Expert Systems With Applications 105 (2018) 159–173

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A hybrid supervised semi-supervised graph-based model to predict one-day ahead movement of global stock markets and commodity prices

Arash Negahdari Kia[a], Saman Haratizadeh[a,*], Saeed Bagheri Shouraki[b]

[a] *University of Tehran, Faculty of New Sciences and Technologies, North Kargar Street, 1439957131 Tehran, Iran*
[b] *Sharif University of Technology, Electrical Engineering Department, Azadi Ave., 1136511155 Tehran, Iran*

ARTICLE INFO

Article history:
Received 8 October 2017
Revised 20 March 2018
Accepted 21 March 2018
Available online 21 March 2018

Keywords:
Financial markets prediction
Hybrid machine learning models
Graph algorithms
Semi-supervised learning

ABSTRACT

Market prediction has been an important machine learning research topic in recent decades. A neglected issue in prediction is having a model that can simultaneously pay attention to the interaction of global markets along historical data of the target markets being predicted. As a solution, we present a hybrid supervised semi-supervised model called HyS3 for direction of movement prediction. The graph-based semi-supervised part of HyS3 models the markets global interactions through a network designed with a novel continuous Kruskal-based graph construction algorithm called ConKruG. The supervised part of the model injects results extracted from each market's historical data to the network whenever the hybrid model allows with an innovative conditional mechanism. The significance of higher prediction accuracy of HyS3 is comparing to other models is proved statistically against other models including supervised models and network-based semi-supervised predictions.

1. Introduction

Prediction of financial time series has been an interesting machine learning topic according to many surveys (Atsalakis & Valavanis, 2009; Rather, Sastry, & Agarwal, 2017; Sapankevych & Sankar, 2009; Soni, 2011). In this research, we will focus on predicting one-day ahead, falls and rises of financial time series that is referred to as direction of movement. The financial time series used in this study are some famous stock market indexes across the world, the gold spot price of 10 AM London and prices of different types of crude oil. There are complex interrelations among these financial markets. Some models are developed to show these relations in the field of econophysics (Gao, An, & Zhong, 2013; Liu & Tse, 2012; Mantegna & Stanley, 1999; Papana, Kyrtsou, Kugiumtzis, & Diks, 2017). They look at the global market environment as a complex system and model it using networks. Network modeling of relations among direction of movements of these markets showed to be useful for improving performance of prediction in the works of Park and Shin (2013) and Shin, Hou, Park, Park, and Choi (2013). We will provide a novel hybrid supervised semi-supervised model called HyS3 for market prediction using network modeling. A novel algorithm is also developed using a continuous Kruskal-based approach for making a graph called ConKruG to model complex markets interactions. In Table 1 the advantages of HyS3 to other models in the literature are presented briefly. Further descriptions and examples of existing researches related to the points mentioned in Table 1 are given in the following part of the introduction.

It can be assumed that direction of movement prediction is a binary classification problem. There are two major methods to classify unknown directions in machine learning literature: supervised and semi-supervised. In supervised learning, there are enough historical data and the aim of the model is to learn the past patterns to predict future directions. In semi-supervised learning, there are a few known market directions for markets that revealed their prices information. Graph-based semi-supervised learning[1] market prediction, models the interactions among markets with known or unknown direction with a graph. Zhu (2005) has explained graph-based semi-supervised models in his research. The known markets directions will be spread or propagated to markets with unknown directions. In this study, the earth is partitioned into four time zones in a way that for any given time, only markets in one time zone have revealed their close price information. Therefore, the direction of movement for them is known and the other zones have markets with unknown

* Corresponding author.
E-mail addresses: nkia.arash@ut.ac.ir (A.N. Kia), haratizadeh@ut.ac.ir (S. Haratizadeh), bagheri-s@sharif.edu (S.B. Shouraki).

[1] GSSL.

https://doi.org/10.1016/j.eswa.2018.03.037
0957-4174/© 2018 Elsevier Ltd. All rights reserved.

Table 1
Advantages of HyS3 to previous financial prediction researches.

	Some features of prediction models in previous researches	HyS3 advantages
1	Only paying attention to historical data of target market or only paying attention to effective factors data	Paying attention to both historical data of target market and interactions of other effective factors
2	Restricting effective factors in prediction of target market to a limited extent	The possibility of considering a large number of effective factors in prediction with network modeling
3	The necessity of building a model for each target market	A general model to predict a large number of markets at the same time
4	In the network-based prediction researches, the network is initially constructed without attention to prediction purpose then tested for prediction	In ConKruG, the network is constructed from the outset on the basis of prediction purpose
5	Actual daily prediction is neglected in many models for simplicity with trend prediction, smoothing the time series, change detection and noise reduction methods	HyS3 has a prediction for each single day for a large number of markets

directions that can be predicted through a GSSL method. HyS3 uses ConKruG algorithm to make the graph. HyS3 uses a novel approach to inject the information from a supervised model (SVM[2] in our case), to the ConKruG network.

Many researches in machine learning like Patel, Shah, Thakkar, and Kotecha (2015), Kia, Fathian, and Gholamian (2012), Kara, Boyacioglu, and Baykan (2011) and Yao and Tan (2000) tried to predict a stock index or commodity price simply by its historical data. Some other researches like Fathian and Kia (2012), Huang, Nakamori, and Wang (2005) and Thawornwong and Enke (2004) tried to improve the performance of financial predictions by using some external factors as inputs of their supervised models along with historical data of the target market. Park and Shin (2013) and Shin et al. (2013) used some stock prices data in South Korean market and commodity prices in a network-based semi-supervised model without paying attention to the historical data of the time series being predicted and they achieved better results compared to their rival models using only historical data of the target market. The question that arises here is when to use the past data of target markets and when to use the data from other markets to predict the future direction of a market. HyS3 model has a novel approach for deciding how and when to combine information from historical data of target markets and information achieved from global markets interactions.

Finding the external factor(s) that can help the prediction is sometimes a trial and error task. Sometimes having a prior knowledge of the research domain makes this task easier. Researchers choose some helping markets to add as input next to the historical data of the market being predicted and then test if these helping markets make the prediction performance increase (Jaruszewicz & Mańdziuk, 2004). Other times, researchers choose among a list of time series by the means of a feature reduction algorithm (Tsai & Hsiao, 2010). We show that by using a network structure of relations among external factors ,the model can use a large number of external factors according to their distance and position to the markets that are going to be predicted. The network can regulate the effect of external factors in prediction if made with a purposeful method like ConKruG.

As the surveys like Rather et al. (2017), Soni (2011) Atsalakis and Valavanis (2009) and Sapankevych and Sankar (2009) show, in most studies one or a short list of stock indexes or commodity price time series are selected to be predicted by a model. The lack of a general prediction model is seen in the literature. ConKruG network can model interactions among any number of markets for prediction with HyS3.

Using network structure for market prediction is a newly opened research topic and therefore there are not enough studies up to now. Kia, Haratizadeh, and Heshmati (2016), Park and Shin (2013) and Shin et al. (2013) constructed networks that seemed to be helpful in modeling the complex relations among movements of financial time series. They first assumed that a special distance function or a special rule derived from their datasets may be helpful in constructing a prediction network and then tested their assumption. In our approach, ConKruG designs the network gradually by adding any helpful link for the prediction aim.

Another aspect of the research, is the time period of prediction. Narayan, Sharma, and Thuraisamy (2015) discuss the importance of data frequency in sampling. Noisy nature of financial time series makes it difficult to predict direction for short period frequencies. Therefore many models try to predict a threshold of change in the direction (Homm & Breitung, 2012; Roy & Sarkar, 2013) or use longer time periods like week or month (Boyacioglu & Avci, 2010) or simply try to predict the trend by smoothing the time series with moving averages or other methods (Park & Shin, 2013). In this paper, we construct a model for predicting daily direction of movement for everyday in markets across the globe despite the difficulties of improving the accuracy in the noisy financial environment and short time period.

The structure of the paper is as follows: in the next section some preliminaries about the GSSL and SVM are explained. At the end of the section two, we discuss our most important rival models prediction algorithms in brief. The third section explains our hybrid proposed model, HyS3 in detail and describes the proposed novel graph construction algorithm, ConKruG. The final section, explains the model settings and parameters, presents the results and discusses their interpretation. The last section concludes and has some suggestions for further researches.

2. Preliminaries and related works

In our model, we modify the GSSL algorithm to inject the probabilities vector output from SVM. Therefore, a brief description of general forms of GSSL algorithms and SVM are presented. Then the works of researchers in Park and Shin (2013) and Shin et al. (2013) studies are explained to be compared with our method and results in later sections.

