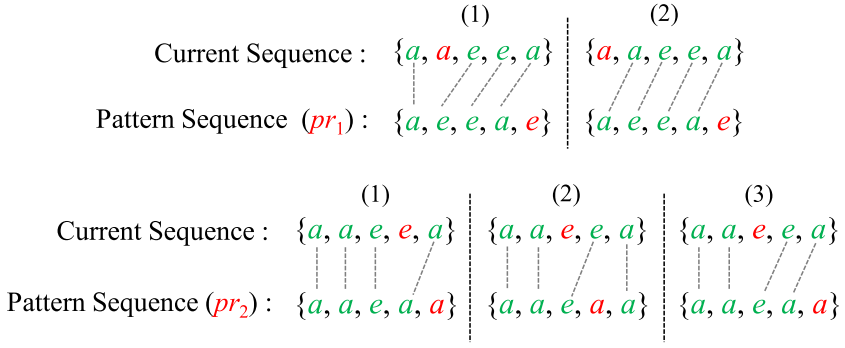


Fig. 11. Different matching situations of the same current sequence and pattern sequence.

historical *K*-line data of the target stock and aggregated into a *PRSet* and ranked according to their *PACC*. Afterward, the *K*-line patterns that are subsequence of the current sequence of the stock to be predicted are filtered. The trend of the *K*-line pattern with the largest *PACC* needs to be selected as the future trend of the target stock in the forecast (when there are multiple *K*-line patterns with the same *PACC*, one of them is chosen at random). The subsequence is tightly constrained, resulting in no *K*-line pattern matching the current sequence, and only random predictions can be made.

The Sim-KP model uses similar sequences as a criterion for *K*-line pattern matching, which results in *PACC* not being the only factor to be considered. Therefore, a scoring mechanism is proposed to select the optimal pattern (Eq. (6)). Fig. 15 depicts the association and differences between these prediction models.

$$MS(sequence, KtsP_i) = Similarity(sequence, E\text{-}KTS(KtsP_i)) \bullet PACC_i \bullet Conf(E\text{-}Trend(KtsP_i)) \quad (6)$$

where *MS*(*sequence*, $KtsP_i$) is the matching score between the current sequence (abbreviated as "*sequence*") and *K*-line pattern ($KtsP_i$), *Similarity*(*sequence*, *E-KTS*($KtsP_i$)) represents the sequence similarity between *sequence* and KTS_i, and $PACC_i$ is the pattern accuracy of the pr_i in *PRSet* (refer to Definition 6 and Eq. (3)). In addition, the confidence coefficient of a particular trend set as *Conf*(*Trend*) are taken into account and *Trend* ∈{"Up", "Down", "Equal"}. This confidence is expressed as the probability of a trend occurring. This is a constraint because it has been noted that the distribution of trends is not uniform. The stock market has inertia, and when a trend of *K*-line pattern is the same as the market movement over the same period, this trend is supposed to have a high degree of credibility.

The whole work is summarized in Fig. 16, where the framework shown in the figure is complete with three components: pre-processing, pattern mining, and trend prediction. The details are shown below:

(1) The target stock's *K*-line time series (history data) is pre-processed (encoding, segmentation, and tagging).
(2) The encoded and tagged sequences are mined to obtain *K*-line patterns (*KtsP* to *PRSet*).

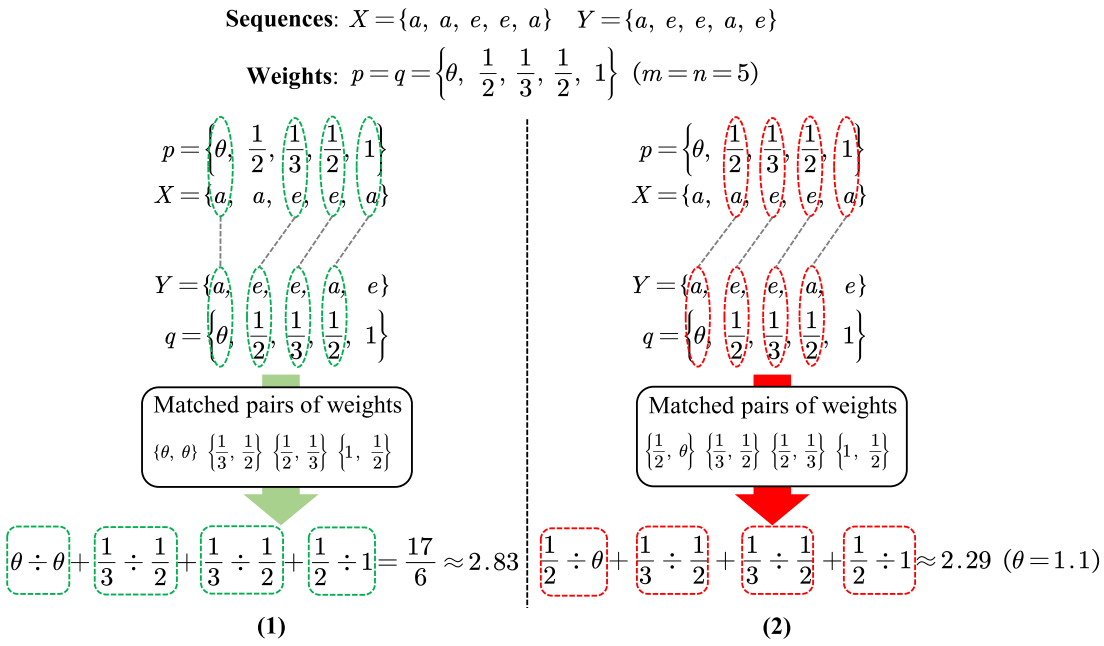


Fig. 12. The calculation of the similarities of different matching situations, the example used is consistent with Fig. 11.

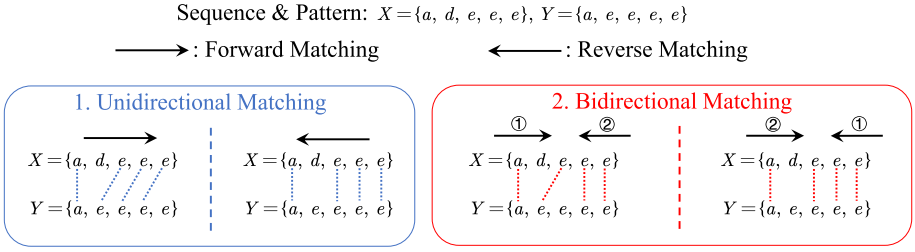


Fig. 13. Examples of Unidirectional and Bidirectional Matching. In bidirectional matching, the ordinal number (① and ②) indicates the order of precedence.

(3) Pattern matching is performed using different algorithms (unidirectional or bidirectional) and criteria (subsequence or similar sequence) to filter out the *K*-line pattern that best matches the current sequence.
(4) The trend of the target stock (the current sequence) is predicted following the "*Trend*" tag of the pattern whose *PACC* is the maximum (the Sub-KP model) or the pattern whose matching score is the maximum (the Sim-KP model). The final prediction depends on the model with the higher accuracy on the test dataset.

5. Experiments and discussion

Careful selection of experimental data is necessary for representative stocks to be considered. Therefore, the constituent stocks of two well-known indices have been focused on. The CSI 300 Index (Code: 000300) is a capitalization-weighted stock market index designed to replicate the performance of the top 300 stocks traded on the Shanghai Stock Exchange and the Shenzhen Stock Exchange, and it is a comprehensive reflection of the overall performance of the Chinese equity market. The CSI 500 Index (Code: 000905) is composed of the top 500 stocks in terms of total market capitalization after removing the constituent stocks of the CSI 300 Index and the top 300 stocks with high market capitalization reflecting the performance of small and mid-cap stocks. The design rules of these two indices ensure that their constituents do not overlap. The validation of our work was carried out based on these 800 (300 + 500) stocks.

5.1. Experimental setup

The list of constituent stocks in the CSI 300 Index and the CSI 500 Index changes in different periods, generally updated every six months and may contain new floatation of shares. To avoid missing data for new stocks, we have selected the list of constituents published on 31 December 2019 to ensure that the data length is consistent across all stocks in the dataset. We collected *K*-line time series data from the publicly available financial data platform (Tushare Pro, https://tushare.pro/) for all stocks in the list between 2019-01-01 and 2021-01-31 for the experiment, where the data for each stock was divided equally into 25 portions, with the first 24 portions serving as training data and the last one as test data.

Further, the experiments are divided into three categories, self-validation, comparative validation, and robustness validation. In self-validation, the effectiveness of the proposed model was verified, and the impacts of different matching algorithms were discussed. In comparative validation, some traditional prediction models were introduced as a comparison. The final validation analyses the robustness of our work. In the following, the experimental design of the first two types of validation is described.

In self-validation, three scenarios were set:

(1) The random prediction according to the statistics of the training dataset (hereinafter referred to as the random model).
(2) The prediction is based on subsequence-driven pattern matching (the Sub-KP model).

