ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/335936874

A Stock Index Prediction Framework: Integrating Technical and Topological Mesoscale Indicators

Conference Paper · July 2019

DOI: 10.1109/IRI.2019.00018

CITATIONS
0

READS
143

6 authors, including:

Zhan Bu
Nanjing University of Finance and Economics
49 PUBLICATIONS **687** CITATIONS

SEE PROFILE

Xi Xiong
Chengdu University of Information Technology
29 PUBLICATIONS **79** CITATIONS

SEE PROFILE

Hong-Liang Sun
University of Nottingham
9 PUBLICATIONS **30** CITATIONS

SEE PROFILE

Chengcui Zhang
University of Alabama at Birmingham
173 PUBLICATIONS **2,520** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Public Opinion Analysis View project

Multiple-Object Image Retrieval Framework Using Hierarchical Region Tree View project

All content following this page was uploaded by Hong-Liang Sun on 19 October 2019.

The user has requested enhancement of the downloaded file.

A Stock Index Prediction Framework: Integrating Technical and Topological Mesoscale Indicators

Zi Qi*, Zhan Bu*, Xi Xiong†, Hongliang Sun*, Jie Cao*, and Chengcui Zhang‡
*Nanjing University of Finance and Economics, Nanjing, China
†Chengdu University of Information Technology, Chengdu, China
‡University of Alabama at Birmingham, United States
(Corresponding Author: Zhan Bu, Email: buzhan@nuaa.edu.cn)

***Abstract*—With its growing importance in predicting future stock trends, nearly everyone watches the Chinese financial market. Traditional approaches typically employ a variety of statistical techniques or machine learning methods for stock index predicting, and often rely on analysis of technical indicators. In the existing literature, researchers rarely attempt to predict the stock index by using the topological features of temporal stock correlation networks. Keeping this in mind, we first calculate the correlation coefficient of any two stocks using the classic Visibility Graph Model (VGM). Then, by using the Planar Maximally Filtered Graph (PMFG) method, we generate temporal stock correlation networks from historical stock quantitative data. Next, we choose fourteen frequently adopted Technical Indicators (TIs) and five Topological Mesoscale Indicators (TMIs, extracted from the temporal stock correlation networks) as predictive variables of six machine learning classifiers. To improve forecast accuracy and to address potential overfitting problems, we modify the classic Sequential Backward Selection (SBS) algorithm to learn the most significant predictive variables for each classifier. We then conduct a series of comprehensive experiments on three Chinese stock indices to validate our prediction framework's performance. Experimental results show that using a combination of TIs and TMIs significantly improves forecast accuracy over conventional methods that use either TIs or TMIs exclusively.**

***Keywords*—machine learning; Technical Indicators; Topological Mesoscale Indicators; stock correlation networks;**

I. Introduction

To gain maximum profit from their stock portfolios, investors often strive to predict future stock market trends, which is a daunting task because the financial market is a highly volatile, nonstationary, and nonlinear dynamic system. To measure overall market behavior, investors study various stock indices, such as the American Standard & Poor 500 (S&P 500), the Japanese Nikkei 225, and the British Financial Times Stock Exchange 100 (FTSE 100), via the prices of selected constituent stocks. An effective prediction model for forecasting the stock index provides investors with information about stock fluctuations or a specific portfolio's returns. For short-term investors, it also serves as an early warning system about possible sudden drops in the financial market.

Traditional studies on stock index forecasting identify capital market anomalies with high explanatory values by analyzing macroeconomic indicators, such as the Consumer Price Index (CPI), Gross National Product (GNP), and Gross Domestic Product (GDP). However, because daily macroeconomic indicators are impossible to obtain, these methods are incapable of capturing impact of recent trends. Another research direction relies on technical analysis that assumes stock price and volume are the two most relevant factors in market trend forecasting. Based on this hypothesis, multiple Technical Indicators (TIs) are designed to characterize a particular stock index's fluctuation behavior [1], [2]. However, average investors find it challenging to apply this approach, because too many TIs can be considered forecasting factors. Also, no single TI alone yields sufficiently accurate market prediction. Recent studies seek to construct temporal stock correlation networks by using historical quantitative data and extracting Topological Mesoscale Indicators (TMIs) to capture the co-movement rules of constituent stocks in a particular index [3], [4], [5]. However, most of this research only analyzes and/or presents statistical relationships between the stock index and the predefined TMIs; discerning how to adopt these rules for stock index prediction remains unclear.

Some studies formulate the problem of effectively integrating multiple indicators for stock index prediction as a binary classification problem, i.e., as future trend prediction (the rise or fall of a particular stock index) [2], [6]. The emergence of machine learning and artificial intelligence algorithms has made such binary classification possible, but these algorithms' effectiveness highly depends on the selection of input variables, which remains an open problem in machine learning. Moreover, the noisy stock market may lead machine learning classifiers to build highly complicated models, which could result into the overfitting problem. To address these challenges, we propose a generalized stock index prediction framework composed of four coupled phases: (i) build stock distance matrices based on the Visibility Graph Method (VGM), (ii) generate stock correlation networks using the Planar Maximally Filtered Graph (PMFG) model, (iii) extract TMIs based on complex network theory, and (iv) determine the most predictive variables via modified sequential backward selection (SBS). The most predictive variables are used to train the machine learning classifiers, and we then use those classifiers to predict future trends in three Chinese stock indices. The main goal of this work is to explore whether the combination of TIs and TMIs can improve the forecast accuracy of existing classifiers. This paper's contributions are summarized as follows:

- To capture each stock's microscopic fluctuation within a certain time window, we first use the VGM to map the stock price/volume series into a visible network. In this way, we can measure the similarity/distance of any two stocks by Jaccard coefficient of the two edge sets of the corresponding networks.
- By reviewing the most related works from the past few decades, we choose fourteen frequently adopted TIs as predictive variables. In addition, we extract five TMIs from temporal stock correlation networks (filtered from the corresponding stock distance matrices via the PMFG model) to complement the variable space.
- To combine various predictive variables in machine learning classifiers for stock index prediction, we modify the classic SBS algorithm to learn the most predictive variables for each classifier. Experimental results on three Chinese stock indices show that the combination of TIs and TMIs significantly improve the forecast accuracy of most machine learning classifiers.

The remainder of this paper is organized as follows. In Section II, we review related work on stock market forecasting. In Section III, we provide the main mathematical definitions and clarify this paper's motivation. The details of our stock index prediction framework are introduced in Section IV. Extensive experiments are introduced in Section V to evaluate our framework's performance. We conclude in Section VI.