2.1. GSSL: graph-based semi-supervised learning

Semi-supervised learning aims to classify unlabeled data when there are a few labeled samples and many unlabeled ones. A category of famous methods in semi-supervised learning is GSSL. In GSSL, a network of relations is made among the data samples then the labels are propagated or spread through the network from labeled data to unlabeled ones.

Fig. 1 illustrates a sample undirected graph in which nodes with two circles have labels one or zero and the other nodes do not have labels. The first step in making the graph is to calculate the weight between the nodes. If considering *i* and *j* two nodes in Fig. 1, the weight between them is calculated through formula (1) (Zhu, Ghahramani, & Lafferty, 2003). The function *distance* (*i, j*)

[2] Support Vector Machine.

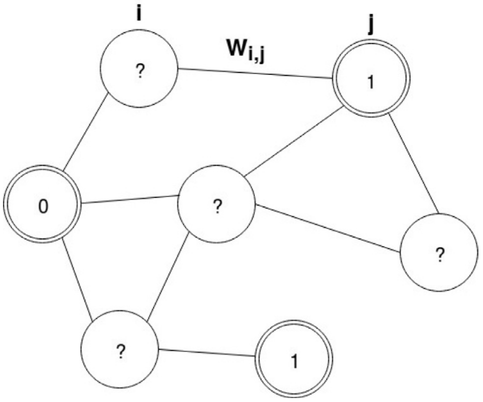


Fig. 1. Graph-based semi-supervised learning (GSSL). Nodes with two circles are for markets with known direction of movements while nodes with one circle are markets that will be predicted through GSSL.

in Eq. (1) should be chosen wisely according to the problem (Baghshah, Afsari, Shouraki, & Eslami, 2014). The σ is to adjust the weight, based on distance and proximity between two nodes in network. The other issue that should be taken into consideration is whether having a link between two nodes or not. Park and Shin (2013) and Shin et al. (2013) used KNN[3] to eliminate edges with smaller weights. This somehow can be seen as a noise reduction method.

Theoretically in semi-supervised learning, a regularization framework is made with three assumptions (Chapelle, Scholkopf, & Zien, 2009). The first assumption is smoothness. In paradigm of our problem it means that markets close to each other in the network are likely to have the same direction of movement. The second assumption is having clusters of the same direction in the network. This means markets in the same cluster in network tend to have the same direction. Final assumption is called manifold assumption that says the data lies in a space with different dimensions than their actual dimensions. The network constructed to show the relations among markets is used as a proxy to the manifold space. The importance of each of these assumptions in definition of regularization framework creates different graph-based semi-supervised methods. Two different types of GSSL are explained in Sections 2.1.1 and 2.1.2.

$$W_{i,j} = \begin{cases} e^{-\frac{distance(i,j)^2}{\sigma^2}} & \text{if } i \text{ is connected to } j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

2.1.1. Label propagation

Park and Shin (2013) and Shin et al. (2013) used the regularization framework in Eq. (2) to find unknown direction of time series in the network.

The function F is going to be minimized in regularization framework. L is the Laplacian matrix of the network that is calculated from the weight matrix, W and degree matrix, D in Eq. (2). Y is a $n \times 1$ vector of labels consisting of two attached parts of $Y_{labeled}$ and $Y_{unlabeled}$. The parameter μ is set from a search domain regarding the validation the data to have the best accuracy.

$$\begin{aligned} &min_F \quad (F-Y)^T(F-Y) + \mu F^T LF \\ &\quad L = D - W \\ &\quad D = \text{diagonal matrix}\left(\sum_j w_{i,j}\right) \end{aligned} \tag{2}$$

The solution to the framework is presented in Eq. (3). F is a $n \times 1$ vector with values between [0, 1]. With a threshold cut in vector F ($F_i > threshold : label = 1 \;\; else \;\; label = 0$), the unknown markets will have a class label, zero or one, showing fall or rise in the direction of movement.

$$F = (I + \mu L)^{-1} Y \tag{3}$$

2.1.2. Label spreading

Zhou, Bousquet, Lal, Weston, and Schölkopf (2004) presented a method for GSSL that was inspired by activation networks in psychology (Anderson, 2013; Shrager, Hogg, & Huberman, 1987). They called it label spreading. The optimization framework of label spreading that should be solved was different from other GSSL methods. Zhou et al. (2004) showed that label spreading achieves higher prediction performance comparing to other GSSL methods. First, a matrix named $Y_{initial}$ is constructed like Eq. (4). Assume l and u are, respectively, the number of nodes with known labels and the number of nodes with unknown labels. It is obvious that $n = l + u$ is the number of all nodes in the network. Y's dimension will be $n \times 2$ in binary classification, and $n \times c$ in a classification problem with c classes. The first l rows of the matrix $Y_{initial}$ show the class label or direction of movement in a day for a market with known direction. For example, if $Y_{initial_{i,0}} = 1$ and $Y_{initial_{i,1}} = 0$ then the class label for node i is zero. In our context it means fall in the direction of financial time series. The next u rows of the $Y_{initial}$ matrix are filled with zeros. It will be shown as a part of HyS3 in Section 3.3 that probabilities of being in a class can be injected to these rows, instead of zeros.

$$Y_{initial} = \begin{array}{c} \text{Fall Class} \quad \text{Rise Class} \\ \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{array}\right] \end{array} \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} \text{Known Labels} \\ \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} \text{Unknown Labels} \end{array} \tag{4}$$

In Eq. (5), label spreading algorithm is presented in a recursive process. First, a graph named S that is a normalized form of W, is made from the weight graph. The parameter α is used to determine the importance of the initial labels in the network and in recursive process of finding solution. Zhou et al. (2004) set the α to 0.99 and we do the same in this research. D is the sum of degrees diagonal matrix like in Eq.(2). Zhou et al. (2004) proved the convergence of recursive formula in Eq. (5). The proof of convergence in our proposed hybrid model is presented in Section 3.3.

$$\begin{aligned} S &= D^{-1/2} W D^{-1/2} \\ Y_t &= \alpha S Y_{t-1} + (1-\alpha) Y_{initial} \end{aligned} \tag{5}$$

The solution to Eq. (5) is in Eq. (6). Zhou et al. (2004) proved $(1 - \alpha S)$ is invertible. Regarding the solution in Eq. (6), label of node i will be the index with the maximum value in row i of matrix $Y_{solution}$.

$$\begin{aligned} Y_{solution} &= (1-\alpha)(1-\alpha S)^{-1} Y_{initial} \\ Label_i &= argmax_j \; Y_{solution_{i,j}}, \qquad j \leq total\ classes \end{aligned} \tag{6}$$

2.2. Support Vector Machines

In SVM, a linear decision hyperplane is found to divide the sample space to two different classes (Cortes & Vapnik, 1995). A kernel is used to transfer samples from their space to another space with different dimensions where they can be separated linearly. Some samples are selected as support vectors that make linear margins between two classes. A regularization parameter is selected

[3] K-Nearest Neighbor.

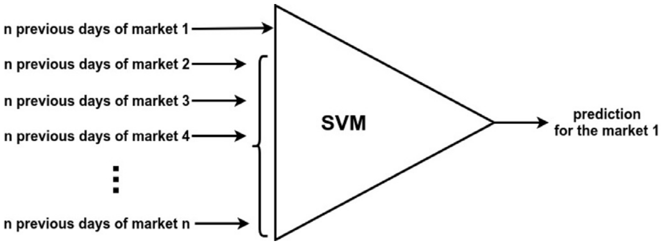

n previous days of market 1
n previous days of market 2
n previous days of market 3
n previous days of market 4
n previous days of market n
SVM
prediction for the market 1

Fig. 2. SVM model with external features.

to avoid overfitting and tolerate a limit of error in classification (Cawley & Talbot, 2010).