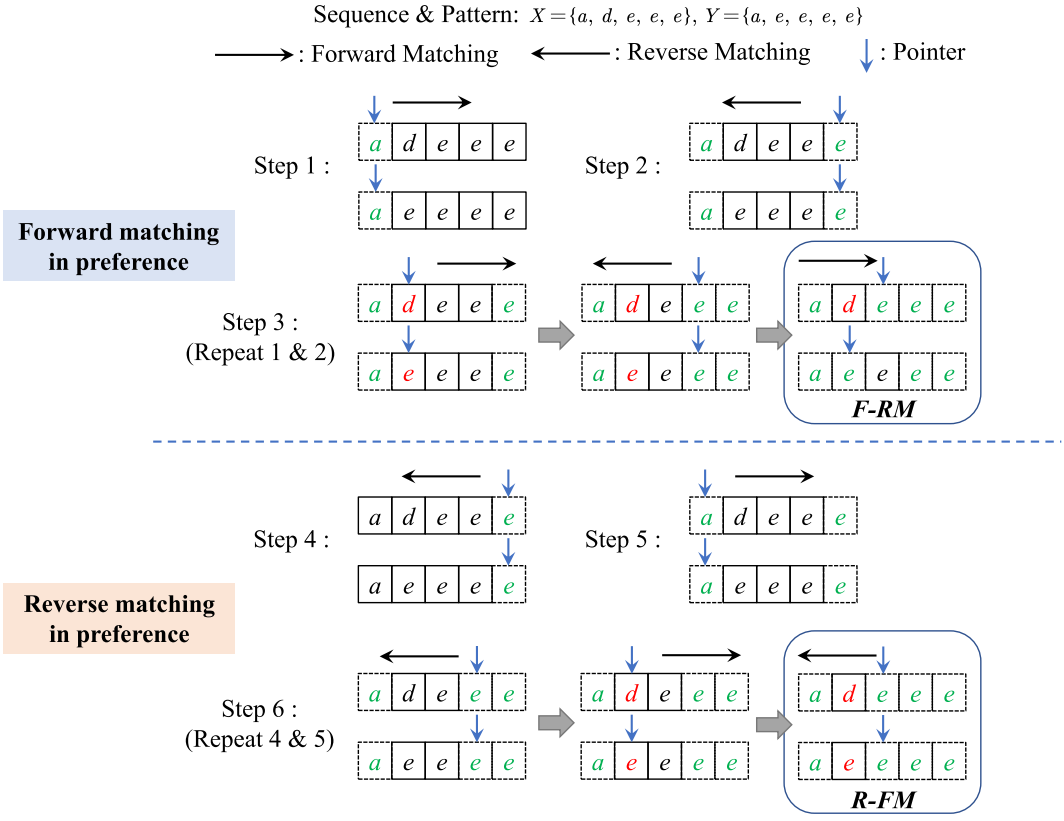


Fig. 14. Illustration of the bidirectional matching.

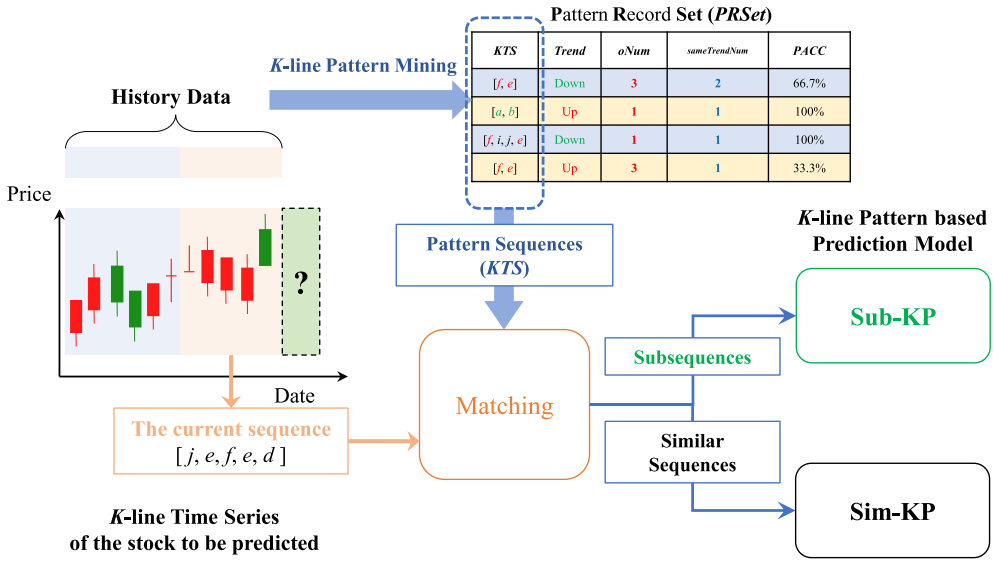


Fig. 15. Two prediction models, where Sub-KP relies on subsequence criteria, and Sim-KP uses similar sequence matching.

(3) The prediction is based on sequence similarity and matching score (the Sim-KP model).

In scenario (1), the probability distributions of the three types of trends ("Up", "Down", "Equal") in the training data were counted, and random trend predictions were made according to the statistical probabilities. This scenario considers that trends in the stock market have continuity and that future price developments have a high probability of following the trends that occur most frequently in the history data.

In scenario (2), 5-day *K*-line data are used to predict the stock trend on the 6th day because, in Chinese financial markets, there are typically five trading days in a week. 5-day *K*-line data are encoded according to their morphology as input data to drive the forecast. A subsequence-driven matching algorithm filters the K-line patterns and ranks the patterns according to *PACC*. The *Trend* tag of the pattern with the highest *PACC* will predict the trend on the 6th day.

In scenario (3), similar sequence-driven matching algorithms and matching score mechanisms will replace subsequence-driven algorithms

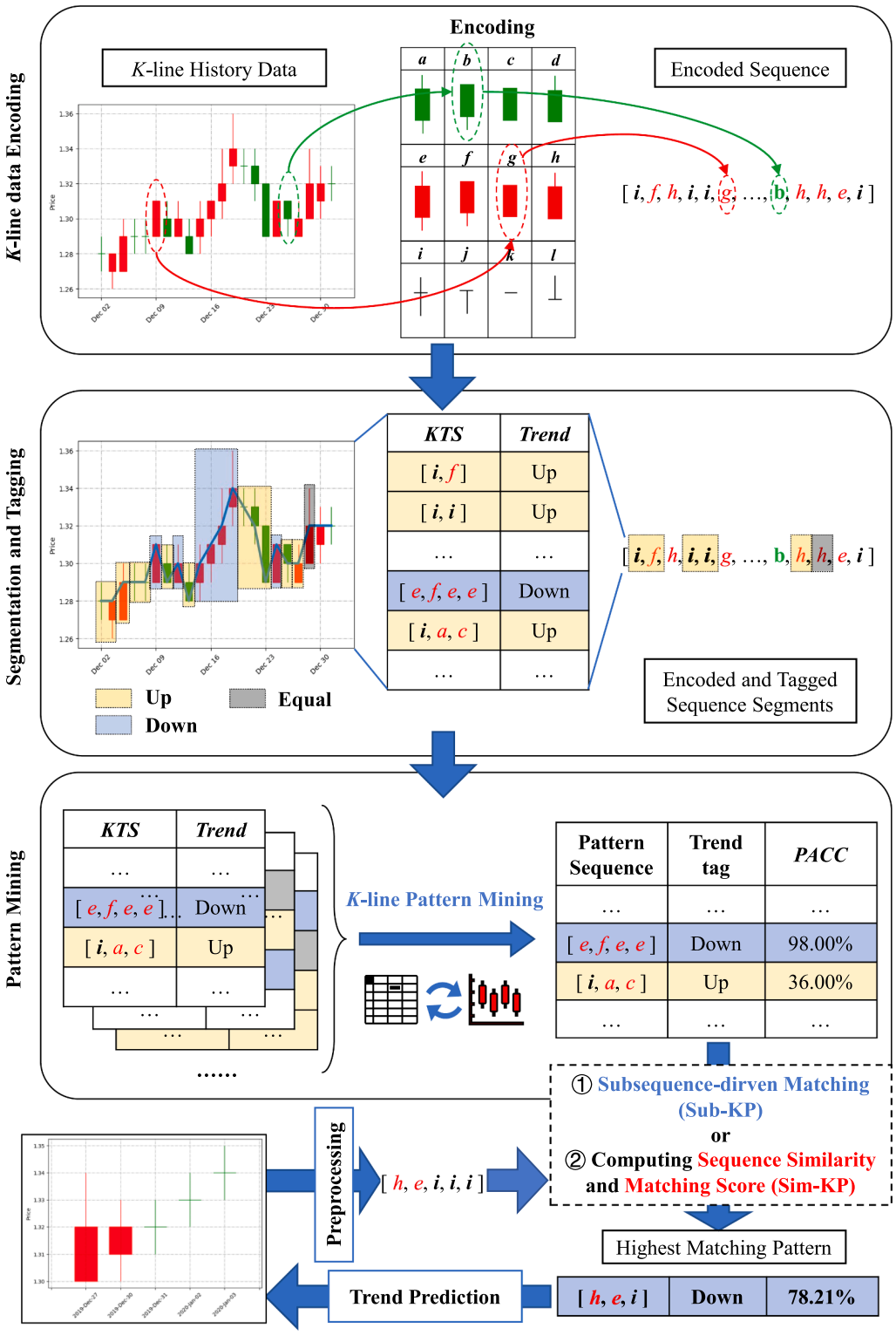


Fig. 16. The complete framework for our work incorporates data pre-processing, pattern mining, pattern matching, and trend prediction.

as the basis for filtering optimal patterns. Similarly, the *Trend* tag of the patterns receiving the highest matching scores will predict the trend on the 6th day.