II. Related Work

Stock market prediction is a hot topic in financial technology (fintech). Existing studies relied heavily on technical analysis to construct the predictive variable space, then used different machine learning models, such as support vector machines (SVMs), artificial neural networks (ANNs), and decision tree (DTs), to forecast the stock index's future direction. Kim et al. [7] applied SVMs and ANNs based on economic indicators to forecast stock index direction. Experimental results showed that SVMs provided a promising alternative for stock market prediction. To boost stock market profits, traders need to use a variety of forecasting techniques to obtain more signals and information. Research showed that ten data mining techniques based on economic variable indicators [8] were used to predict price movements in the Hang Seng Index of Hong Kong. From that comparison, the SVM was better than using a least squares support vector machine (LS-SVM) for in-sample prediction, but an LS-SVM was better than the SVM for out-of-sample forecasts in terms of hit and error rates. Although SVMs are prevalent in stock prediction research, it is difficult to implement them for large-scale training samples, because such samples consume a great deal of time and computational resources. Kara et al. [2] proved that ANNs and SVMs were useful prediction tools, and that ANNs were significantly better than SVMs in predicting the Istanbul stock exchange's direction. Ten TIs were selected as model inputs, but the ANN model trained by the Back Propagation (BP) algorithm had some limitations, such as easily converging to the local minimum because of the stock market data's tremendous noise and complex dimensionality. Qiu [1] used an optimized ANN model to predict the direction of the Japanese stock market index based on TIs. The model addressed local convergence for nonlinear optimization problems, and results showed that the method was more effective and resulted in higher prediction accuracy compared to SVMs. Liu and Song [9] also proposed a novel model (a multilayer stochastic ANN bagging) by integrating bagging and an ANN. Specifically, they used weak ANNs to get information without overfitting and obtained better results by combining weaker results via optimized bagging. The model used a combination of indicators and achieved a 3 to 15 percent improvement over S&P 500 index prediction compared to other classifiers. Most of these studies have some limitations for medium- to long-term prediction. Basak et al. [6] used Random Forest (RF) and eXtreme Gradient Boosting (XGBoost) classifiers to facilitate this connection by using DT ensembles to build a predictive model, resulting in higher accuracies for long-term predictions.

Generally, the stock market is a well-defined complex system consisting of interacting stocks and instruments that inspires researchers to model correlation coefficients among stocks and then construct stock networks. Stock networks received increasing attention in the scientific literature over the last two decades. Tumminello et al. [3] investigated the PMFGs of the 300 most capitalized stocks traded on the New York Stock Exchange from 2001 to 2003, extracting topological properties such as the average shortest path length, betweenness, and degree from them. Analysis confirmed that the selected stocks composed a hierarchical system that progressively structured itself as topological properties changed. Xia et al. [4] proposed a threshold model to build Chinese stock networks and studied their topological properties. The results showed that during stock market turbulence, network structure exhibited fairly high modularity, mean cross-correlation, and mean degree. Xu et al. [5] also proposed an efficient method for estimating the optimal threshold for stock network construction. By analyzing the generated networks' topological characteristics, three global financial crises could be distinguished from an evolutionary perspective. The common point of these last three methods is that the researchers built stock networks by calculating the Pearson correlation coefficient [10] between stocks. Real cases would have more nonlinear relationships. Additionally, large thresholds favor strong correlations, which leads to discarded information. To address these issues, Guo et al. [11] developed two stock networks through mutual information and correlation coefficients and studied these networks' topological properties to analyze the Chinese stock market. But because trading and investment decisions made by a large number of market participants change across different time scales, true correlations among different stocks differ across various time scales. Therefore, Wang et al. [12] proposed multiscale correlation networks to analyze the US stock market in terms of wavelet analysis, the Minimum Spanning Tree (MST), and PMFGs; they found that the topological features of MSTs and PMFGs at different wavelet time scales in particular were different.

In summary, previous works mainly concluded that we can apply stock correlation network topologies to detect stock market turbulences by analyzing temporal characteristic evolution. In other words, we can detect market instability through the evolution of a financial network's topological structure. Thus, the TMIs of stock correlation networks are effect factors that influence stock prices and future trends, which are significant for predicting a stock index's future trends. Inspired by these insights, this study will extract TMIs from stock correlation networks and combine TIs as predictive variables in machine learning classifiers for stock index prediction.

III. Basic Definitions and Motivation

We introduce some preliminary definitions as the basis of our proposed method. Stock index prediction is a challenging task because of the stock market's highly volatile and non-stationary nature. Predicting a stock index trade for a short- or long-term time range relies on capturing and modeling the actions of the index's constituent stocks, typically by observing and evaluating, historical stock quantitative data. Let $\mathbf{I} = \{1, 2, \cdots, n\}$ denote the set of n constituent stocks in a particular stock index, such as the American S&P 500, the Japanese Nikkei 225, or the British FTSE 100. For $\forall i \in \mathbf{I}$, let $\mathbf{p}_i = (p_{i,1}, p_{i,2}, \cdots, p_{i,T})$ and $\mathbf{r}_i = (r_{i,1}, r_{i,2}, \cdots, r_{i,T})$ denote the time series of daily closing prices and daily turnover rates of stock i over T consecutive trading days, respectively. $p_{i,t}$ and $r_{i,t}$ indicate the closing price and turnover rate of stock i at day t, respectively. Let $\mathbf{p}^t = (p_{1,t}, p_{2,t}, \cdots, p_{n,t})^\top$ denote the closing price vector of all constituent stocks at day t, thus the stock index at day t can be given by $I^t = \mathcal{F}(\mathbf{p}^t)$, where $\mathcal{F}(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is the index calculation function.