The Eq. (7) is the optimization problem that can be solved with quadratic programming. The parameter C is the regularization factor. The parameters α_i are the support vectors. b is the bias of decision hyperplane and $K(x, x_i)$ is the kernel function.

$$Maximize \sum_{i=1}^{n} \alpha_i - 1/2 \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j . y_i y_j . K(x_i, x_j)$$

$$s.t. : \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$0 \le \alpha_i \le C, \quad i = 1, \dots, n \tag{7}$$

In Eq. (8), $f(x)$ is the decision function that is calculated to find the class labels. kernel function used in our study is RBF[4] that is presented in Eq. (9). The best value for parameter γ in RBF kernel will be selected regarding the validation set of the datasets.

$$f(x) = \sum_{i=1}^{n} y_i \alpha_i K(x, x_i) + b \gtrless 0 \tag{8}$$

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \tag{9}$$

The SVM classifier's output can be presented as probability of membership in a class. The logit function used in Eq. (10) gets the result of SVM decision function $f(x)$ in Eq. (8) and transfers the result into probability space (Platt et al., 1999).

$$P(Class|Input) = P(y = 1|x) = \frac{1}{1 + e^{-f(x)}}$$

$$P(y = 0|x) = 1 - P(y = 1|x) \tag{10}$$

The SVM can also use external helping features as its input to improve its prediction performance. In this study, we fed all the data that is used in our network to a SVM model and calculated the prediction performance of it. The performance of simple SVM and SVM with external features is then compared to our proposed model. The general schema of the SVM model with external features is presented in Fig. 2.

2.3. Graph-based semi-supervised market prediction

Park and Shin (2013) used 200 stock price indexes in South Korean stock market. Shin et al. (2013) used different oil prices and exchange rates to make a network for prediction of just one node, the WTI[5] price. They used a vector of technical indicators for each day of each market and used Euclidean distance function for weight calculation in Eq. (1). Finally KNN method was used to eliminate edges with small weights in network. For more explanation, if two vectors of technical indicators for two stocks of s_1 and s_2 are $s_1 = [f_{1,1}, f_{1,2}, \dots, f_{1,n}]$ and $s_2 = [f_{2,1}, f_{2,2}, \dots, f_{2,n}]$ and $f_{i,j}$ refers to technical indicator j of market i then the distance is the norm-2 of s_1 and s_2 and will be calculated as $d = \sqrt{\sum_j^n (f_{1,j} - f_{2,j})^2}$. The weight is then calculated using $exp(-d/\sigma^2)$ where σ is just a parameter that can be set regarding the performance of the algorithm in validation set. The KNN method first sorts all the weights in a descending order and then keeps the first biggest K weights in the network and eliminates others. The justification of filtering a network and eliminating some edges with smaller weights is noise reduction. General schema of their prediction model is presented in Fig. 3. Each day all the stocks that have revealed their data are used to calculate their technical indicator vectors, a vector of some technical indicators used in the works of Park and Shin (2013) and Shin et al. (2013). Then the network is made by KNN and label propagation method is used to predict the unknown labels or markets.

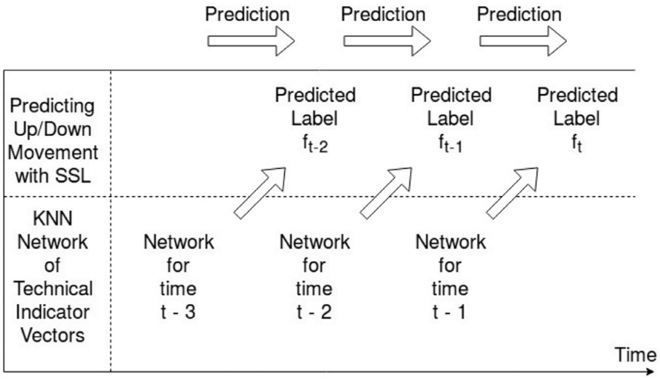

Prediction
Prediction
Prediction
Predicting Up/Down Movement with SSL
Predicted Label f_{t-2}
Predicted Label f_{t-1}
Predicted Label f_t
KNN Network of Technical Indicator Vectors
Network for time t - 3
Network for time t - 2
Network for time t - 1
Time

Fig. 3. Park and Shin model schema for up/down movements prediction. SSL: semi-supervised learning.

Park and Shin (2013) used 3-days moving average to smooth the time series and calculate the direction of movement. Shin et al. (2013) used 5-moving average in monthly data. The method for calculating the direction label in their works is given in Eq. (11). x_t refers to the actual value of stock x in day t and MA refers to n-moving average of stock x from day t to day $t - n + 1$. For example $MA_3(x_{10})$ will be $(x_{10} + x_9 + x_8)/3$. In our research, we predict the actual daily direction of movement without smoothing the data with moving average.

$$label_t = sign(x_t - MA_n(x_t)) \qquad n = 3 \ or \ 5 \tag{11}$$

3. HyS3 model

An explanation of HyS3, the proposed hybrid supervised semi-supervised prediction is presented in this section. Fig. 4 presents an overview of our approach. The sub-systems of the model are numbered for better explanation in text.

The first block of the model is data preparation block. It prepares data as described in Section 3.1 and divides the dataset. Some parts of the model like ConKruG only use train and validation datasets. It is worth mentioning that prediction models (when trained or used), must not see the future data in anyway and in any part of their processes. The data flow of the model strongly abandons seeing the future data before prediction. The blocks that are fed with the test set, only see the part of the test that its time has passed. In other words, blocks seven and eight are streamed with only the past data of the test set and not the whole set.

Block two of the HyS3 constructs a network for graph-based semi-supervised prediction. This sub-process named ConKruG algorithm is explained in Section 3.2 with details. We show in Section 4.3 that this network provides the best prediction accuracy results when used in GSSL comparing to networks in the past re-

[4] Radial Basis Function.

[5] West Texas Intermediate crude oil.

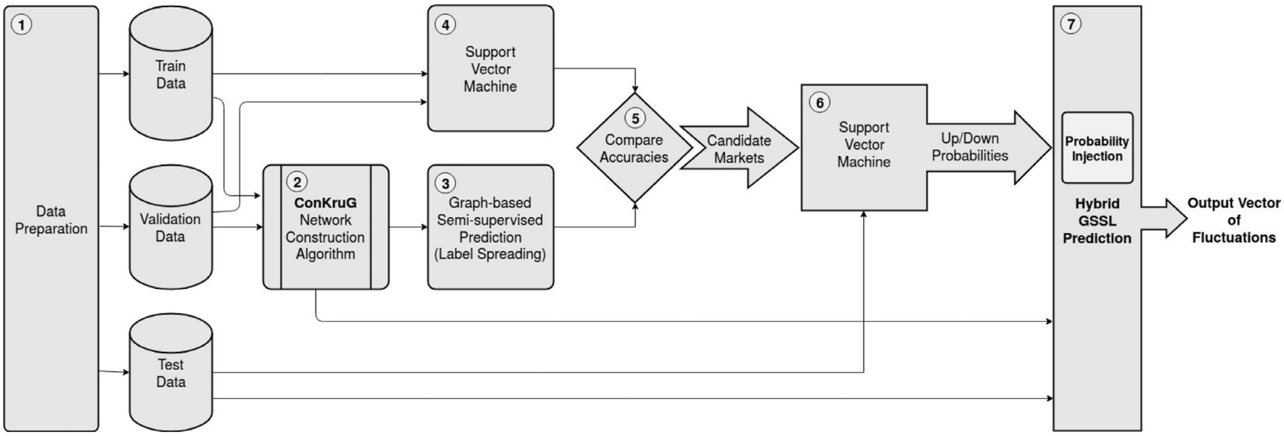


Fig. 4. **HyS3**, hybrid supervised semi-supervised proposed model schema.

searches. This network will be used in other modules of the HyS3 model.

Blocks three and four of the HyS3, provide prediction accuracy results of respectively a semi-supervised model with ConKruG network and a supervised model. The reason for acquiring these accuracy results is that by comparing their results we will find out in which markets the supervised model has done a better job. For any markets that supervised prediction has better accuracy results we label that market as a candidate for probability injection in the hybrid prediction. But if GSSL with ConKruG network provides better accuracy in validation set comparing to supervised model, we do not inject the probabilities extracted from supervised method in the initial phase of the market labeling in block seven. Fig. 4 shows that both the supervised and semi-supervised prediction blocks of three and four must be finished before the comparison starts in block five of HyS3.

We use SVM for our supervised prediction model in HyS3. Before choosing SVM for the supervised component of our model, we tried different single and ensemble supervised models and SVM with RBF kernel outperformed all of them. The next best model after SVM was random forest that is a state of the art ensemble model. The results of random forest are also presented in the results section of the paper. Ho (1998) explained random forest in detail. The benefits of using SVM to other supervised models are explained in Section 2.2 and in works of Lin, Yeh, Lee et al. (2013), Sui, Hu, Yu, Xie, and Qi (2007), Shin, Lee, and Kim (2005) and Kim (2003). Some advantages of SVM are avoiding the overfitting with regularization parameters, not being trapped in local optima solutions, and kernel trick which makes it potentially a good choice for making expert systems in different eras (Cawley & Talbot, 2010). The semi-supervised prediction block or block four also uses label spreading algorithm explained in Section 2.1.2 due to its benefits to other GSSL models (Zhou et al., 2004).