In comparative validation, two traditional trend prediction models, the SVM and the LSTM, are adjusted and compared with the proposed models, and implemented based on the widely used frameworks named "Pytorch" and "scikit-learn". There are some points about comparative validation that need to be explained:

- **Why are the SVM and the LSTM models chosen to be compared with proposed models, not other hybrid models or the most representative state-of-the-art models?**

As the primary purpose of this work is to propose a pure model to find a fuzzy correlation between particular *K*-line patterns and future trend of a stock, which is part of empirical analysis of stock trend prediction, the input is fixed as multivariate time series, and output is fixed

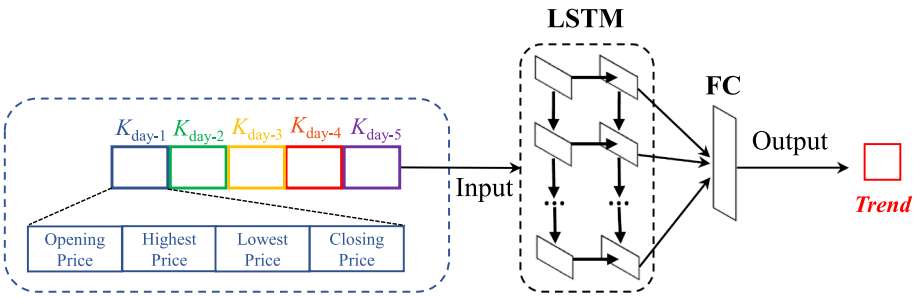


Fig. 17. The configuration of the LSTM in comparative validation.

Table 10
The parameter settings of the SVM and LSTM.

Parameters	LSTM	SVM
Num of Layer	2	/
Num of Full connection layer	1	/
Input Size	20	20
Num of Hidden units	3	/
Output Size	1	1
Kernel Function	/	Radial Basis Function (RBF)

as a tag of stock trend. Most traditional prediction models could not process multivariate time series directly, which need a lot of adjustment and feature tuning, much less the hybrid models, which defeat the primary purpose. The target of the proposed two models is to cover as many samples as possible without the complex tuning. However, the most representative models of the state-of-the-art need to be accurately tuned and ultimately optimized for a single sample, which is not consistent with our target. Furthermore, as widely used models, the performance of the SVM and the LSTM has been shown to be suitable for comparative validation (Halim & Rehan, 2020; Halim, Kalsoom, Bashir, & Abbas, 2016; U et al., 2020).

- **The data preprocessing and the configuration of the comparative models.**

Predictions from both the Sub-KP and the Sim-KP models are required for self-validation and comparative validation, and their experimental setups are kept consistent. There are some substantial differences between the SVM compared to our proposed method. In order to keep that the same training dataset, four price indices of one day are pieced together following the order (the opening price, the highest price, the lowest price, and the closing price). 5-day *K*-line data are integrated with the form of chronological order. Then, the input of SVM is a vector with 20 parameters (4 indices/day * 5 days). The trend on the 6th day is used to tag the training dataset, and if the trend is "Up", the tag is "1″. If the trend is "Down", the tag is "-1″, and if the trend is under correction ("Equal"), the tag of 0 is attainable. On the other hand, the input of LSTM and the tag are the same as the SVM, and the configuration of LSTM is depicted in Fig. 17. However, LSTM was used for regression other than classification. If the output is>0, the *Trend* is "Up", otherwise, the *Trend* is "Down". The Trend is under correction only when the output equals 0 ("Equal"). The number of training epochs is 1,000 for each stock during the training process. The parameter settings of the SVM and LSTM are shown in the Table 10 below.

- **Unification of the outputs and the metrics.**

All models are trained on the same train dataset and strictly tested on the same test dataset. For consistency, the tags and the representations of trends are unified, the trend of upward (tag = "Up") are converted to "1″, the trend of downward (tag = "Down") are converted to "-1″, and trend of correction (tag = "Equal") is converted to "0″.

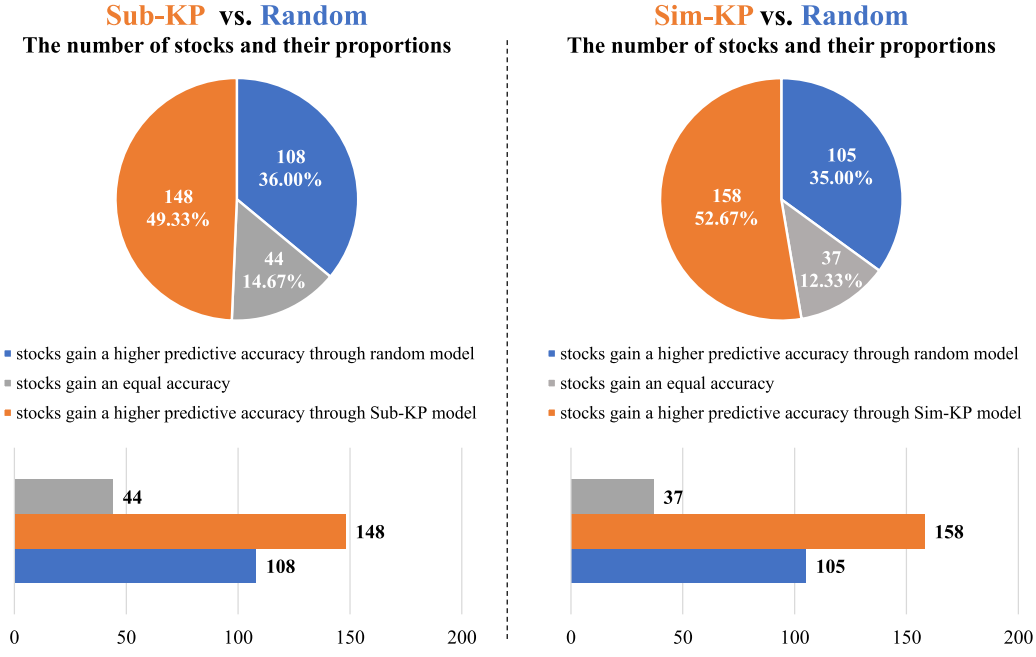


Fig. 18. Comparison results between different proposed models and the random model.

Table 11
Partial illustration of the stocks achieved the best *PA* by Sub-KP model.

No.	Stock Code	Sub-KP	Random	No.	Stock Code	Sub-KP	Random
1	600660	0.8500	0.4000	21	601898	0.7000	0.5000
2	002460	0.7500	0.4500	22	600038	0.7000	0.6500
3	002294	0.7500	0.6000	23	600004	0.6500	0.6000
4	600886	0.7500	0.6000	24	300498	0.6500	0.5000
5	000895	0.7368	0.6316	25	300033	0.6500	0.5000
6	002958	0.7222	0.3889	26	002466	0.6500	0.5500
7	600928	0.7222	0.6111	27	002422	0.6500	0.4000
8	600000	0.7000	0.2500	28	002120	0.6500	0.5500
9	002179	0.7000	0.3500	29	601186	0.6500	0.4000
10	002024	0.7000	0.3500	30	601018	0.6500	0.4000
11	000709	0.7000	0.2500	31	601009	0.6500	0.5000
12	000629	0.7000	0.3000	32	600999	0.6500	0.5500
13	601577	0.7000	0.4500	33	600998	0.6500	0.6000
14	601198	0.7000	0.5500	34	600926	0.6500	0.5000
15	601162	0.7000	0.6000	35	600887	0.6500	0.5500
16	601111	0.7000	0.4500	36	600809	0.6500	0.5500
17	600919	0.7000	0.5500	37	600276	0.6500	0.6000
18	600637	0.7000	0.3500	38	600233	0.6500	0.4500
19	600369	0.7000	0.5000	39	603156	0.6500	0.4500
20	603501	0.7000	0.3500	40	601985	0.6500	0.5000

Table 12
Partial illustration of the stocks achieved the best *PA* by Sim-KP model.