For the time window ending at day t ($l \le t \le T$) with length l ($2 \le l \ll T$), $\mathbf{p}_i^{(t,l)} = (p_{i,t-l+1}, p_{i,t-l+2}, \cdots, p_{i,t})$ and $\mathbf{r}_i^{(t,l)} = (r_{i,t-l+1}, r_{i,t-l+2}, \cdots, r_{i,t})$ denote the movements of the closing price and turnover rate of stock i, respectively. For each such l-dimensional vector $\mathbf{p}_i^{(t,l)}$, the corresponding ordinal partition is defined as $\Theta_i^{(t,l)} = (\alpha_{i,1}^{(t,l)}, \alpha_{i,2}^{(t,l)}, \cdots, \alpha_{i,l}^{(t,l)})$, such that $\alpha_{i,o}^{(t,l)} = p_{i,t-l+o}$ ($o \in \{1, 2, \cdots, l\}$). Analogously, the corresponding ordinal partition with regard to $\mathbf{r}_i^{(t,l)}$ is denoted by $\Phi_i^{(t,l)} = (\beta_{i,1}^{(t,l)}, \beta_{i,2}^{(t,l)}, \cdots, \beta_{i,l}^{(t,l)})$, where $\beta_{i,o}^{(t,l)} = r_{i,t-l+o}$. Thus, we can represent the movements of the closing prices and turnover rates associated with all constituent stocks within this observation window as $\Theta^{(t,l)} = (\alpha_{i,o}^{(t,l)}) \in \mathbb{R}^{n \times l}$ and $\Phi^{(t,l)} = (\beta_{i,o}^{(t,l)}) \in \mathbb{R}^{n \times l}$, respectively. A straightforward approach for stock index prediction is to establish a price forecasting model for each constituent stock, such as $\mathcal{M}_i(\Theta_i^{(t,l)}, \Phi_i^{(t,l)}) \to \widetilde{p}_{i,t+1}$, where $\widetilde{p}_{i,t+1}$ indicates the closing price of stock i at time $t+1$ predicted by the forecasting model $\mathcal{M}_i(\cdot,\cdot) : \mathbb{R}^l \times \mathbb{R}^l \to \mathbb{R}$, using historical stock data. Thus, the predicted stock index at time $t+1$ can be given by $\mathcal{F}(\widetilde{\mathbf{p}}^{t+1})$, where $\widetilde{\mathbf{p}}^{t+1} = (\widetilde{p}_{1,t+1}, \widetilde{p}_{2,t+1}, \cdots, \widetilde{p}_{n,t+1})^\top$. The effectiveness of this approach highly relies on the forecast accuracy of each $\mathcal{M}_i(\cdot,\cdot) : \mathbb{R}^l \times \mathbb{R}^l \to \mathbb{R}$, which usually performs poorly in practical application.

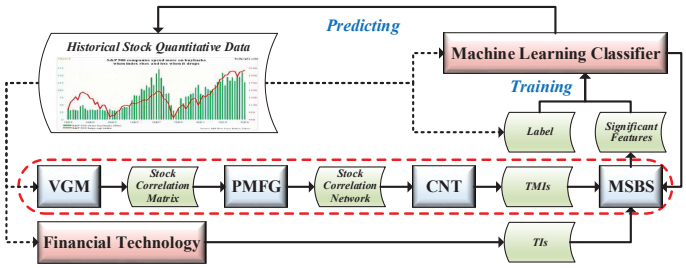

Fig. 1: The framework for our stock index prediction approach.

Instead of predicting stock index value, in this paper we focus on predicting stock index direction. Naturally, we can formulate this as a classic binary classification problem. In the literature, several machine learning classifiers [13] have been developed for stock market prediction, most of which extract TIs directly from historical stock quantitative data to construct the training set, such as $TrainSet = \{\langle (f_1^t, f_2^t, \cdots, f_\kappa^t), label^t \rangle | t = l, l+1, \cdots, T\}$, where $(f_1^t, f_2^t, \cdots, f_\kappa^t)$ denotes the feature vector, including κ TIs extracted from $\Theta^{(t,l)}$ and/or $\Phi^{(t,l)}$; $label^t = +1$ if $I^{t+1} > I^t$ and $label^t = -1$ otherwise. In contrast, we consider the co-movement effect of the constituent stocks in a particular stock index. Using the temporal data of $\Theta^{(t,l)}$ and/or $\Phi^{(t,l)}$, we can generate stock correlation networks from which certain TMIs can be extracted to complement the variable space, enabling us to combine the strength of TIs and TMIs in machine learning classifiers for stock index prediction. The motivation of this work is to study such TMIs' effectiveness via the prediction accuracy of machine learning classifiers.

IV. Methodology

This section presents our proposed framework for stock market prediction in more detail—specifically, how it extracts effective TMIs from historical stock quantitative data and combines these predictive variables with conventional TIs into a prediction model. Fig. 1 shows the overall framework, which is comprised of four major phases: build stock distance matrices based on the VGM, generate stock correlation networks using the PMFG model, extract TMIs based on the Complex Network Theory (CNT), and learn the most predictive variables using the modified version of SBS (MSBS). The most predictive variables are used to train the machine learning classifiers, which we then use to predict movement in three Chinese stock indices.

A. Visibility Graph Model

In time-series analysis, VGM can be used to map a given time series into a visible network. The basic motivation for using these methods is that, by performing such a transformation, we can then apply a range of techniques from complex network theories to analyzing and quantifying the time series' dynamic characteristics. To our knowledge, Zhang et al. [14] was the first to employ this type of approach, constructing visible networks from pseudoperiodic time series

and representing each cycle as a single node in the network. In Zhang's method, nodes are linked if corresponding cycles are sufficiently similar. Thus, someone can choose a measure of similarity and a threshold to generate different visible networks using the same pseudoperiodic time series. Another representative work was proposed by Lacasa [15], who focused on measuring correlation and self-similarity in the distribution of data values for temporal successors. The "visibility constraint" maps individual scalar observations to samples and connects nodes subject to a linear convexity constraint. This model's main strength is that it is well-suited for characterizing different stochastic processes and certain self-similar sequences.

Given an ordinal partition of the time series for the daily closing price of stock i, such as $\Theta_i^{(t,l)} = (\alpha_{i,1}^{(t,l)}, \alpha_{i,2}^{(t,l)}, \cdots, \alpha_{i,l}^{(t,l)})$, we can treat it as a set of l-ordered data points, where each data point can be represented as $(o, \alpha_{i,o}^{(t,l)})$ $(o \in \{1, 2, \cdots, l\})$. For clarity, we can visualize these l data points by using vertical bars in the landscape (see Fig. 2). By considering each data point as a node in a visible network, the graph representation of $\Theta_i^{(t,l)}$ can be achieved from the following visibility criteria: any two arbitrary data points $(x, \alpha_{i,x}^{(t,l)})$ and $(y, \alpha_{i,y}^{(t,l)})$ $(\forall x, y \in \{1, 2, \cdots, l\}, x < y)$ will have visibility and will be connected if any other data point $(z, \alpha_{i,z}^{(t,l)})$ placed between them $(\forall z \in \{1, 2, \cdots, l\}, x < z < y)$ fulfills:

$$\alpha_{i,z}^{(t,l)} < \alpha_{i,y}^{(t,l)} + (\alpha_{i,x}^{(t,l)} - \alpha_{i,y}^{(t,l)}) \frac{y - z}{y - x}. \tag{1}$$