After finding the candidate markets that have better accuracies with SVM in comparison block or block five, the test begins. In block six the membership probability of being in a class of rise or fall is calculated for candidate markets with the seen test data up to the status time. For the markets that are not in the candidate set no probability will be calculated and injected in block seven. The hybrid prediction with the probability injection part is the heart of HyS3 that is explained with details in Section 3.3. The innovation of the HyS3 is injecting the candidate markets probabilities into the rows of an initial state matrix that is used in the formulation of label spreading algorithm. The following parts of this section, explain dataset and data preparation, ConKruG algorithm and hybrid prediction with injection or the first, third and seventh block of the HyS3 model schema in more detail.

3.1. Data gathering and preparation

The dataset used in this study is a collection of some famous stock indexes across the globe along with London 10 AM gold price and three types of crude oil price data that are collected on a daily basis. Information like total time series in each time zone and their types are presented in Table 2. The dataset is presented in Kia (2018). Some other detailed information about the dataset, its resources and the abbreviations used in figures and graphs of this research are presented in Appendix Table A.6. The websites that the data are gathered from is mentioned in the caption of Table A.6 in Appendix A. The historical range of dataset is from 17th of September 2007 to 4th of June 2015 consisting of 2014 daily samples excluding holidays for 36 time series.

More statistical information like mean and standard deviation of each market, the number of up and down classes in each time series and the whole dataset, and the imbalance ratios are presented in Appendix Table B.7. Imbalance ratio or number of majority class samples divided to number of minority class samples, $IR = (\frac{\#majority\ class}{\#minorityclass})$ shows that our dataset can be considered balanced according to Fernández, García, del Jesus, and Herrera (2008). They divided the imbalanced datasets into three groups of low, medium and high imbalance. Our dataset has imbalance ratio less than 1.5 and will not belong to any of imbalanced dataset groups according to their study. Also the cost of misclassification of up or down classes in our forecasting is not different for us. These information are presented to validate using directional accuracy when evaluating the prediction model. The only imbalanced market in our dataset is TEPIX that belongs to low imbalance group. In Section 4.3 we will see that our model does not claim to be better in imbalanced data of TEPIX.

The data is converted to the percentage of change with $\frac{x_{n+1}-x_n}{x_n}$ as the direction of movement. By using this change percentage, and with a less/greater comparison to one, the class labels are also calculated. In supervised part of the model where SVM is used, the past data of n-days before are used as input of model. We call it n-day delay input.

An important issue that should be mentioned is the time zones of the time series in the dataset. The change in calendar date is taking place continuously in a 24-h period on Earth. This requires attention in correlation-based calculations in prediction researches. Therefore the earth is partitioned into four time zones of America, Europe, Central Asia, and Australia and far east, where each zone's close prices of stock indexes or commodity prices cannot see other zones close prices. Four different zones partitioned like Fig. 5. We assume that each zone's prices are fully revealed then the correlation calculations are done. The important matter is that we predict

Table 2
Markets used in the research and their types in different time zones.

Type\Region	Time Zone 1 America (Known Zone)	Time Zone 2 Europe and Africa	Time Zone 3 Middle East, Central Asia and some parts of Russia	Time Zone 4 Far East and Australian	Sum
Stock indexes	8	13	4	7	33
Different types of oil	1	2			3
Gold		1			1
Sum	9	16	4	7	36

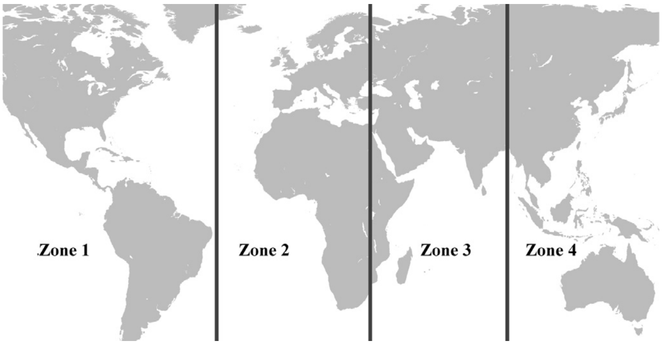


Fig. 5. Different time zones for partitioning markets.In this research, markets in the zone one have known up/down directions of movement and spread their direction labels to markets in other zones.

all the time zones with America continent's data. This data of zone one when revealed in any day, do not see other zones future data in calculations. If we want to calculate a one-day lagged correlation between two time series in two different time zones, the zones are important. As an example, for calculating one-day lagged correlation between ASX in Australia (zone 4), and one-day ahead of NYSE in America (zone 1), we must use the same calendar day for both of these markets, but for the reverse lagged correlation, we must use NYSE time series data of day t and ASX time series for day $t+1$. Formulations of lagged cross correlation are presented in Section 3.2 where network construction is explained.

Finally, the division ratio among train, validation, and test data in our dataset are respectively [0.7, 0.2, 0.1] with [1409, 402, 202] daily samples for each set without considering the holidays and repeating the data for rare cases of missing values. By summing the days for three sets of train, validation and test we find it one day less than the whole dataset. It is due to the way we calculated the change percentage.

3.2. ConKruG: network construction algorithm

We designed an algorithm for constructing a network for the purpose of direction prediction. The pseudo-code of the algorithm is presented in this paper. ConKruG (Continuous Kruskal-based Graph) algorithm uses train and validation sets of the time series regarding their respective time zones and constructs a network to be used in GSSL.

This algorithm is provided by us to make a network to be used in graph-based semi-supervised learning algorithm. The graph-based semi-supervised algorithm is dependent on the graph it uses. The better the graph or network shows the up/down movement relation among markets, the better the prediction will be. ConKruG uses the maximum spanning tree of the correlation network of markets as its base structure and then adds other edges to this tree if they help improving the accuracy of the prediction for the whole model.

Pseudo-code of algorithm ConKruG

Algorithm: Continuous Kruskal-based Graph Constructor (ConKruG).

Input: Train and validation dataset, Time zones of markets and commodity prices
Output: graph $G(V, E)$

```
$G(V,E)$ = symmetric delayed correlational graph of markets in **Train dataset** with respect of **Time zones of markets**
$MST(V',E')$ = maximumSpanningTree($G$)
  //It is obvious that $V' \subseteq V$ and $E' \subseteq E$
//Error of GSSL prediction on **Train** and **Validation** datasets in MST
//Index zero means that this values are for the first stage of the graph construction
$error_{train,0}$ = calculateError(MST, Train)
$error_{validation,0}$ = calculateError(MST, Validation)
$EL = E - E'$ //EL: Edges left to be considered
$SEL = sort(EL, descending)$ //Sort all the edges left by weight in a descending order
$i = 0$ //Variable i is for counting added edges
graphList = [ ]//List for storing constructed graphs
validationErrors = [ ]//List for storing errors of constructed graphs on validations set
for $e(m_i, m_j)$ in $SEL$: //$e$ is the edge between markets $m_i$ and $m_j$
    //$m_i \in V'$ and $m_j \notin V'$
    $i = i + 1$
    $E'' = \{e\} \cup E'$ , $V'' = \{m_j\} \cup V'$
    $error_{train,i}$ = calculateError($G''(V'', E'')$, Train)
    $error_{validation,i}$ = calculateError($G''(V'', E'')$, Validation)
    if $error_{train,i} \leqslant error_{train,i-1}$:
        graphList.append($G''$)
        validationErrors.append($error_{validation,i}$)
    else:
        $i = i - 1$
        $E'' = E' - \{e\}$ , $V'' = V' - \{m_j\}$
$r = argmin(validationErrors)$
return graphList($r$)
```

The reason for naming the algorithm ConKruG (Continuous Kruskal-based Graph) is that first, a maximum spanning tree is created with the famous Kruskal (1956) method, but the algorithm does not stop at this moment and edge adding continues. By adding the first edge in a weight descending order like Kruskal algorithm to the maximum spanning tree, we will have a graph with cycle. The maximum spanning tree will be the base structure of our network and has two characteristics. First it makes a connected graph of the markets which is necessary for our GSSL predictive method. Secondly, it includes the edges with highest weights meaning the most important relationships of rise/fall between financial markets (Mantegna, 1999). For finding the weights, one-day lagged cross correlation is calculated as in Eq.(12) (Kullmann, Kertész, & Kaski, 2002), then the distance is calculated as in Eq. (13) (Mantegna, 1999). In Eq. (12), C stands for cross-correlation, X and Y for the markets time series, and τ for the lag which in our case is one day. As explained in Sections 2.1.1 and 2.1.2, GSSL works with undirected networks and it is obvious that generally $C_{X,Y}(\tau) \neq C_{Y,X}(\tau)$. Therefore there should be a symmetric function to calculate the final distance between two nodes. The geometric mean of asymmetric distances is used as in Eq. (14) between each two markets. Geometric mean increases the effect of small lagged correlations in a way that the symmetric distance becomes small. It is a strict method for weighting the edges that have only a one-way big lagged correlation. This helps us only have high weights for edges that are sure to have a two-way high or medium lagged correlation connection between their markets.