No.	Stock Code	Sim-KP	Random	No.	Stock Code	Sim-KP	Random
1	600919	0.9000	0.5500	21	600566	0.7000	0.5000
2	601878	0.8000	0.5000	22	600547	0.7000	0.6500
3	300017	0.7500	0.5000	23	600436	0.7000	0.6000
4	002466	0.7500	0.5500	24	603259	0.7000	0.4500
5	002352	0.7500	0.7000	25	601877	0.7000	0.6500
6	000709	0.7500	0.2500	26	600100	0.7000	0.5500
7	601800	0.7500	0.5000	27	300124	0.6842	0.4737
8	600999	0.7500	0.5500	28	000627	0.6842	0.4737
9	600660	0.7500	0.4000	29	600928	0.6667	0.6111
10	002945	0.7368	0.4737	30	600015	0.6500	0.2500
11	002958	0.7222	0.3889	31	300144	0.6500	0.4500
12	600010	0.7000	0.3000	32	300070	0.6500	0.4000
13	300015	0.7000	0.4000	33	002714	0.6500	0.5500
14	002120	0.7000	0.5500	34	002555	0.6500	0.5000
15	002024	0.7000	0.3500	35	002202	0.6500	0.5000
16	002008	0.7000	0.5000	36	000629	0.6500	0.3000
17	601319	0.7000	0.5500	37	000568	0.6500	0.4500
18	601288	0.7000	0.2000	38	601198	0.6500	0.5500
19	601162	0.7000	0.6000	39	601117	0.6500	0.4000
20	601111	0.7000	0.4500	40	601012	0.6500	0.4500

It is assumed that the data of $\boldsymbol{N_{test}}$ days are in the test dataset, the actual trend of any given day d_i in the test dataset is $T_{ac}(d_i)$ ($T_{ac}(d_i) \in [-1, 0, 1]$), and the predicted trend of day d_i is $T_{pr}(d_i)$ ($T_{pr}(d_i) \in [-1, 0, 1]$). Function 7 is defined to judge whether a predicted trend is the same as the actual trend, and the predictive accuracy *PA* for each stock on the test dataset is defined by Eq. (7).

Function 7 *predRight* ($T_{ac}(d_i)$, $T_{pr}(d_i)$) returns to 1 when $T_{ac}(d_i) = T_{pr}(d_i)$, or else, it returns to 0.

$$PA = \frac{\sum_{i=1}^{N_{test}} predRight\left(T_{ac}(d_i), T_{pr}(d_i)\right)}{N_{test}} \tag{7}$$

Experiments on data of the constituent stocks of CSI 300 are used to verify the effectiveness of the proposed models. The same experiments are also carried out on the constituent stocks of CSI 500 further to explore the large samples' performance and verify the proposed model's robustness.

5.2. Results and discussion

Three hundred constituent stocks of the CSI 300 Index were experimentally analyzed at first. The predictive accuracy *PA* is chosen as the metric.

5.2.1. Incremental comparison in Self-validation

In self-validation, each proposed model was compared with the random model first. These experiments were divided into two groups: Sub-KP vs. Random and Sim-KP vs. Random. The following is a summary of the results.

In the experiment comparing Sub-KP with the random model, 148 stocks could gain a higher predictive accuracy through the Sub-KP model in scenario (2), and 44 stocks could gain an equal accuracy. Only 108 stocks get better accuracy through scenario (1) random prediction.

In the other group, 158 stocks could gain a higher predictive accuracy by the Sim-KP model in scenario (3), and 37 stocks get an equal accuracy, only 105 stocks could get better prediction through the random model in scenario (1). The comparisons are shown in Fig. 18.

Subsequently, the experimental results of the random, Sub-KP, and Sim-KP models are analyzed together. The Sub-KP model achieves the best predictive accuracy on 148 stocks, and due to space constraints, some of these results are presented in Table 11 (40 of 148). It was indicated that *K*-line pattern mining and Subsequence-drive matching

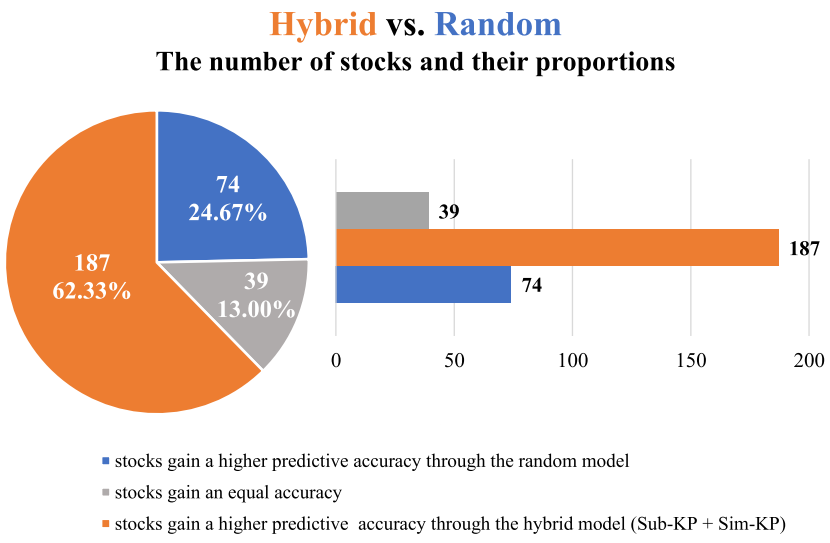


Fig. 19. Comparison results between our work and the random model.

Table 13
Partial illustration of the stocks achieved the best *PA* by the random model.

		Our work						Our work			
No.	Stock code	Sub-KP	Sim-KP	Best *PA*	Random	No.	Stock code	Sub-KP	Sim-KP	Best *PA*	Random
1	000157	0.5500	0.5000	0.5500	0.7500	16	601398	0.4500	0.4500	0.4500	0.6500
2	603019	0.4500	0.4000	0.4500	0.7500	17	601006	0.4500	0.4500	0.4500	0.6500
3	002736	0.6000	0.6000	0.6000	0.7000	18	600674	0.6000	0.6000	0.6000	0.6500
4	002081	0.5500	0.5500	0.5500	0.7000	19	600570	0.5500	0.5000	0.5500	0.6500
5	002032	0.5000	0.5000	0.5000	0.7000	20	600208	0.3500	0.5500	0.5500	0.6500
6	601607	0.4500	0.5000	0.5000	0.7000	21	300413	0.4000	0.4500	0.4500	0.6000
7	601216	0.5500	0.5500	0.5500	0.7000	22	002304	0.5000	0.5500	0.5500	0.6000
8	601155	0.3500	0.4000	0.4000	0.7000	23	002007	0.5000	0.5000	0.5000	0.6000
9	600837	0.5000	0.3000	0.5000	0.7000	24	000768	0.5000	0.5000	0.5000	0.6000
10	600741	0.6500	0.6000	0.6500	0.7000	25	000630	0.4000	0.5500	0.5500	0.6000
11	600352	0.4500	0.6000	0.6000	0.7000	26	000625	0.5500	0.5500	0.5500	0.6000
12	601997	0.3500	0.5500	0.5500	0.7000	27	000423	0.3500	0.5000	0.5000	0.6000
13	600048	0.4500	0.6500	0.6500	0.7000	28	601698	0.4667	0.4000	0.4667	0.6000
14	002456	0.5000	0.5000	0.5000	0.6500	29	601628	0.4500	0.5000	0.5000	0.6000
15	000002	0.4500	0.4500	0.4500	0.6500	30	601336	0.5000	0.4500	0.5000	0.6000

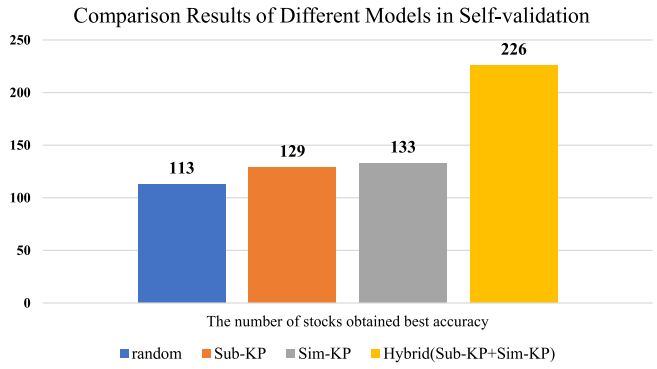


Fig. 20. Comparison between different models.

are effective under some conditions, consistent with our expectations.

The profile of the 300 stocks is complex and varied. The poor performance of the Sub-KP model on some samples drove us to adopt the Sim-KP model to improve accuracy further. Among 300 stocks that were experimentally analyzed, 158 stocks obtained the highest predictive accuracy with the Sim-KP model, and on 125 of these stocks, the Sim-KP model was a significant improvement over the Sub-KP model, some of the comparative results are shown in Table 12 (40 of 125). These results illustrate how considering sequence similarity and performing a matching score calculation can further uncover valid trend support information.

Both the Sim-KP model and the Sub-KP model are part of our work, considering both as a whole, and its predictions on 300 stocks compared to the random model are shown in Fig. 19.