Based on the aforementioned criteria, we easily can check that the extracted network from $\Theta_i^{(t,l)}$ is always connected (each node sees at least its nearest neighbors), undirected (there is no direction defined in the edges), and invariant under affine transformations of the series data ([15]. Similarly, we can also extract a visible network from $\Phi_i^{(t,l)}$ using the same visibility criteria). Edge sets of the visible networks extracted from $\Theta_i^{(t,l)}$ and $\Phi_i^{(t,l)}$ are denoted by

$$\begin{aligned} \mathcal{E}_i^{(t,l)} = \{e_{x,y}^{(t,l)} | \forall z \in \{1, 2, \cdots, l\}, x < z < y, \\ \alpha_{i,z}^{(t,l)} < \alpha_{i,y}^{(t,l)} + (\alpha_{i,x}^{(t,l)} - \alpha_{i,y}^{(t,l)}) \frac{y - z}{y - x}\}, \end{aligned} \tag{2}$$

and

$$\begin{aligned} \mathcal{L}_i^{(t,l)} = \{e_{x,y}^{(t,l)} | \forall z \in \{1, 2, \cdots, l\}, x < z < y, \\ \beta_{i,z}^{(t,l)} < \beta_{i,y}^{(t,l)} + (\beta_{i,x}^{(t,l)} - \beta_{i,y}^{(t,l)}) \frac{y - z}{y - x}\}, \end{aligned} \tag{3}$$

respectively. Thus, the correlation coefficient between stocks i and j in the observation window of length l $(2 \leq l \ll T)$ ending at day t is measured by

$$s_{i,j}^{(t,l)} = \gamma \frac{|\mathcal{E}_i^{(t,l)} \cap \mathcal{E}_j^{(t,l)}|}{|\mathcal{E}_i^{(t,l)} \cup \mathcal{E}_j^{(t,l)}|} + (1 - \gamma) \frac{|\mathcal{L}_i^{(t,l)} \cap \mathcal{L}_j^{(t,l)}|}{|\mathcal{L}_i^{(t,l)} \cup \mathcal{L}_j^{(t,l)}|}, \tag{4}$$

where $\gamma \in (0, 1)$ is the mixed parameter that adjusts the weights of the price and turnover movements (in our experiments, we set $\gamma = 0.5$ as the default value). Considering that

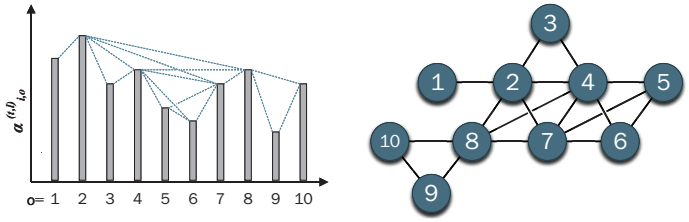

Fig. 2: The illustration of a time series (10 data points) and the associated graph derived from the VGM proposed by Lacasa et al. [15].

the correlation coefficient cannot measure the distance between two stocks, we use the following function to transform the correlation coefficient to distance, such that

$$d_{i,j}^{(t,l)} = \sqrt{2(1 - s_{i,j}^{(t,l)})}. \tag{5}$$

Therefore, the distances among all constituent stocks in such an observation window are denoted by an $n \times n$ stock distance matrix, such as $\mathbf{D}^{(t,l)} = (d_{i,j}^{(t,l)}) \in \mathbb{R}^n \times \mathbb{R}^n$, such that $\forall i, j \in \mathbf{I}, d_{i,j}^{(t,l)} = d_{j,i}^{(t,l)}$ and $\forall i \in \mathbf{I}, d_{i,i}^{(t,l)} = 0$.

B. Stock Correlation Network

Over the last decade, several studies have proposed network-based approaches for modeling stock correlations because of their proven efficacy in predicting stock market dynamics. In our framework, the stock distance matrix $\mathbf{D}^{(t,l)}$ defined in (5) can be viewed as the weighted adjacency matrix of a fully connected stock network, where the distance between any stock pair, such as $d_{i,j}^{(t,l)}$, can be regarded as the weight of the corresponding undirected link. The lower this weight is, the more the two stocks correlate. Accordingly, the most straightforward approach to building a stock correlation network is to design a filtering criterion that retains highly correlated stock pairs and discards lowly correlated ones.

In the existing literature, three filtering criteria are commonly used: the Statistical Threshold Method (STM), the Minimum Spanning Tree (MST), and PMFGs [16]. The STM treats any two stocks as two connected nodes if the distance between them, such as $d_{i,j}^{(t,l)}$, is lesser than or equal to a specified threshold θ (see the top right of Fig. 3, where $\theta = 0.3$). It is worth noting that the filtered network's topological structure depends heavily on the selection of threshold θ. To our knowledge, determining how to adaptively select the best θ values for different stock quantitative datasets remains a challenge. Alternatively, we can directly adopt classic MST algorithms to extract the MST from $\mathbf{D}^{(t,l)}$ (see the bottom left of Fig. 3). However, this method may cause a loss of information such as some highly correlated stock pairs being discarded and some lowly correlated ones being retained because of the topological reduction criteria. To address STM and MST drawbacks, Tumminello [16] adopted a compromise that extracted representative links to form a filtered graph that not only preserved MSTs' hierarchical organization but

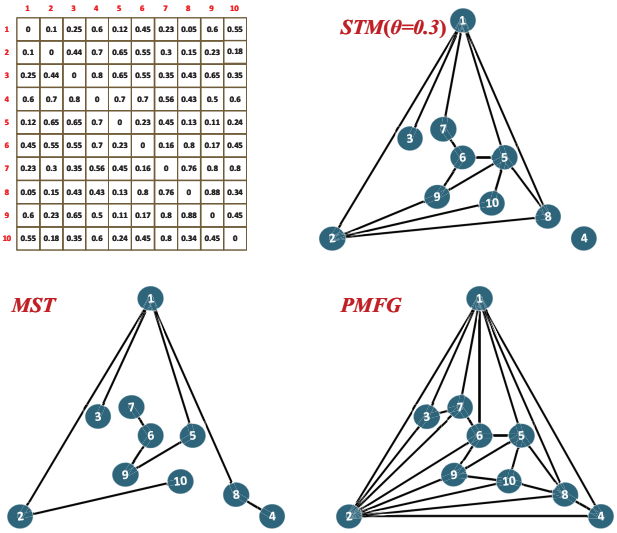

Fig. 3: Different stock correlation networks filtered by STM, MST, and PMFG, respectively.

also contained a larger amount of information in their internal structure. Given a stock distance matrix $\mathbf{D}^{(t,l)}$, we can construct such a graph by connecting stocks with the shortest distance following the filtering procedure below:

- **Step 1:** arrange the stock pairs in ascending order according to the value of the distance $d_{i,j}^{(t,l)}$ to construct an ordered list $\mathcal{L}^{(t,l)}$;
- **Step 2:** select the first unvisited stock pair (i,j) in $\mathcal{L}^{(t,l)}$ and add an edge (such as $e_{i,j}^{(t,l)}$, $i<j$) if and only if the resulting graph can still be embedded on a surface of genus $g \leq k$ after the edge insertion; and
- **Step 3:** repeat the previous step until all stock pairs in $\mathcal{L}^{(t,l)}$ have been visited.