ConKruG continues as all the edges are sorted weight descending and respectively are added to the graph. Each time one of the edges is added to the graph, ConKruG immediately checks whether the new graph does not decrease the prediction accuracy in train set. If the answer is yes, the edge remains in the graph. This is a greedy approach to find the best graph for GSSL prediction purpose (Cormen, 2009). Each time after adding an edge, both the accuracies in train and validation sets are stored in a data structure. If the condition of adding an edge (not decreasing the train accuracy) satisfies, then the evolving network up to that stage, is also stored in another data structure. Finally in ConKruG, the corresponding network with the highest accuracy in the validation set is considered as the output of the algorithm. For avoiding the overfitting, we look for the highest accuracy in the validation set. Fig. 6 shows first steps of edge adding in the ConKruG algorithm and the final result produced by the algorithm.

$$C_{X,Y}(\tau) = \frac{E[(X_t - \mu_X)(Y_{t+\tau} - \mu_Y)]}{\sigma_X \sigma_Y} \tag{12}$$

Our dataset is a collection of time series for most famous markets in the world. But anyone can use the ConKruG algorithm in different datasets of some time series in other fields of knowledge that graph-based semi-supervised prediction seems useful.

3.3. Hybrid prediction with probability injection

The novel method for combining results of supervised SVM method and semi-supervised graph-based label spreading algorithm comes in the label initialization phase for the network. As described in Section 2.1.2 the $Y_{initial}$ matrix is organized in a manner that all first l rows that represent the markets with known directions are filled with their corresponding class label and the other u rows for unknown market nodes are filled with zeros. In our proposed method we inject the probabilities came from SVM supervised prediction to these u rows, if the market is in the candidate set. The candidate set of markets as mentioned in Section 3 are those markets that have been shown higher accuracy results for supervised prediction comparing GSSL in their validation data. The initial labels matrix made by this injection method is presented in (15).

$$Y_{initial} = \begin{array}{c} \begin{array}{ll} \qquad\text{Fall Class} & \qquad\qquad\text{Rise Class} \end{array} \\ \left[\begin{array}{ll} 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 1 & 0 \\ candidate : probability(Rise) & probability(Fall) \\ 0 & 0 \\ 0 & 0 \\ candidate : probability(Rise) & probability(Fall) \\ candidate : probability(Rise) & probability(Fall) \\ \vdots & \vdots \\ 0 & 0 \\ candidate : probability(Rise) & probability(Fall) \end{array}\right] \end{array} \begin{array}{l} \\ \left.\vphantom{\begin{array}{l}1\\0\\ \vdots\\1\end{array}}\right\}\text{Known Labels} \\ \left.\vphantom{\begin{array}{l}c\\0\\0\\c\\c\\ \vdots\\0\\c\end{array}}\right\}\text{Unknown Labels} \end{array} \tag{15}$$

$$distance(X, Y) = \sqrt{2(1 - C_{X,Y}(\tau))} \tag{13}$$

$$symmetric\ distance(X, Y) = \sqrt{distance(X, Y).distance(Y, X)} \tag{14}$$

The prediction accuracies are shown in Fig. 7 after adding the edges to the graph in each step of the algorithm on all three sets of train, validation and test. If considering that there are n markets time series in the dataset, then there are at least $n - 1$ edges in the ConKruG network which is the maximum spanning tree (MST) of the complete correlation-based network of the time series in the dataset. Therefore the first added edge in the Fig. 7 is the nth edge in the output network of the ConKruG algorithm. The reason for showing the test set results is merely to demonstrate the process of accuracy change in steps of the ConKruG. As the description and pseudo-code of the algorithm show, none of the steps in the algorithm imply any use of the test set data.

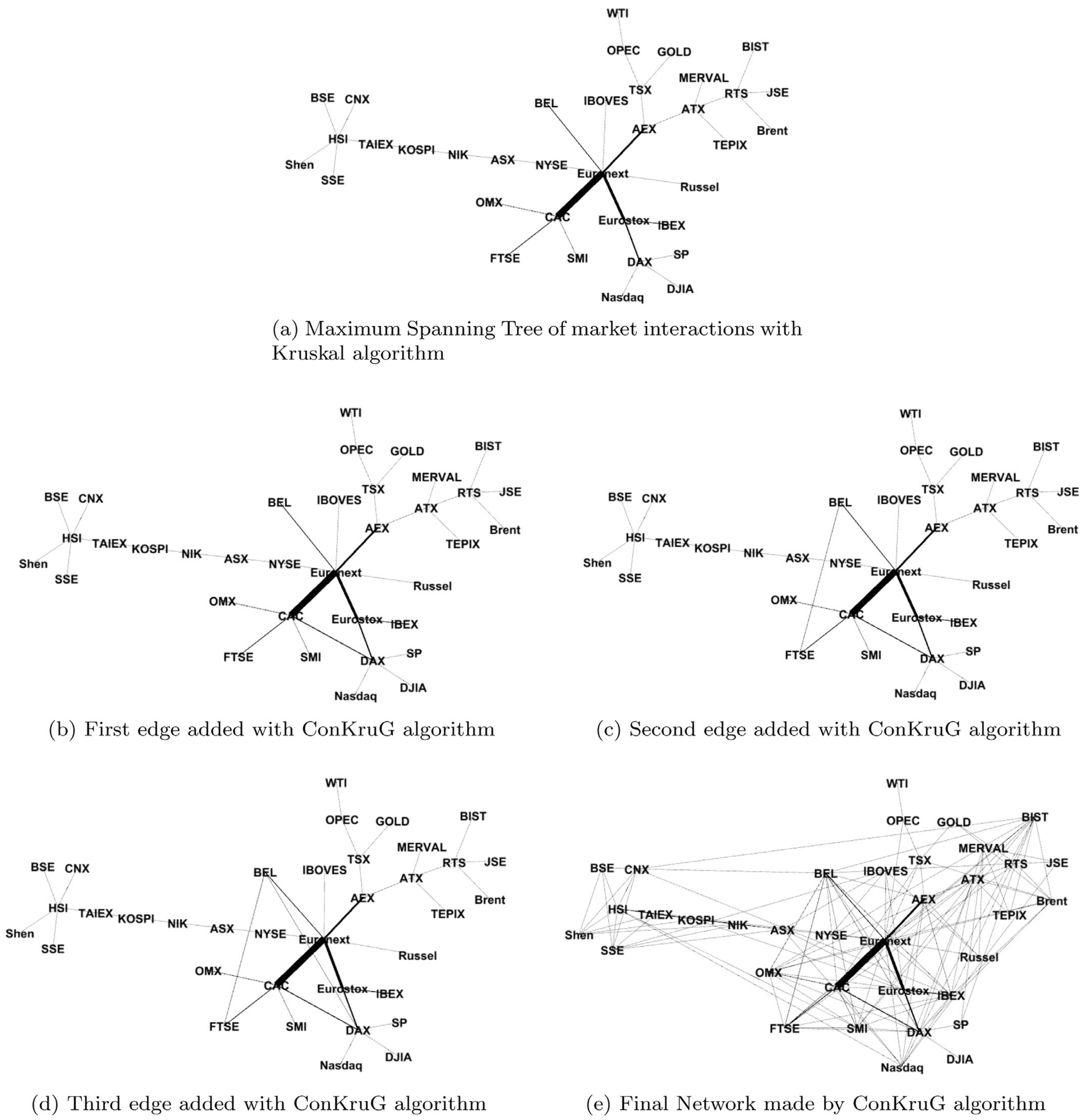
The decreasing nature of the error (increasing of the accuracy) in Fig. 7 in train set is due to the way ConKruG orders the edges to check if they can be added to the MST. The edges are sorted by their weights in the complete correlation-based graph of the markets time series dataset decreasingly. The weights show the similarity between two markets up and down movements with lag of one day. Therefor more effective edges in the accuracy are checked first. If the first edges satisfy the ConKruG condition, they will reduce the prediction error more than the last edges. But it should be mentioned that no edge will be added unless it has positive effect on the prediction performance.