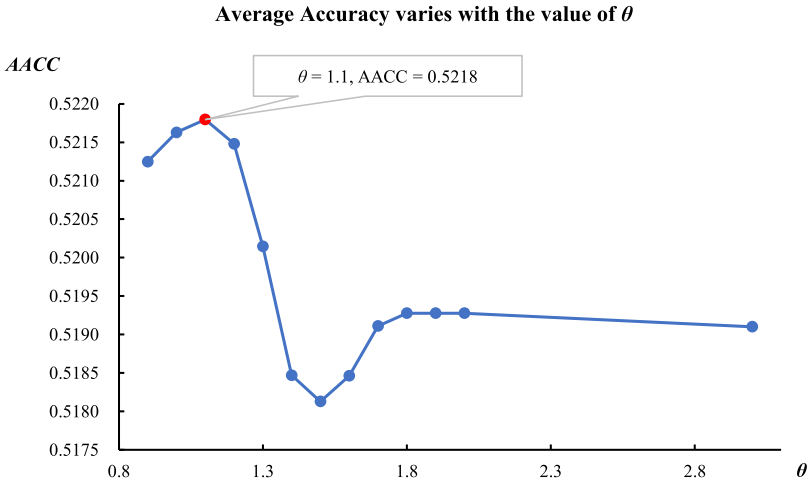
Although our work achieved optimal forecasts on 187 stocks, the random model still bested 74 stocks. Table 13 shows the experimental data for representative 30 stocks of 74, where the best *PA* of both Sub-KP and Sim-KP will be used as the score of our work for comparison with the random model.

All experimental data were aggregated, and when both models achieved the same accuracy on the same stock (grey sectors in Fig. 18 and Fig. 19), this stock was counted as the best *PA* achieved by both for the sake of fairness.

In the 300 constituent stocks of the CSI 300, 113 stocks (37.67% of the whole sample) could obtain the best accuracy through a random prediction model based on statistics, 129 stocks (43% of the whole sample) could obtain the best accuracy through the Sub-KP model, and 133 stocks (44.33% of the whole sample) could obtain the best accuracy through the Sim-KP model, our proposed models achieved the best performance on 226 stocks (75.33% of the whole sample). The comparison results are shown in Fig. 20.

According to these results, the following points need to be analyzed:

(1) **No absolute advantages, only relative ones**. It was found that neither the Sub-KP model nor the Sim-KP model was superior in all samples, which is consistent with the diversity and complexity



θ	*AACC*
0.9	0.5212
1	0.5216
1.1	**0.5218**
1.2	0.5215
1.3	0.5201
1.4	0.5185
1.5	0.5181
1.6	0.5185
1.7	0.5191
1.8	0.5193
1.9	0.5193
2	0.5193
3	0.5191

Fig. 21. Correlation between *AACC* and the value of *θ*.

Table 14
Partial illustration of the experimental results comparative validation.

		Our Work							Our Work				
No.	Code	Sim-KP	Sub-KP	Best *PA*	SVM	LSTM	No.	Code	Sim-KP	Sub-KP	Best *PA*	SVM	LSTM
1	600919	0.7000	**0.9000**	**0.9000**	0.5000	0.6000	21	002179	**0.7000**	0.6000	**0.7000**	0.4000	0.4000
2	600660	**0.8500**	0.7500	**0.8500**	0.5000	0.5500	22	002120	0.6500	**0.7000**	**0.7000**	0.5000	0.5500
3	601878	0.6000	**0.8000**	**0.8000**	0.3500	0.6000	23	002024	**0.7000**	**0.7000**	**0.7000**	**0.7000**	0.4000
4	300017	0.5500	**0.7500**	**0.7500**	0.5500	0.4000	24	002008	0.5500	**0.7000**	**0.7000**	0.4500	0.4000
5	002466	0.6500	**0.7500**	**0.7500**	0.5000	0.5500	25	000629	**0.7000**	0.6500	**0.7000**	0.6500	0.2500
6	002460	**0.7500**	0.4500	**0.7500**	0.6000	0.4000	26	601577	**0.7000**	0.6000	**0.7000**	0.5000	0.5000
7	002352	0.7000	**0.7500**	**0.7500**	0.5000	0.5000	27	601319	0.5000	**0.7000**	**0.7000**	0.6000	0.6500
8	002294	**0.7500**	0.5000	**0.7500**	0.4500	0.5500	28	601288	0.6000	**0.7000**	**0.7000**	0.6000	0.6000
9	000709	0.7000	**0.7500**	**0.7500**	0.5000	0.5000	29	601198	**0.7000**	0.6500	**0.7000**	0.6500	0.5500
10	601800	0.5000	**0.7500**	**0.7500**	0.7500	0.7500	30	601162	**0.7000**	**0.7000**	**0.7000**	0.3000	**0.7000**
11	600999	0.6500	**0.7500**	**0.7500**	0.4500	0.5500	31	601111	**0.7000**	**0.7000**	**0.7000**	0.4500	0.4500
12	600886	**0.7500**	0.6500	**0.7500**	0.5500	0.4000	32	600637	**0.7000**	0.6500	**0.7000**	0.4500	0.6000
13	002945	0.6316	**0.7368**	**0.7368**	0.5789	0.3158	33	600566	0.4500	**0.7000**	**0.7000**	0.4500	**0.7000**
14	000895	**0.7368**	0.6316	**0.7368**	0.5789	0.5789	34	600547	0.5000	**0.7000**	**0.7000**	0.5000	0.5000
15	002958	**0.7222**	**0.7222**	**0.7222**	0.5000	0.5000	35	600436	0.5000	**0.7000**	**0.7000**	0.4500	0.5000
16	600928	**0.7222**	0.6667	**0.7222**	0.5556	0.5556	36	600369	**0.7000**	0.4500	**0.7000**	0.4500	0.6000
17	600029	**0.7000**	0.5500	**0.7000**	0.6000	0.5000	37	600340	0.6500	**0.7000**	**0.7000**	0.6500	0.3500
18	600010	0.5500	**0.7000**	**0.7000**	0.5500	0.5500	38	603501	**0.7000**	0.6500	**0.7000**	0.5500	0.5500
19	600000	**0.7000**	0.5000	**0.7000**	0.5000	0.5000	39	603259	0.5500	**0.7000**	**0.7000**	0.6500	0.6500
20	300015	0.5500	**0.7000**	**0.7000**	0.5500	0.5500	40	601898	**0.7000**	0.5500	**0.7000**	0.4000	0.5500

of financial markets. However, both models proposed in this paper have comparative advantages over the statistical-based random model. The advantages of both models are evident through the comparison results in Fig. 18 and Fig. 19.

(2) **The Sim-KP model compensates to some extent for the shortcomings of the Sub-KP model.** Sub-KP is a forecasting method based on sequence pattern mining and subsequence matching, and its forecasting accuracy is not satisfactory in some samples. The data were analyzed, and it was found that without the introduction of sequence similarity, the prediction results may be influenced by some high frequency and short length patterns that do not represent a clear trend. However, the current sequence may better match lengthy patterns with the sequence similarity, which could stand for a more definite trend and significantly improve prediction accuracy.

(3) **Overall performance.** As different models have different accuracies for different stocks, the accuracy of a single stock is not representative of the average accuracy of each model. For fairness, the **average accuracy *AACC*** of different models **on the same test dataset** is used to compare and represent their effectiveness. ***AACC*** is defined by Eq. (8).

$$AACC = \frac{\sum_{i=1}^{N} PA_i}{N} \tag{8}$$

where N is the number of stocks being experimented with, and PA_i is the predictive accuracy PA of the i^{th} stock defined by Eq. (7). The $AACC$ of the random model is 48.21%, the Sub-KP model and the Sim-KP model reach 51.96% and 51.04% separately.

(4) **The choice of the value of θ.** In the Sim-KP model, the accuracies of different stocks vary with the value of θ in Eq. (4). In order to cater to a larger number of samples and achieve a better $AACC$ for the whole sample, experiments were conducted for different values of θ. The $AACC$ for the whole sample varies with the value of θ, as shown in Fig. 21. The first element of the sequence is crucial, it marks the beginning of the sequence, and therefore, in theory, it should have a greater weight than any other element at any other position. To give a complete picture of how the $AACC$ varies with the value of θ, the value of θ was measured from 0.9 to 3. It is easy to see that the average accuracy tends to rise as the value of θ increases, peaking at around $\theta = 1.1$ and then falling. Finally, the θ value of 1.1 was chosen for our experiment.

(5) **The working principle of the Sub-KP and the Sim-KP models.** The prediction model based on sequential pattern mining and sequence similarity could improve the accuracy of stock trend prediction, as the patterns that lead to a certain trend may be cyclical. Conversely, such patterns may not appear in absolutely identical sequences when various factors are present in the market. *K*-line patterns may be reproduced in similar sequences, but the overall future trend of the *K*-line pattern may remain the same as it was historically.

Experiments in self-validation show that the models proposed in this paper can accurately mine *K*-line patterns and that *K*-line patterns can effectively improve the accuracy of some stock trend predictions.