The above filtering procedure can generate a simple, undirected, and connected graph embedded on a surface of genus $g=k$. It is worth noting that when $g=0$, the resulting graph can be embedded on the sphere, which Tumminello et al. [16] named a PMFG. The number of edges in a PMFG is $3(n-2)$ and $(n-1)$ for an MST, where n is the number of nodes. Because each node should participate in at least a clique of three nodes, a PMFG is actually a topological triangulation of the sphere (see the bottom right of Fig. 3).

Formally, the PMFG in an observation window of length l $(2 \leq l \ll T)$ ending at day t is defined as a two-tuple, such that $\mathbf{G}^{(t,l)} = \langle \mathbf{I}, \mathbf{E}^{(t,l)} \rangle$, where $\mathbf{I} = \{1, 2, \cdots, n\}$ denotes the set of n stocks, and $\mathbf{E}^{(t,l)} = \{e_{i,j}^{(t,l)} | \forall i,j \in \mathbf{I}, i<j\}$ is the set of $3(n-2)$ filtered edges. Let $N_i^{(t,l)} = \{j | \exists e_{i,j}^{(t,l)} \in \mathbf{E}^{(t,l)}, or\ \exists e_{j,i}^{(t,l)} \in \mathbf{E}^{(t,l)}\}$ denote the neighbor set of stock i in $\mathbf{G}^{(t,l)}$; the degree of stock i in such an observation window is then denoted by $k_i^{(t,l)} = |N_i^{(t,l)}|$. Obviously, for all $2 < l \ll T$ and $t = l, l+1, \cdots, T$, we have $m^{(t,l)} = \frac{1}{2}\sum_{i=1}^{n} k_i^{(t,l)} = 3(n-2)$, where $m^{(t,l)}$ is the number of edges in the filtered network $\mathbf{G}^{(t,l)}$.

C. Feature Extraction

To forecast stock market direction using machine learning techniques, we need to extract a certain number of predictive variables. By reviewing the most closely related works on stock market prediction from the past few decades [1], [2], [6], [7], [8], [9], [17], [18], [19], [20], fourteen widely used TIs are chosen for bearish or bullish signals (see details in Section V-A). In addition to these, we can also extract some TMIs from a stock correlation network $\mathbf{G}^{(t,l)}$. To study the filtered stock correlation network's topological and statistical properties, we introduce some mesoscale measures of $\mathbf{G}^{(t,l)}$ as follows:

Definition 1: **Average Edge Distance (AEDis).** Following the notion of "normalized tree length" proposed by Onnela et al. [21], the AEDis of $\mathbf{G}^{(t,l)}$ is defined as the average distance of all edges in the stock correlation network:

$$AEDis^{(t,l)} = \frac{\sum_{e_{i,j}^{(t,l)} \in \mathbf{E}^{(t,l)}} d_{i,j}^{(t,l)}}{3(n-2)}, \tag{6}$$

where $3(n-2)$ is the number of edges present in $\mathbf{G}^{(t,l)}$, and $d_{i,j}^{(t,l)}$ denotes the distance between stocks i and j. For a given observation window of length l $(2 \leq l \ll T)$ ending at time t, the smaller the $AEDis^{(t,l)}$ is, the shorter the distance is between stocks, implying that stocks have a tendency to become more correlated.

Definition 2: **Characteristic Path Length (CPLen).** The CPLen of $\mathbf{G}^{(t,l)}$ is defined as the average distance of the shortest path between any two stocks:

$$CPLen^{(t,l)} = \frac{2}{n(n-1)} \sum_{i<j} \xi_{i,j}^{(t,l)}, \tag{7}$$

where $\xi_{i,j}^{(t,l)}$ indicates the length of the shortest path from stocks i to j. The smaller the $CPLen^{(t,l)}$ is, the higher the connectivity of $\mathbf{G}^{(t,l)}$ is, implying that a piece of information (for example, stock price fluctuations) can propagate through the stock market more quickly.

Definition 3: **Diameter.** The diameter, or the linear size of the stock correlation network, is defined as the shortest path length between the two most distant stocks:

$$Diameter^{(t,l)} = \max_{i<j} \xi_{i,j}^{(t,l)}. \tag{8}$$

Similar to CPLen, the smaller the $Diameter^{(t,l)}$ is, the more significant the observation network's "small-world" property [22] is.

Definition 4: **Clustering Coefficient (CCoef).** The CCoef of a stock in $\mathbf{G}^{(t,l)}$ is the ratio of existing edges connecting a stock's neighbors to each other to the maximum possible number of such edges, meaning that the CCoef of such a network is the average of the clustering coefficients of all stocks:

$$CCoef^{(t,l)} = \frac{1}{n} \sum_{i=1}^{n} \frac{2\epsilon_i^{(t,l)}}{k_i^{(t,l)}(k_i^{(t,l)}-1)}, \tag{9}$$

where $k_i^{(t,l)}$ is the number of neighbors of stock i, and $\epsilon_i^{(t,l)}$ is the number of edges between $k_i^{(t,l)}$ neighbors in $\mathbf{G}^{(t,l)}$. The $CCoef^{(t,l)}$ measures the property "all my friends know each other". A high-clustering coefficient for a network is another indication of the "small world" property [22].