The proof of the convergence for recursive phases of label spreading when using the new probability injected initial matrix is like the proof with previous initial vector in works of Zhou et al. (2004). The proof that the change of initial vector does not change the convergence of the algorithm is presented in (16). After iterating the recursive process of label spreading in (5) we will have the Y_t in (16). Eq. (16) shows that the limits of two parts of the summation in parentheses tend to the specified values, and the changed value of $Y_{initial}$ after injecting the probabilities, does not affect the convergence of the result. Fig. 8 shows the labels spread with HyS3 for a sample day in test set.

$$\begin{aligned} Y_t &= ((\alpha S)^{t-1} + (1 - \alpha)\sum_{i=0}^{t-1}(\alpha S)^i)Y_{initial} \\ \lim_{t\to\infty}(\alpha S)^{t-1} &= 0 \\ \lim_{t\to\infty}\sum_{i=0}^{t-1}(\alpha S)^i &= (I - \alpha S)^{-1} \end{aligned} \tag{16}$$

3.4. Complexity analysis of HyS3

Training computational complexity of the model is calculated for modules 1–5 of Fig. 4 and the test computational complexity is calculated for modules 6 and 7. The supervised prediction module 4 has $O(supervised)$ that is in our case $O(SVM)$. We used the sklearn python package for implementing SVM that uses libsvm algorithm. The order of libsvm implementation is between $O(n_{features} \times n_{samples}^2)$ to $O(n_{features} \times n_{samples}^3)$ (Chang & Lin, 2011).



(a) Maximum Spanning Tree of market interactions with Kruskal algorithm

(b) First edge added with ConKruG algorithm

(c) Second edge added with ConKruG algorithm

(d) Third edge added with ConKruG algorithm

(e) Final Network made by ConKruG algorithm

Fig. 6. Some steps of network construction by ConKruG algorithm and the final result.

Let us assume that we have m markets and d days (samples). We can say that the worst case of SVM time complexity will be $O(md^3)$. In module 2, the ConKruG algorithm has two parts of a Kruskal algorithm running to find a spanning tree with complexity of $O(ElogV)$ with E and V being the number of edges and nodes in the network (Kruskal, 1956) and the other part of checking the significance of edges and choosing if they should be in the final network. Our network at the initial stage of the ConKruG is a complete network and therefore the Kruskal phase time order will become $O(m^2 \log m)$. After building the spanning tree the significance of all the other edges of the complete network are examined with GSSL algorithm of label spreading in Zhou et al. (2004). The complexity order of GSSL label spreading algorithm in module 3, is $O(t \times E)$ that t is the number of iterations in Zhou et al. (2004) algorithm that the labels converge and E is as before, the number of edges. Assuming the worst case of having all the edges in the ConKruG network the order of GSSL algorithm will become $O(tm^2)$ and the order of second phase in ConKruG will become $O(tm^2 \times m^2) = O(tm^4)$. The parameter t or the number of iterations for convergence is depended on the structure of the graph and definition of convergence of results. Having all these orders and by considering the HyS3 scheme in Fig. 4 we can finally say that the worst case training complexity time order of our algorithm will be $O(m^3d^3 + tm^4)$. Accordingly, the test phase of the

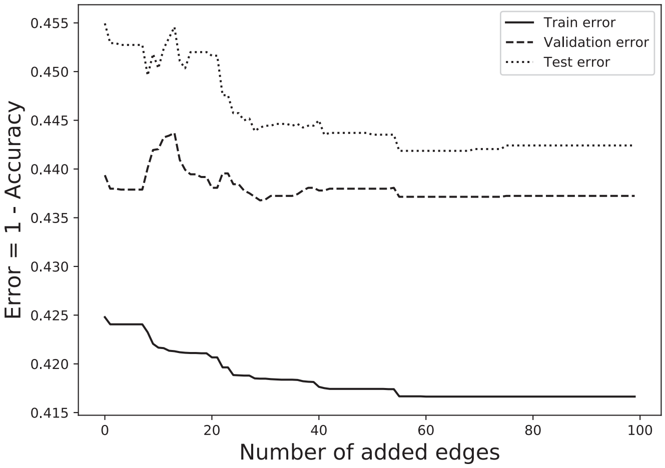


Fig. 7. Errors in different stages of edge adding in ConKruG algorithm in different datasets. Errors in test set are merely shown for reader's attention and it is not used in any part of the ConKruG algorithm.

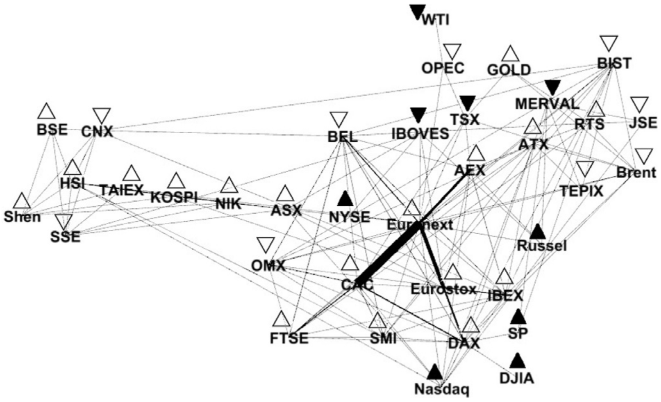


Fig. 8. Up and down labels after prediction with HyS3 on ConKruG network for a simple day. Solid triangles are for known zone (America continent) and hollow triangles for unknown zones. The direction of triangles shows rise or fall of the market directions.

algorithm, or modules 6 and 7 together will have the time complexity of $O(md^3 + tm^2)$. In the scale of financial time series data the number of daily samples cannot exceed a certain amount. With the computational power of computers today the number of iterations will not be the important issue in the complexity order of the algorithm. Therefore we can assume that the main part of the time complexity orders will be number of markets or nodes (sources of information) in the network. So a good and more simple approximation for the train and test phases of the HyS3 model can be $O(m^4)$ and $O(m^2)$. The time for running the algorithm with our dataset in a machine with Intel core i7-6500U CPU with 4 MB of cache and 8 GB of DDR4 RAM was 32 min and 44 s.

4. Experimental settings and results

The evaluation results of our proposed model is the accuracy of direction prediction. The number of two classes of rise and fall movements are balanced with an acceptable approximation for stock indexes and commodity prices. The statistical significance of the higher accuracy of our model is presented by an appropriate statistical test.

The codes of ConKruG algorithm and HyS3 model are in github repository (Kia, Haratizadeh, & Shouraki, 2018) with LGPL v3.0 license. Other models that led to the production of our final HyS3-ConKruG model are also presented in the github repository codes.

All the model construction and codings have been done with python 3 by using scikit-learn package for machine learning algorithms, networkx package for graph data structure and numpy and pandas packages for matrix calculations and time series modification. The network illustrations in the paper are done with the help of both networkx in python and gephi software. The programming language used and its packages mentioned and gephi have GNU public license and BSD licenses for free usage.

4.1. Experimental settings

Model parameters are founded solely regarding the validation set accuracy and by searching in a domain. All the parameters in Table 3 are explained in Section 2. The parameter α is fixed in the work of Zhou et al. (2004) and in our work. The parameters σ, μ and K are searched in domains like the works of Park and Shin (2013) and Shin et al. (2013) because we want to have a fair comparison between the HyS3 results and their graph-based methods. Number of trees that are used in the random forest model is set in parameter $\#estimators$. The parameters of SVM component in HyS3 are chosen from their respective search domain shown in Table 3 to have the best mean validation set accuracy for all the time series in dataset.

4.2. Evaluation method

The Eq. (17) shows the accuracy calculation formula (Fawcett, 2006). In our study, positive classes or classes with label one mean up direction of movement in time series and the same is applied to the negative classes meaning, classes with zero and down direction.

HyS3 is tested for all the samples that is all the time series multiplied all the days in the test set. We wanted to compare our model with a number of rival models. Therefore we wanted to have the number of test samples alike in all comparisons. In the supervised models and hybrid models with supervised component, since there is a daily delay parameter discussed in Section 2.2, a 7 day decrease for test samples will be seen. This is equal to best delay parameter founded for SVM model.