5.2.2. *Horizontal contrast in comparative validation*

For the comparative validation, two models were introduced, SVM and LSTM, and for consistency, 300 constituents of the CSI 300 were used as experimental data. The configurations and experimental results of the Sub-KP and Sim-KP models are consistent with those in the self-validation, and the configurations of the SVM and LSTM have been described in Section 5.1.

Combining all the experimental results and considering the case where multiple models would achieve the same result on the same stock,

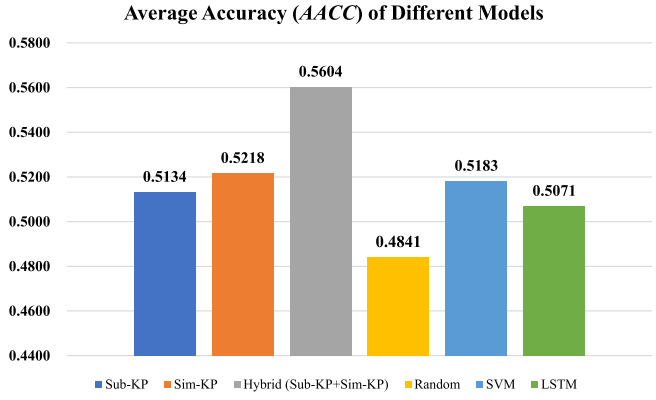


Fig. 22. The *AACC* was used as a criterion to count the performance of all models.

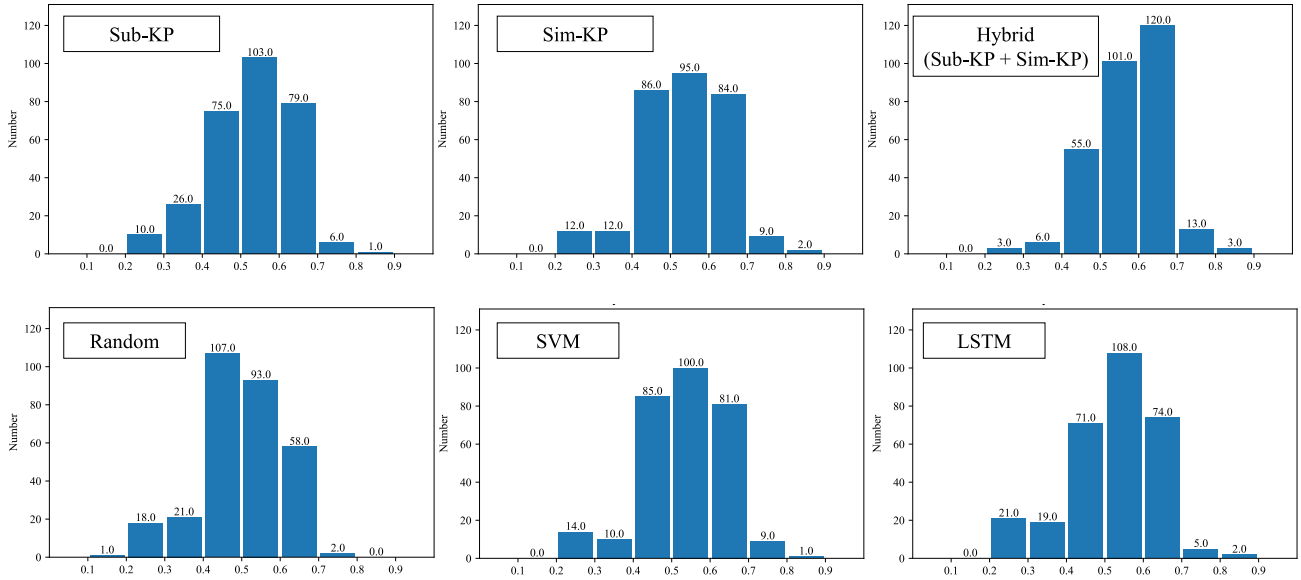


Fig. 23. Predictive accuracy (*PA*) distribution of different models.

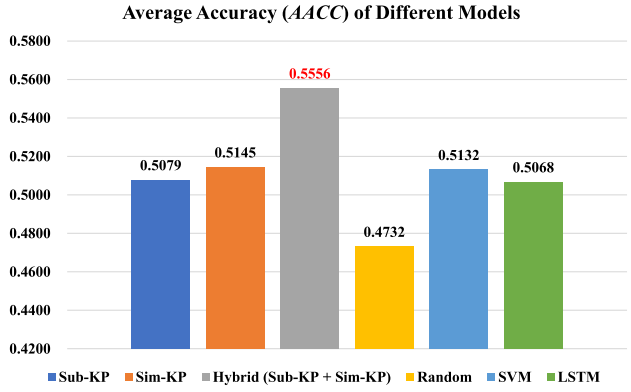


Fig. 24. *AACC* of different models experimented on the data of the CSI 500.

the Sub-KP model achieved the best prediction on 100 stocks, Sim-KP scored 103 stocks, and SVM and LSTM performed best on 109 and 102 stocks, respectively. Our work is composed of Sub-KP and Sim-KP, so our work together achieves optimal forecasts on 178 stocks (59.33% of the total experimental sample). Table 14 shows some of the experimental data to demonstrate the strengths of our work (40 of 178 stocks).

The running time is counted for all models in the same experimental environment, configured with OS: Ubuntu 16.04, CPU: Intel Xeon(R) E5-2698 v4 2.20Ghz, Memory: 256 GB, GPU: Nvidia Tesla V100. For each stock, the entire training and testing process takes about 4.17 s with the model proposed in this paper, 0.57 s with the SVM model, about 1 min and 29.56 s with the LSTM model. Consequently, the proposed model was accompanied by a higher accuracy, fewer computing sources, and a shorter data processing period than the LSTM.

The predictive accuracy achieved by the Sub-KP and Sim-KP models was not satisfactory for some stocks, then a detailed analysis of the process of pattern matching and trend forecasting was carried out to explain the poor results. As a result of our analysis, it is realized that the uncertainty of financial markets is the primary cause. Our work is based on the belief that 'patterns' uncovered from historical data correlate with 'trends', and that when the pattern matches are accurate, forecasts can be made based on the trend information associated with them. In some samples, the proposed model found the best matching pattern accurately, and the trend associated with the pattern matched the statistical results, but the true trend in the future did not match any known knowledge. This means that the future to be predicted is a new pattern that has never existed before, and it may be that unexpected events or policy changes have led to the creation of a new pattern.

Although the proposed model could not perform best on all samples through experiments in comparative validation, a higher average accuracy could be achieved, offsetting the mediocre performance of SVM and LSTM in some samples.

Fig. 22 shows the average performance of the different models for all samples. The random model achieved the lowest accuracy of 48.81%, the Sub-KP and Sim-KP models obtained 51.34% and 52.18%, while the two comparison models, SVM and LSTM, had *AACC* of 51.83% and 50.71%, respectively. The combined performance of our work was 56.04%, the best performance. In order to explore the accuracy distribution of samples obtained by different models, the frequency distribution histogram of accuracy with different models is drawn and shown in Fig. 23.

The distribution chart clearly shows that our proposed model effectively increases the number of stocks with a predictive accuracy between 0.5 and 0.9, while the number of stocks in the low accuracy range is significantly lower than in the other comparable models.

5.2.3. *Robustness validation*

To verify the performance on a larger dataset, the experiments are rerun with the same configuration on the 500 constituents of the CSI 500, still using the average accuracy *AACC* (defined by Eq. (8)) as the evaluation metric, and the results are shown in Fig. 24.

There is no overlap between the constituents of the CSI 300 and CSI 500, with the CSI 500 representing a new environment. The experimental results show that the proposed model has good robustness and maintains its strengths after changes in the size and characteristics of the samples.

5.2.4. *Comprehensive discussion*

After incremental comparison and horizontal contrast, it is found that two proposed models, the Sub-KP model, and the Sim-KP model, can cover different stocks in the market, and the hybrid model could gain significant improvements to cover as many stocks as possible. The hybrid model's principle is based on the regular that similar volatilities may recur due to some periodic change, which is also the basis of the empirical analysis of the financial market.

Although the complex financial market is affected by various factors and shows up as a non-linear status, the hybrid model could eliminate the variation caused by the influence of complex factors to find the hidden rules effectively. That is why the proposed model could work better than the traditional prediction model on the overall performance.

However, if a stock does not follow the regular basis, the current fluctuation has not appeared before, and the proposed model would not work well. That is the shortcoming of traditional sequence pattern mining methods because all patterns are mined from history and cannot create new patterns that do not exist in history.

As the trend predicted by the proposed model is only correlated to a stock's history, with an improved sequence pattern mining method, effective sequence similarity, and sequence matching method, the proposed model could show good robustness.