Stock index performance in an observation window of length l ($2 \leq l \ll T$) ending at day t is given by $\log\left(\frac{I^t}{I^{t-l+1}}\right)$. For $\forall i \in \{1, 2, \cdots, n\}$, let $c_i^{(t,l)}$ denote the categorical attribute of stock i, which is defined as

$$c_i^{(t,l)} = \begin{cases} +1 & \log\left(\frac{p_{i,t}}{p_{i,t-l+1}}\right) > \log\left(\frac{I^t}{I^{t-l+1}}\right) \\ -1 & otherwise \end{cases}. \tag{10}$$

In other words, stock i is labeled as $+1$ if it outperforms the stock index in the given observation window; otherwise, it is -1 . Similarly, all n stocks can be classified into two classes according to their own performance.

Definition 5: **Assortativity Coefficient (ACoef).** The ACoef of $\mathbf{G}^{(t,l)}$ measures how much better the attributes match across edges than expected at random:

$$ACoef^{(t,l)} = \frac{\sum_{i,j}\left(A_{i,j}^{(t,l)} - \frac{k_i^{(t,l)}k_j^{(t,l)}}{6(n-2)}\right)\delta(c_i^{(t,l)}, c_j^{(t,l)})}{6(n-2) - \sum_{i,j}\frac{k_i^{(t,l)}k_j^{(t,l)}}{6(n-2)}\delta(c_i^{(t,l)}, c_j^{(t,l)})}, \tag{11}$$

where $\mathbf{A}^{(t,l)} = (A_{i,j}^{(t,l)}) \in \mathbb{R}^n \times \mathbb{R}^n$ is the $n \times n$ adjacency matrix of $\mathbf{G}^{(t,l)}$; $\delta(c_i^{(t,l)}, c_j^{(t,l)}) = 1$ if $c_i^{(t,l)} = c_j^{(t,l)}$; otherwise, $\delta(c_i^{(t,l)}, c_j^{(t,l)}) = 0$.

There are many other measures that can be used to construct the topological and statistical properties of temporal stock correlation networks. Readers who are interested in this research topic can refer to some recent articles [23], [24], [25].

D. Feature Selection

In many data-driven classification applications, the number of raw predictive variables can expand from tens to hundreds, and some variables are either redundant or irrelevant. If a machine learning classifier uses all the variables to train its classification model, it could lead to high computation cost and an inclination toward overfitting. As shall be explained in the following, our method avoids these pitfalls by removing some noisy/irrelevant variables to improve classification performance. Formally, let $\mathbf{F} = \{f_1, f_2, \cdots, f_K\}$ denote the initial set of variables; the task of feature selection is to choose a subset of $\mathbf{F}$ (for example, $\mathbf{F}' \subseteq \mathbf{F}$) that can efficiently describe the input data while reducing noise or irrelevant variables and still providing good prediction results. We further use $\mathbb{J}(\mathbf{F}', \mathcal{M})$ to denote the classification accuracy of a particular machine learning classifier (for example, $\mathcal{M}$) when using the predictive variables in $\mathbf{F}'$. With this notation, the feature selection problem can be translated into the following maximization problem:

$$\widehat{\mathbf{F}} = \arg\max_{\mathbf{F}' \subseteq \mathbf{F}} \mathbb{J}(\mathbf{F}', \mathcal{M}). \tag{12}$$

To determine the "best" $\widehat{\mathbf{F}}$ associated with classifier $\mathcal{M}$, we modify the classic Sequential Backward Selection (SBS) [26] approach, specifically, by taking the whole K raw variables as input. At each step τ, we employ a take-out-and-put-in strategy to remove some variables from the current feature subset $\mathbf{F}^\tau$. Note that each abandoned variable should improve classifier performance if it is removed from $\mathbf{F}^\tau$. We repeat this procedure until the termination criterion is satisfied (for example, the classifier performance cannot be improved by removing any single variable). Algorithm 1 shows our feature selection algorithm's pseudocode. Note, too, that our proposed approach is independent of the user-specified machine learning classifier $\mathcal{M}$.

Algorithm 1: The modified SBS

Require: $\mathbf{F} = \{f_1, f_2, \cdots, f_K\}$ and $\mathcal{M}$;
Ensure: The "best" feature subset $\widehat{\mathbf{F}}$;

```
τ ← 0;
F^τ ← F;
repeat
    R^τ ← ∅;
    for each f_k ∈ F^τ do
        if J(F^τ − {f_k}, M) > J(F^τ, M) then
            R^τ.add(f_k);
        end if
    end for
    F^{τ+1} ← F^τ − R^τ;
    τ ← τ + 1;
until R^{τ−1} == ∅
F̂ ← F^τ;
return F̂;
```

V. Experimental Results

A. Experimental Setting

To test our framework, we collect the daily stock quantitative data of the corresponding constituent stocks of three Chinese stock indices: Shenzhen 100 (from August 2017 to October 2018, denoted as *SZ100*), Hushen 300 (from January 2016 to December 2017, denoted as *HS300*) and Shanghai Stock Exchange 380 (from May 2017 to May 2018, denoted as *SSE380*). In our experiments, the observation window's length l is specified as $\{5, 10, 20\}$, respectively, and for each stock index, we first calculate the stock distance matrix using the time-series data of all constituent stocks between trading days $t-l+1$ and t to generate the corresponding stock correlation network $\mathbf{G}^{(t,l)}$. We also extract five TMIs (see Definitions 1 through 5 in Section IV-C) from $\mathbf{G}^{(t,l)}$ to characterize a given stock index in the observation window of $[t-l+1, t]$. Alternatively, one could also describe the stock index on trading day t using TIs. The label of the data sample associated with trading day t is determined by the stock index on trading day $t+1$, that is $label^t = +1$ if $I^{t+1} > I^t$ and $label^t = -1$ otherwise.

In this paper, we consider six popular machine learning classifiers: SVM (SVM), Naive Bayes (NB), K-Nearest Neighbor (KNN), C4.5, MultiLayer Perceptron (MLP) and AdaBoost. The performance of a particular classifier $\mathcal{M}$ using a given feature set $\mathbf{F}$ can be quantitatively evaluated via the following metric:

$$\mathbb{J}(\mathbf{F}, \mathcal{M}) = \frac{tp + tn}{tp + fp + tn + fn}, \tag{13}$$

where tp is the number of samples correctly labeled as positive, tn is the number of samples correctly labeled as negative, fp is the number of samples incorrectly labeled as positive, and fn is the number of samples incorrectly labeled as negative.