Total number of samples $(day \times (market\, or\, commodity))$ is 195×27 that equals 5265 individual unpredicted instances of $(day, time\, series)$. 195 has come from decreasing 7 daily delay parameter from 202 days in test set. There are 27 markets for prediction that has come from decreasing 9 known zone markets and commodity prices from 36 total markets and commodity prices in our dataset. Predictions result for each individual unpredicted instances of $(day, time\, series)$ are calculated and the average of correctly predicted (true positives plus true negatives) is reported in Section 4.3 as the accuracy. This is the same as calculating with Eq. (17). The average prediction of HyS3 is higher than the rival models as shown in Table 4. But it is also shown that this improvement in accuracy is statistically significant.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{\#right\ preditcted\ directions}{\#all\ predicted\ directions} \tag{17}$$

We used the unpaired t-test for showing the statistical significance of our results. Unpaired t-test is more conservative method than paired t-test due to not having the assumption of analyzing the same samples in different times (Rice, 2006). In different models the features used for input samples are different because of their different nature. Some used a delayed input (1-day to n-days before as feature vector) and some only used the past day. Unpaired t-test is a more rigorous method for our hypothesis testing than the paired one. In unpaired t-test t and df are calculated with

Table 3
Search domains for different parameters of models being compared. These parameters are set regarding the validation data.

Parameter	Model(s)	Search domain
α	All graph-based models except Park and Shin label propagation method	0.99 according to Zhou et al. (2004)
σ	All graph-based models except complete graph with all weights = 1	$\{0.1, 0.2, 0.3, \ldots, 3.0\}$
μ	Park and Shin model with label propagation	{0.01, 0.1, 0.3, 0.5, 0.7, 1, 10, 100} (Park & Shin, 2013)
K	For K-NN in all Park and Shin models	{2, 3, 4, 5} (Park & Shin, 2013)
C misclassification trade-off of SVM	HyS3 and SVM	$\{10^n \mid n \in \{-3, \ldots, 5\}\}$
γ RBF kernel parameter	HyS3 and SVM	$\{10^n \mid n \in \{-3, \ldots, 5\}\}$
Delay	HyS3 and SVM	$\{1, 2, 3, \ldots, 30\}$
#estimators	Random forest	$\{10, 100, 1000, 10,000\}$

Table 4
Prediction accuracies and statistical significance of the difference among them in different models.

Model	Mean accuracy (%)	Variance of accuracy (%)	Parameters	*p*-value Against Park and Shin model with label propagation	*p*-value Against Park and Shin model with label spreading
Park and Shin with label propagation	51.29	24.98	$K = 2, \mu = 10$	1	0.0005
Park and Shin with label spreading	54.65	24.78	$K = 5$	0.0005	1
SVM as a stand-alone supervised model	51.24	24.99	$Delay = 7, C = 10,000$ $\gamma = 0.01$	0.9465	Less than 0.0001
Random forest as a state of the art ensemble supervised model	51.21	24.98	$\#estimators = 100$	0.8695	Less than 0.0001
SVM with all other markets as helping features	54.25	24.82	$Delay = 7, C = 10,000$ $\gamma = 0.01$	Less than 0.0001	0.4079
Complete network all weights = 1 with label spreading	54.38	24.81	–	0.00015	0.7718
Complete correlation-based network with label spreading	55.13	24.74	$\sigma = 0.4$	Less than 0.0001	0.625
Maximal Spanning Tree with label spreading	54.51	24.80	$\sigma = 0.4$	0.0009	0.8752
ConKruG network with label spreading	55.76	24.67	$\sigma = 0.4$	Less than 0.0001	0.2562
Hybrid model with complete correlation-based network with label spreading	55.87	24.66	$\sigma = 0.4, delay = 7$ $C = 10,000, \gamma = 0.01$	Less than 0.0001	0.2108
HyS3 with ConKruG network	**56.78**	24.54	$\sigma = 0.4, delay = 7$ $C = 10,000, \gamma = 0.01$	Less than 0.0001	**0.0285**

Eq. (18) and the results are used to find the *p*-value. As most of the scientific studies, we find *p*-value of less than 0.05 proving our alternative hypothesis or (H_a: $\mu_1 \neq \mu_2$) and reject null hypothesis ($H_0 : \mu_1 = \mu_2$). It is clear that μ_1 and μ_2 are accuracies of the two models being compared. It should be mentioned that in Eq. (18), $\bar{X}$ means the average of time series X (accuracy of the model), and s^2 means the variance of accuracy for the model. n_1 and n_2 are the number of instances that are fed to the models as inputs that are equal in our tests as discussed before.

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{s_1^2/n_1 + s_2^2/n_2}}$$

$$df = \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{(s_1^2/n_1)^2/(n_1 - 1) + (s_2^2/n_2)^2/(n_2 - 1)} \tag{18}$$

4.3. Results and discussion

The results of HyS3 and its rival models are presented in Table 4 for comparison. First two rows show the results for two types of GSSL prediction: label propagation and label spreading results for Park and Shin model networks. Park and Shin used the label propagation model in their study but we achieved higher accuracy using their network with label spreading GSSL algorithm. These models are the most important graph-based financial predictions in the literature up to now.

Next, a simple SVM model is used for daily direction of movement prediction in dataset and the result is also presented. SVM was the most promising method as explained in Section 3. We tested SVM against many other single and ensemble supervised methods. The best model after SVM was random forest which its results are also provided for comparison (1000 run average). The SVM result shows that using only the past data of a time series in a supervised model for financial markets prediction is not a

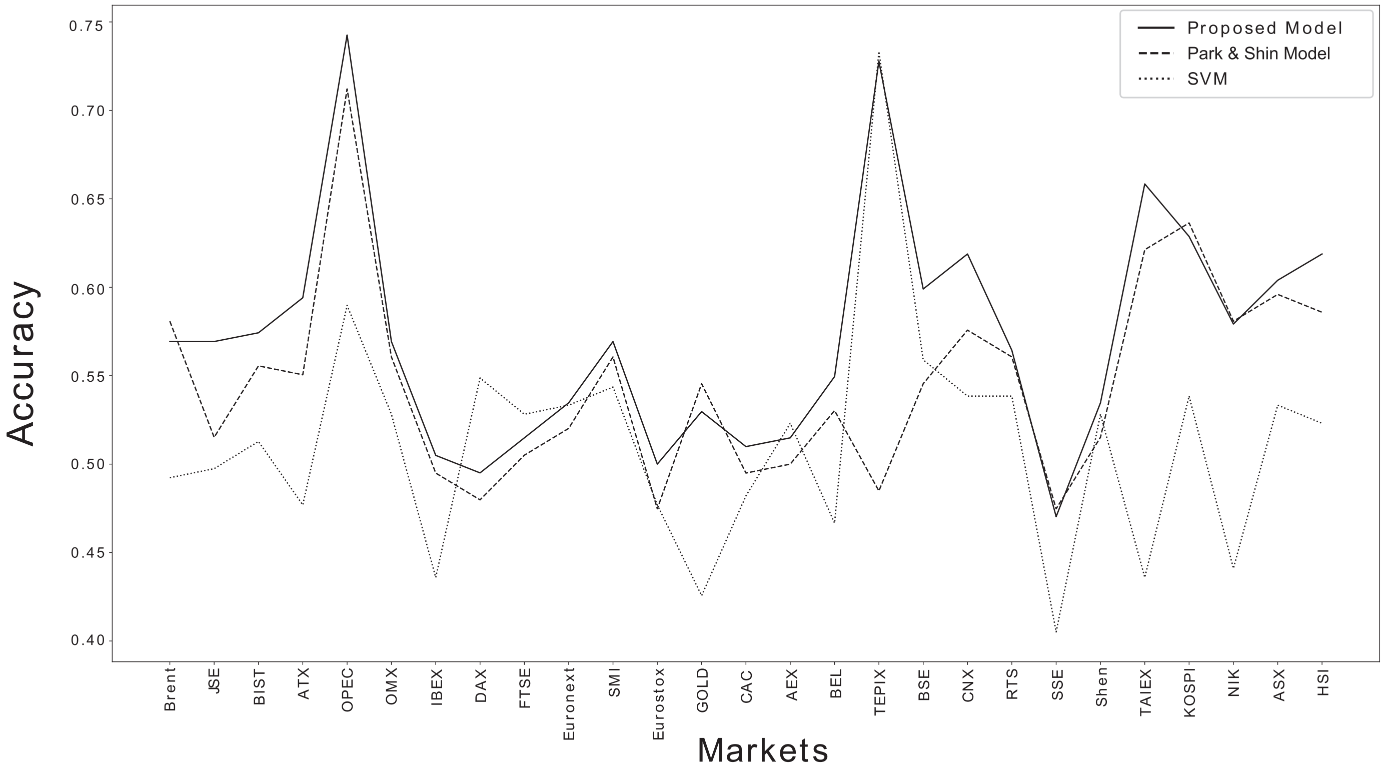


Fig. 9. Comparison between SVM, Park and Shin model, and the proposed hybrid model HyS3 (with ConKruG network).