6. Conclusions

This paper presents a multivariate financial time series stock forecasting model based on sequential pattern mining and sequence similarity. The model improves the accuracy of stock trend forecasting by applying *K*-line pattern mining to multivariate time series data. The traditional sequential pattern mining is improved based on the morphological characteristics of the *K*-line and combined with empirical data analysis to improve the poor performance of the classical forecasting model on certain stocks. The sequence similarity proposed in this work is compatible with financial market volatility under the influence of various factors and can effectively locate similar volatility. The model's validity was verified through experiments and analysis of financial time series data of the constituents of the CSI 300 and the CSI 500 indices. The proposed hybrid model could gain 56.05% and 55.56% average accuracy on two datasets, while the SVM model was 51.83%, 51.32%, and the LSTM model was 50.71%, 50.68%. It has gained significant improvements in overall performance.

Experiment results show that the proposed hybrid model has better effects than traditional prediction models and is robust on different datasets. There are no complex hyperparameters, no complex inputs, and it is friendly for people majoring in different subjects. As real stock data verify the model, and the effectiveness is only decided by the correlation between a stock's history data and future trends, it could be deployed to resolve real business problems. It can be used in large-scale stock screening, timing detection of buy-in or sold-out stocks, portfolio adjustment, etc. The proposed model broadens the application of sequential pattern mining in the financial field and has nice interpretability.

Although the hybrid model still could not cover all the stocks in the market, it still verifies that there are hidden rules between the history data and the future trend of a stock, which researchers could further mine. The future research direction is how to improve the existing mining methods to apply for the financial time series mining and how to design effective similarity to find the association between financial time series.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported by the Science and Technology Planning Project of Shenzhen Municipality [Grant Number JCYJ20190806112210067] and the National Natural Science Foundation of China [Grant Number 61872113].

Appendix

Table A1
Abbreviations(Abbr) of financial data sources with corresponding details (In alphabetical order).

Abbr	Detailed Descriptions	Abbr	Detailed Descriptions
BIST	Borsa İstanbul	**N 255**	Nikkei 255 Index
BOI	Bank of India	**NASDAQ**	National Association of Securities Dealers Automated Quotations
CNPI	ChiNext Price Index	**NSE**	National Stock Exchange
CSC	Chinese Securities Company	**NYSE**	New York Stock Exchange
CSI 300	The China Securities Index 300	**RB**	Rebar Contract
CSM	Chinese Stock Market	**S&P 500**	Standard & Poor's 500 Index
DJIA	Dow Jones Industrial Average	**SBI**	State bank Of India
Eurostoxx	Eurostoxx portfolios	**SSCI**	Shanghai Securities Composite Index
EURUSD	the exchange rate of Euro to US dollar	**SSFI**	Shanghai Securities Fund Index
FTSE 100	Financial Times Stock Exchange 100 Index	**SZSESMEP**	Shenzhen Stock Exchange Small&Medium Enterprises Price Index
GEM	Growth Enterprise Market	**SZSE**	Shenzhen Stock Exchange Component Index
HSI	Hang Seng Index	**TAIEX**	Taiwan Stock Exchange Capitalization
IBEX 35	Spanish Ibex35 markets	**TSE**	Tehran Stock Exchange
IF 300	Index Futures of CSI 300	**YES**	Yes bank
M	Meal Contract		

Table A2
Abbreviations(Abbr) of machine learning approaches/models with corresponding details (In alphabetical order).

Abbr	Detailed Descriptions	Abbr	Detailed Descriptions
AE	Auto Encoder	**KNN**	K-Nearest Neighbor
ANN	Artificial Neural Network	**LASSO**	Least Absolute Shrinkage and Selection Operator
APCC	Absolute Pearson Correlation Coefficient	**LLE**	Local Linear Embedding dimensional reduction algorithm
BPNN	Back Propagation Neural Network	**LSTM**	Long Short Term Memory
CAE	Convolutional AutoEncoder	**LR**	Logistic Regression
CCEA	Cooperative CoEvolution Algorithms	**Meta**	Meta-classifier
CEEMD	Complementary Ensemble Empirical Mode Decomposition	**NVG**	Natural Visibility Graph
CM	Cloud Model	**N.J.**	MATLAB library developed by Nate Jensen
CMDV	Correlation Matrix of Different Variables	**OLS**	Out-of-sample Predictive Regression
CNN	Convolutional Neural Network	**PCA**	Principal Component Analysis
DCP	Deep Candlestick Predictor	**PCC**	Pearson Correlation Coefficient
DRL	Deep Reinforcement Learning	**PSR**	Phase-Space Reconstruction
DTW	Dynamic Time Warping	**RC**	Random Committee
EEDM	Ensemble Empirical Mode Decomposition	**ResNet**	Residual Network
FM	Fuzzy Modeling	**RF**	Random Forest
GA	Genetic Algorithm	**RNN**	Recurrent Neural Networks

(*continued on next page*)

Table A2 (*continued*)

Abbr	Detailed Descriptions	Abbr	Detailed Descriptions
GASF	Gramian Angular Summation Field	**RS**	Random Subspace
GBDT	Gradient Boosting Decision Tree	**SVM**	Support Vector Machine
GC-CNN	Graph Convolutional Feature based Convolutional Neural Network	**SVR**	Support Vector Regression
GRU	Gated Recurrent Unit	**TFS**	Two-stage Feature Selection model
HAICG	Heuristic Algorithms of Imperialist Competition and Genetic	**VNN**	Vanilla Neural Network
ICGN	Improved Graph Convolutional Network	**WA**	Weighted Average
KELM	Kernel Extreme Learning Machine	**YOLO**	Real-time Object Detection System

References

Agrawal, R. (1994). Fast Algorithms for Mining Association Rules. *The Proc. of 20th Int. Conf. on Very Large Databases (VLDB), Santiago de Chile, Chile*.

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. *Proceedings - International Conference on Data Engineering, 3–14*. https://doi.org/10.1109/icde.1995.380415

Ahmadi, E., Jasemi, M., Monplaisir, L., Nabavi, M. A., Mahmoodi, A., & Amini Jam, P. (2018). New efficient hybrid candlestick technical analysis model for stock market timing on the basis of the Support Vector Machine and Heuristic Algorithms of Imperialist Competition and Genetic. *Expert Systems with Applications, 94*, 21–31. https://doi.org/10.1016/j.eswa.2017.10.023

Alhnaity, B., & Abbod, M. (2020). A new hybrid financial time series prediction model. *Engineering Applications of Artificial Intelligence, 95*(August), Article 103873. https://doi.org/10.1016/j.engappai.2020.103873

Ananthi, M., & Vijayakumar, K. (2020). Stock market analysis using candlestick regression and market trend prediction (CKRM). *Journal of Ambient Intelligence and Humanized Computing, 12*(5), 4819–4826. https://doi.org/10.1007/s12652-020-01892-5

Birogul, S., Temur, G., & Kose, U. (2020). YOLO Object Recognition Algorithm and "Buy-Sell Decision" Model over 2D Candlestick Charts. *IEEE Access, 8*, 91894–91915. https://doi.org/10.1109/ACCESS.2020.2994282

Chen, W., Jiang, M., Zhang, W. G., & Chen, Z. (2021). A novel graph convolutional feature based convolutional neural network for stock trend prediction. *Information Sciences, 556*, 67–94. https://doi.org/10.1016/j.ins.2020.12.068

Chen, J.-H., & Tsai, Y.-C. (2020). Encoding candlesticks as images for pattern classification using convolutional neural networks. *Financial Innovation, 6*. https://doi.org/10.1186/s40854-020-00187-0

Cheng, D., Yang, F., Xiang, S., & Liu, J. (2022). Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition, 121*, Article 108218. https://doi.org/10.1016/j.patcog.2021.108218

Dami, S., & Esterabi, M. (2021). Predicting stock returns of Tehran exchange using LSTM neural network and feature engineering technique. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-021-10778-3

Fengqian, D.i., & Chao, L. (2020). An adaptive financial trading system using deep reinforcement learning with candlestick decomposing features. *IEEE Access, 8*, 63666–63678. https://doi.org/10.1109/ACCESS.2020.2982662

Guo, S. J., Hsu, F. C., & Hung, C. C. (2018). Deep Candlestick Predictor: A Framework toward Forecasting the Price Movement from Candlestick Charts. In *Proceedings - International Symposium on Parallel Architectures, Algorithms and Programming*. https://doi.org/10.1109/PAAP.2018.00044

Halim, Z., Kalsoom, R., Bashir, S., & Abbas, G. (2016). Artificial intelligence techniques for driving safety and vehicle crash prediction. *Artificial Intelligence Review, 46*(3), 351–387. https://doi.org/10.1007/s10462-016-9467-9

Halim, Z., & Rehan, M. (2020). On identification of driving-induced stress using electroencephalogram signals: A framework based on wearable safety-critical scheme and machine learning. *Information Fusion, 53*(May 2019), 66–79. https://doi.org/10.1016/j.inffus.2019.06.006

Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., & Hsu, M. C. (2000). FreeSpan: Frequent pattern-projected sequential pattern mining. *Proceeding of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 355–359.