To study the TMIs' effectiveness in improving machine learning classifiers' prediction accuracy, we design three types of classifiers:

- The first type of classifiers only selects predictive variables from the set of fourteen TIs, i.e., $\mathbf{F}_1^0$=$\{ROC, RSI, \%R, \%K, \%D, MA, EMA, WMA, TMA, MACD, MBI, CCI, PSY, OBV\}$.
- The second type only chooses predictive variables from the set of five TMIs, i.e., $\mathbf{F}_2^0$=$\{AEDis, CPLen, Diameter, CCoef, ACoef\}$.
- The third type chooses predictive variables from the set of fourteen TIs and five TMIs, such as $\mathbf{F}_3^0 = \mathbf{F}_1^0 \bigcup \mathbf{F}_2^0$.

To verify the effectiveness of the proposed, each dataset is divided into three nonoverlapping parts in chronological order: the first part (60 percent, denoted by **DataA**) is combined with the middle part (20 percent, denoted by **DataB**) for the feature selection task (we iteratively train a specific machine learning classifier using **DataA** and test the learned classification model on **DataB**). Finally, we employ the resulting "best" variable subset $\widehat{\mathbf{F}}$ on the last part (20 percent, denoted by **DataC**) to test a given machine learning classifier's overall prediction accuracy.

B. Performance of Feature Selection

To evaluate the effectiveness of the proposed feature selection algorithm, we examine its performance on the *SSE380* dataset with l set to 10. The initial feature sets associated with each individual classifier are $\mathbf{F}_1^0$, $\mathbf{F}_2^0$ and $\mathbf{F}_3^0$, respectively, where the superscript indicates the iteration round. Fig. 4 shows the processes of feature selection for different classifiers. In each subfigure, the y-axis represents a particular classifier's accuracy (such as training using **DataA** and testing using **DataB**) when employing the given feature set. From Fig. 4, three observations are obvious. First, for all classifiers except MLP and AdaBoost, $\mathbb{J}(\mathbf{F}_1^0, \mathcal{M}) > \mathbb{J}(\mathbf{F}_2^0, \mathcal{M})$, implying that, compared to TMIs, TIs might be more useful in training a machine learning classifier to predict the stock index. Second, for all classifiers except NB, the proposed feature selection algorithm works well, and the "best" feature set can be learned within three iterations. Third, for all classifiers, using the "best" combination of TIs and TMIs can achieve the highest classification accuracy, indicating the added differentiation power enabled by TMIs. It is worth noting that by changing the dataset and the observation window length, the main findings obtained in this experiment are also tenable. We present and discuss more details about the forecast results of different machine learning classifiers on **DataC** in the next subsection.

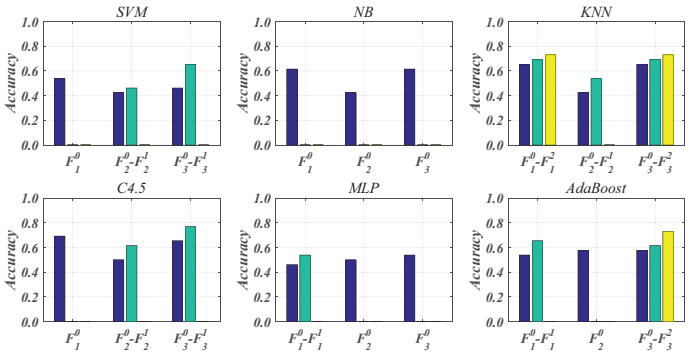

Fig. 4: Feature selection for different machine learning classifiers on *SSE380*, where $l = 10$.

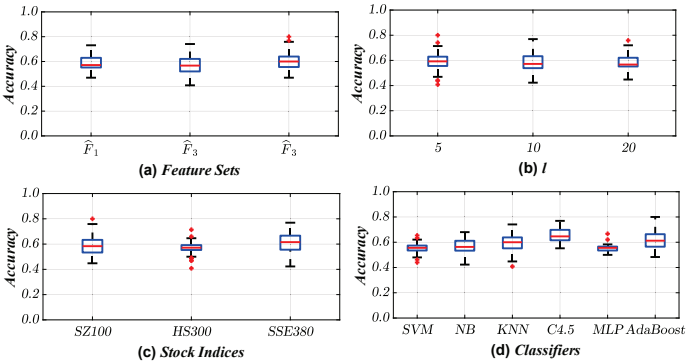

Fig. 5: Stock index prediction performance using different (a) feature sets, (b) observation windows, (c) stock indices, and (d) machine learning classifiers.

C. Performance of Stock Index Prediction

To further evaluate our proposed stock index prediction framework, we denote the initial variable sets to be $\mathbf{F}_1^0$, $\mathbf{F}_2^0$ and $\mathbf{F}_3^0$ for each classifier $\mathcal{M}$ and each stock index, respectively. Then, we employ Algorithm 1 to obtain the "best" variable subsets (such as $\widehat{\mathbf{F}}_1$, $\widehat{\mathbf{F}}_2$ and $\widehat{\mathbf{F}}_3$) for each set of l. Finally, we test each classifier's accuracy on **DataC** by using the selected predictive variables. Fig. 5 provides details about these result. Without loss of generality, we can calculate stock index prediction accuracy in four different ways, such as via various subsets, observation windows, datasets, and classifiers. In each subfigure, the y-axis represents the classification accuracy.

As Fig. 5 shows, we can compare stock index prediction accuracy between different abscissa variables. Fig. 5(a) shows each classifier's accuracy on **DataC** using $\widehat{\mathbf{F}}_1$, $\widehat{\mathbf{F}}_2$, and $\widehat{\mathbf{F}}_3$, respectively. We obtain the highest accuracy by using $\widehat{\mathbf{F}}_3$, indicating that the "best" combination of TIs and TMIs performs better than using TIs or TMIs alone. Hence, we conclude that the proposed TMIs can improve the classifier's ability in predicting the stock index's direction. Fig. 5(b)

shows each classifier's accuracy on **DataC** using settings of $l \in \{5, 10, 20\}$. For the three different observation window sizes, the setting with $l = 20$ obtained the lowest overall classification accuracy, which may imply that the observation window cannot be too large for stock index prediction under our methods. Fig. 5(c) shows each classifier's accuracy on **DataC** using *SZ100*, *HS300*, and *SSZ380*. Clearly, our method on *SSZ380* performs the best and offers the highest prediction accuracy, whereas the accuracy on *HS300* is relatively low. Fig. 5(d) shows each classifier's accuracy on **DataC** using different classifiers. Here, the highest classification accuracy is 80.00 percent, obtained by `AdaBoost`, even though `C4.5` performs slightly better than `AdaBoost` overall; `SVM` and `MLP` perform the worst in stock index prediction. This finding indicates that C4.5 could be a better classifier for stock index prediction.

VI. Conclusion

In this paper, we combine TIs and TMIs to predict next-day market movements in three Chinese stock index datasets. We use six popular machine learning classifiers (`SVM`, `NB`, `KNN`, `C4.5`, `MLP`, and `AdaBoost`) and test their accuracy by applying two types of input indicators. The experiments reveal that the combination of TIs and TMIs can provide better performance and with a best 80% accuracy. We also compare the accuracy of combining TIs and TMIs versus using only TIs or TMIs, and our findings show that our method is more effective and results in higher prediction accuracy. However, going forward, we can further improve this study's prediction performance in three ways: by extracting more TMIs from stock networks, setting different observation windows (because they can affect prediction accuracy), and proposing an investment strategy based on this study's prediction outcomes for future research, practical use, and further validation.

Acknowledgment

This research was partially supported by the National Natural Science Foundation of China under Grant 71871109, Grant 71871233, Grant 71801123, and Grant 91646204, in part by the Beijing Natural Science Foundation under Grant 9182015, and in part by the Postgraduate Research & Practice Innovation Program of Jiangsu province of China under Grant KYCX18_1438.

References

[1] M. Qiu and Y. Song, "Predicting the direction of stock market index movement using an optimized artificial neural network model," *PLoS One*, vol. 11, no. 5, pp. e0 155 133(p1–p11), 2016.

[2] Y. Kara, M. A. Boyacioglu, and O. K. Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange," *Expert systems with Applications*, vol. 38, no. 5, pp. 5311–5319, 2011.

[3] M. Tumminello, T. D. Matteo, T. Aste, and R. N. Mantegna, "Correlation based networks of equity returns sampled at different time horizons," *The European Physical Journal B*, vol. 55, no. 2, pp. 209–217, 2006.

[4] L. Xia, D. You, X. Jiang, and Q. Guo, "Comparison between global financial crisis and local stock disaster on top of chinese stock network," *Physica A: Statistical Mechanics and its Applications*, vol. 490, pp. 222–230, 2018.

[5] X. J. Xu, K. Wang, L. Zhu, and L. J. Zhang, "Efficient construction of threshold networks of stock markets," *Physica A: Statistical Mechanics and its Applications*, vol. 509, pp. 1080–1086, 2018.

[6] S. Basak, S. Kar, S. Saha, L. Khaidem, and S. R. Dey, "Predicting the direction of stock market prices using tree-based classifiers," *North American Journal of Economics and Finance*, pp. 1–16, 2018.

[7] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.

[8] P. Ou and H. Wang, "Prediction of stock market index movement by ten data mining techniques," *Modern Applied Science*, vol. 3, no. 12, pp. 28–42, 2009.

[9] H. Liu and B. Song, "Stock trends forecasting by multi-layer stochastic ann bagging," in *International Conference on Tools with Artificial Intelligence*. IEEE, November 2017, Conference Proceedings, pp. 322–329.

[10] H. Zhou, Z. Deng, Y. Xia, and M. Fu, "A new sampling method in particle filter based on pearson correlation coefficient," *Neurocomputing*, vol. 216, pp. 208–215, 2016.

[11] X. Guo, H. Zhang, F. Jiang, and T. Tian, "Development of stock correlation network models using maximum likelihood method and stock big data," in *2018 IEEE International Conference on Big Data and Smart Computing*. IEEE, January 2018, Conference Proceedings, pp. 455–461.

[12] G.-J. Wang, C. Xie, and S. Chen, "Multiscale correlation networks analysis of the us stock market: a wavelet analysis," *Journal of Economic Interaction and Coordination*, vol. 12, no. 3, pp. 561–594, 2017.

[13] F. Pereira, T. Mitchell, and M. Botvinick, "Machine learning classifiers and fmri: a tutorial overview," *Neuroimage*, vol. 45, no. 1 Suppl, pp. S199–S209, 2009.

[14] J. Zhang and M. Small, "Complex network from pseudoperiodic time series: Topology versus dynamics," *Physical Review Letters*, vol. 96, no. 23, pp. 238 701(p1–p4), 2006.

[15] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, and J. C. Nuno, "From time series to complex networks: The visibility graph," *Proceedings of the National Academy of Sciences*, vol. 105, no. 13, pp. 4972–4975, 2008.

[16] M. Tumminello, T. Aste, T. D. Matteo, and R. N. Mantegna, "A tool for filtering information in complex systems," *Proceedings of the National Academy of Sciences*, vol. 102, no. 30, pp. 10 421–10 426, 2005.

[17] R. Dash and P. K. Dash, "A hybrid stock trading framework integrating technical analysis with machine learning techniques," *The Journal of Finance and Data Science*, vol. 2, no. 1, pp. 42–57, 2016.

[18] E. A. Gerlein, M. McGinnity, A. Belatreche, and S. Coleman, "Evaluating machine learning classification for financial trading: An empirical approach," *Expert Systems with Applications*, vol. 54, pp. 193–207, 2016.

[19] O. Hegazy, O. S. Soliman, and M. A. Salam, "A machine learning model for stock market prediction," *International Journal of Computer Science and Telecommunications*, vol. 4, no. 12, pp. 17–23, 2014.

[20] A. D. Ijegwa, V. O. Rebecca, F. Olusegun, and O. O. Isaac, "A predictive stock market technical analysis using fuzzy logic," *Computer and Information Science*, vol. 7, no. 3, 2014.

[21] J. P. Onnela, A. Chakraborti, K. Kaski, and J. Kertisz, "Dynamic asset trees and portfolio analysis," *The European Physical Journal B-Condensed Matter*, vol. 30, no. 3, pp. 285–288, 2002.

[22] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *International Journal of Science*, vol. 393, no. 6684, pp. 440–442, 1998.

[23] Z. Bu, H.-J. Li, C. Zhang, J. Cao, A. Li, and Y. Shi, "Graph k-means based on leader identification, dynamic game and opinion dynamics," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2019.

[24] J. Cao, Z. Bu, Y. Wang, H. Yang, J. Jiang, and H.-J. Li, "Detecting prosumer-community groups in smart grids from the multiagent perspective," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–13, 2019.

[25] Z. Bu, Y. Wang, H.-J. Li, J. Jiang, Z. Wu, and J. Cao, "Link prediction in temporal networks: Integrating survival analysis and game theory," *Information Sciences*, vol. 498, pp. 41–61, 2019.

[26] K. Z. Mao, "Orthogonal forward selection and backward elimination algorithms for feature subset selection," *IEEE Transactions on Systems*, vol. 34, no. 1, pp. 629–634, 2004.

View publication stats