Hu, G., Hu, Y., Yang, K., Yu, Z., Sung, F., Zhang, Z., … Miemie, Q. (2018). Deep Stock Representation Learning: From Candlestick Charts to Investment Decisions. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. https://doi.org/10.1109/ICASSP.2018.8462215

Hu, W., Si, Y. W., Fong, S., & Lau, R. Y. K. (2019). A formal approach to candlestick pattern classification in financial time series. *Applied Soft Computing Journal, 84*, Article 105700. https://doi.org/10.1016/j.asoc.2019.105700

Hu, J., & Zheng, W. (2020). A deep learning model to effectively capture mutation information in multivariate time series prediction. *Knowledge-Based Systems, 203*, Article 106139. https://doi.org/10.1016/j.knosys.2020.106139

Huang, S. C., Chiou, C. C., Chiang, J. T., & Wu, C. F. (2020). Online sequential pattern mining and association discovery by advanced artificial intelligence and machine learning techniques. *Soft Computing, 24*(11), 8021–8039. https://doi.org/10.1007/s00500-019-04100-5

Huang, Y., Mao, X., & Deng, Y. (2021). Natural visibility encoding for time series and its application in stock trend prediction. *Knowledge-Based Systems, 232*, Article 107478. https://doi.org/10.1016/j.knosys.2021.107478

Iqbal, M., & Pao, H. K. (2021). Mining non-redundant distinguishing subsequence for trip destination forecasting. *Knowledge-Based Systems, 211*, Article 106519. https://doi.org/10.1016/j.knosys.2020.106519

Jiang, M., Liu, J., Zhang, L., & Liu, C. (2020). An improved Stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms. *Physica A: Statistical Mechanics and Its Applications, 541*(258), Article 122272. https://doi.org/10.1016/j.physa.2019.122272

Le, C., Shrestha, K. J., Jeong, H. D., & Damnjanovic, I. (2021). A sequential pattern mining driven framework for developing construction logic knowledge bases. *Automation in Construction, 121*(July 2020), 103439. https://doi.org/10.1016/j.autcon.2020.103439

Ley, M. (2002). *The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives BT - String Processing and Information Retrieval* (pp. 1–10). Springer Berlin Heidelberg.

Li, H., Li, Z., Peng, S., Li, J., & Tungom, C. E. (2020). Mining the frequency of time-constrained serial episodes over massive data sequences and streams. *Future Generation Computer Systems, 110*, 849–863. https://doi.org/10.1016/j.future.2019.11.008

Li, H., Liang, M., & He, T. (2017). Optimizing the Composition of a Resource Service Chain with Interorganizational Collaboration. *IEEE Transactions on Industrial Informatics, 13*(3). https://doi.org/10.1109/TII.2016.2616581

Liang, M., Wang, X., & Wu, S. (2021). A novel time-sensitive composite similarity model for multivariate time-series correlation analysis. *Entropy, 23*(6). https://doi.org/10.3390/e23060731

Lin, Y., Liu, S., Yang, H., & Wu, H. (2021). Stock trend prediction using candlestick charting and ensemble machine learning techniques with a novelty feature engineering scheme. *IEEE Access, 9*, 101433–101446. https://doi.org/10.1109/ACCESS.2021.3096825

Long, J., Chen, Z., He, W., Wu, T., & Ren, J. (2020). An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market. *Applied Soft Computing Journal, 91*, Article 106205. https://doi.org/10.1016/j.asoc.2020.106205

Madbouly, M., Elkholy, M., Gharib, Y., & Darwish, S. (2020). *Predicting Stock Market Trends for Japanese Candlestick Using Cloud Model* (pp. 628–645). doi:10.1007/978-3-030-44289-7_59.

Marszałek, A., & Burczyński, T. (2014). Modeling and forecasting financial time series with ordered fuzzy candlesticks. *Information Sciences, 273*, 144–155. https://doi.org/10.1016/j.ins.2014.03.026

Meng, X., Ma, J., Qiao, H., & Xie, H. (2021). Forecasting US Stock Market Returns: A Japanese Candlestick Approach. *Journal of Systems Science and Complexity, 34*(2), 657–672. https://doi.org/10.1007/s11424-020-9126-8

Mohanty, D. K., Parida, A. K., & Khuntia, S. S. (2021). Financial market prediction under deep learning framework using auto encoder and kernel extreme learning machine. *Applied Soft Computing, 99*, Article 106898. https://doi.org/10.1016/j.asoc.2020.106898

Naranjo, R., Arroyo, J., & Santos, M. (2018). Fuzzy modeling of stock trading with fuzzy candlesticks. *Expert Systems with Applications, 93*, 15–27. https://doi.org/10.1016/j.eswa.2017.10.002

Naranjo, R., & Santos, M. (2019). A fuzzy decision system for money investment in stock markets based on fuzzy candlesticks pattern recognition. *Expert Systems with Applications, 133*, 34–48. https://doi.org/10.1016/j.eswa.2019.05.012

Nison, S. (2001). *Japanese candlestick charting techniques: A contemporary guide to the ancient investment techniques of the Far East*. New York Institute of Finance.

Niu, T., Wang, J., Lu, H., Yang, W., & Du, P. (2020). Developing a deep learning framework with two-stage feature selection for multivariate financial time series forecasting. *Expert Systems with Applications, 148*, Article 113237. https://doi.org/10.1016/j.eswa.2020.113237

Pal, S. S., & Kar, S. (2022). Fuzzy transfer learning in time series forecasting for stock market prices. *Soft Computing, 2005*. https://doi.org/10.1007/s00500-021-06648-7

Pedrycz, W., & Chen, S.-M. (Eds.). (2013). *Time Series Analysis, Modeling and Applications - A Computational Intelligence Perspective*. Springer. https://doi.org/10.1007/978-3-642-33439-9.

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., & Hsu, M. C. (2001). PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. *Proceedings - International Conference on Data Engineering, 215–224*. https://doi.org/10.1109/icde.2001.914830

Shih, S. Y., Sun, F. K., & Lee, H. Y. (2019). Temporal pattern attention for multivariate time series forecasting. *Machine Learning, 108*(8–9), 1421–1441. https://doi.org/10.1007/s10994-019-05815-0

Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/bfb0014140

Tarus, J. K., Niu, Z., & Yousif, A. (2017). A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. *Future Generation Computer Systems, 72*, 37–48. https://doi.org/10.1016/j.future.2017.02.049

Thakkar, A., Patel, D., & Shah, P. (2021). Pearson Correlation Coefficient-based performance enhancement of Vanilla Neural Network for stock trend prediction.

Neural Computing and Applications, 33(24), 16985–17000. https://doi.org/10.1007/s00521-021-06290-2

U, J. H., Lu, P. Y., Kim, C. S., Ryu, U. S., & Pak, K. S. (2020). A new LSTM based reversal point prediction method using upward/downward reversal point feature sets. *Chaos, Solitons and Fractals*, *132*, 109559. doi:10.1016/j.chaos.2019.109559.

Udagawa, Y. (2019). Mining Stock Price Changes for Profitable Trade Using Candlestick Chart Patterns. In *BT - Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services, iiWAS 2019* (pp. 118–126). https://doi.org/10.1145/3366030.3366053

Udagawa, Y. (2018). *Predicting Stock Price Trend Using Candlestick Chart Blending Technique*. doi:10.1109/BigData.2018.8622402.

Wang, W., Tian, J., Lv, F., Xin, G., Ma, Y., & Wang, B. (2021). Mining frequent pyramid patterns from time series transaction data with custom constraints. *Computers and Security, 100*, Article 102088. https://doi.org/10.1016/j.cose.2020.102088

Wang, M., & Wang, Y. (2019). Evaluating the Effectiveness of Candlestick Analysis in Forecasting U.S. Stock Market. In *ICCDA 2019: Proceedings of the 2019 3rd International Conference on Compute and Data Analysis*. https://doi.org/10.1145/3314545.3314555

Wu, S., Wang, X., Liang, M., & Wu, D. (2021). Pfc: A novel perceptual features-based framework for time series classification. *Entropy, 23*(8), 1–23. https://doi.org/10.3390/e23081059

Yu, Z., Qin, L., Chen, Y., & Parmar, M. D. (2020). Stock price forecasting based on LLE-BP neural network model. *Physica A: Statistical Mechanics and Its Applications, 553*, Article 124197. https://doi.org/10.1016/j.physa.2020.124197

Yu, P., & Yan, X. (2020). Stock price prediction based on deep neural networks. *Neural Computing and Applications, 32*(6), 1609–1628. https://doi.org/10.1007/s00521-019-04212-x

Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning, 42*(1–2), 31–60. https://doi.org/10.1023/A:1007652502315

Zhang, Y., Yan, B., & Aasma, M. (2020). A novel deep learning framework: Prediction and analysis of financial time series using CEEMD and LSTM. *Expert Systems with Applications, 159*, Article 113609. https://doi.org/10.1016/j.eswa.2020.113609

Zhipeng, J. (2019). Financial time series forecasting based on characterized candlestick and the support vector classification with cooperative coevolution. *Journal of Computers, 14*, 195–209. https://doi.org/10.17706/jcp.14.3.195-209