Table 5
Comparing the accuracy results of top 10 predicted markets with HyS3 to results of rival supervised and semi-supervised models.

Models\Markets	NIK (%)	ATX (%)	BSE (%)	ASX (%)	CNX (%)	HSI (%)	KOSPI (%)	TAIEX (%)	TEPIX (%)	OPEC (%)
HyS3	57.92	59.41	59.90	60.40	61.88	61.88	62.87	65.84	73.77	74.26
Park and Shin (with label spreading)	58.08	55.05	54.55	59.60	57.58	58.59	63.64	62.12	48.48	71.21
SVM	44.10	47.69	55.90	53.33	53.85	52.31	53.85	43.59	73.33	58.97

good method comparing network-based models using other markets data. Supervised models only achieve good accuracy results in some markets like OPEC, TAIEX or TEPIX but in average it seems that using the data from other related markets to predict a target market is a more promising approach for prediction.

After presenting the SVM accuracy, we try to evolve the network structure in GSSL label spreading prediction model by first using the simplest kind of network: a complete graph with all edges weighted one. This simplest network achieves an accuracy of 54.38%. This and Park and Shin model with label spreading confirm the study of Zhou et al. (2004) that label spreading achieves better results than label propagation models. The complete weight-one network has better results than Park and Shin model with label propagation but it is less accurate than Park and Shin model with label spreading algorithm. This means a more intellectual network structure can help increasing the prediction accuracy.

Next, a correlation-based complete network is used and the accuracy result increases again comparing to the complete network with all weights equal to one. The correlation-based networks and their importance in econophysics was our road map to move towards this way of graph construction. The difference between results of the complete correlation-based network and ConKruG network shows that having all the edges in the network is not a good idea due to noisy and unrelated information added to the model by less correlated markets.

For achieving higher accuracy with tried our ConKruG algorithm for network construction. First a maximum spanning tree was made and then other phases of the ConKruG algorithm proceeded till the final network was ready. In Table 4, one row is for maximum spanning tree results with label spreading. The accuracy of the tree is a little higher than the complete weight-one network but not higher than the correlation-based complete network. This means that some useful information are lost in maximum spanning tree. After proceeding with the ConKruG algorithm, the results of using the final ConKruG network with label spreading algorithm again shows an increase in accuracy comparing the previous models of complete correlation-based and other networks that we tried.

After testing the graph-based algorithms with different networks and seeing the result of SVM as a supervised stand-alone method, the hybrid model results are presented in Table 4. First the correlation-based complete network is used with hybrid model of SVM probability injection. We get a higher accuracy over all models except the HyS3. HyS3 using the ConKruG network achieves the best accuracy result. This result is also statistically significant compared to the accuracy results of our rival models in Park and Shin (2013) and Shin et al. (2013) works. It should be mentioned that the accuracy improvement of HyS3 model is a synergy made from its non-separable modules like ConKruG and supervised predictor probability injection.

Other columns of Table 4 present the accuracy variances used in *t*-tests and best parameters used in the models which were obtained by running the models in validation set.

Fig. 9 shows the accuracy of the predicted markets in unknown zones for three models of SVM as representative of supervised models, Park and Shin with label spreading as the representative of the best results achieved with methods and networks made in literature of graph-based semi-supervised financial prediction research and our proposed hybrid model HyS3 using novel ConKruG algorithm. The accuracy chart for all three models shows that while in some markets, some of these models perform better than others, but in general, our model offers a higher prediction accuracy. In addition to this, Fig. 9 shows that HyS3 in general has considered the proper combination of the power of supervised model and semi-supervised model as its results.

Finally, Table 5 shows top 10 markets in terms of the prediction accuracy for HyS3, and their respective accuracy for SVM and Park and Shin model with label spreading for consideration.

5. Conclusion and future researches

A better prediction can be gained by using global markets information together with historical data of the markets being predicted. Capability of graph-based semi-supervised methods to use global market information for prediction depends on the constructed network that models markets interactions. A network that is built from the beginning of the construction process by looking at the type of problem that it seeks to solve is more suitable for prediction. We created an innovative network construction algorithm that modeled falls and rises of markets to reach best-possible prediction compared to other existing networks in the literature of research. In our observations, the use of other global markets information alone without using historical data of the market being predicted, will help to have a better prediction rather than only using a markets past data. However, the historical data of a market is also useful in predicting its future, but it should be used at the right time. We injected information extracted from past markets data as a primary knowledge into the network and used this knowledge to improve prediction. Injection of primary knowledge is possible through the probabilities calculated in a supervised approach. This knowledge, which is injected purposefully to some point of the ConKruG network using HyS3 method, can improve prediction. Using historical market data for each market, along with data from other global markets, we were able to provide a model for prediction that would provide a better prediction accuracy than other existing models.

In the future, these issues can be taken into consideration. Other sources of information, like ideas of users in the social networks about stocks and commodities can be taken into consideration. As an example Weng, Ahmed, and Megahed (2017) used user tweets in the tweeter for predicting a stock price. Researchers can make daily time series from tweets or user views of wikipedia entries of markets and commodities and present a node for each of these time series in the network structure as a non-traditional source of information other than prices, indexes, or technical and fundamental indicators. Another further research topic can be using a different learning phase setup to predict inflection points of the markets prices or indexes time series. As an example, by changing the labels from up/down movements to top and down turning points and a no comment class, the model will be learned to predict the turning points of the time series. Another suggestion for further research is using directed networks for prediction that can reduce information loss when building a network. At the same time, we should expect that directed networks may increase noise or the chance of overfitting in training process if not built carefully. The use of more varied data in network construction can also be considered. For example, currency exchange rates data can also be used in network construction along with our data. Finally, HyS3 along with ConKruG algorithm can be tested in non-financial areas of research.

Appendix A. Dataset information

Appendix B. Statistical information and imbalance ratio

Table A6
Market names, their abbreviations and their time zones used in the graphs with the resource of the datasets. Yahoo Finance: http://finance.yahoo.com, Google Finance: http://finance.google.com, Federal Reserve Bank of St. Luis. Economic Research Center: https://research.stlouisfed.org/fred2/, U.S. Energy Information Administration: http://tonto.eia.gov, Organization of Petroleum Exporting Countries: http://www.opec.org, Tehran Stock Exchange: http://www.tse.ir.

#	Data	Abbreviation	Time zone	Resource
1	New York Stock Exchange Composite	NYSE	1	Yahoo Finance
2	Nasdaq Composite	Nasdaq	1	Yahoo Finance
3	S&P 500	SP	1	finance.yahoo.com
4	Dow Jones Industrial Average	DJIA	1	Federal Reserve Bank of St. Louis
5	Russell 2000	Russel	1	Google Finance
6	S&P/TSX Composite (Toronto Stock Exchange)	TSX	1	Yahoo Finance
7	Indice Bovespa	IBOVES	1	Yahoo Finance
8	Mercado Valero	MERVAL	1	Google Finance
9	West Texas Intermediate Oil Price	WTI	1	U.S. Energy Information Administration
10	Financial Times Stock Exchange 100	FTSE	2	Yahoo Finance
11	Euronext 100	Euronext	2	Yahoo Finance
12	Euro Stoxx 50	Eurostox	2	Yahoo Finance
13	CAC 40	CAC	2	Yahoo Finance
14	Amsterdam Exchange Index	AEX	2	Yahoo Finance
15	BEL 20	BEL	2	Yahoo Finance
16	Deutscher Aktienindex	DAX	2	Yahoo Finance
17	Swiss Market Index	SMI	2	Yahoo Finance
18	OMX Stockholm 30	OMX	2	Yahoo Finance
19	IBEX 35	IBEX	2	Yahoo Finance
20	Austrian Traded Index	ATX	2	Yahoo Finance
21	Borsa Istanbul 100	BIST	2	Yahoo Finance
22	Johannesburg Stock Exchange	JSE	2	Google Finance
23	Brent Crude Oil Price	Brent	2	U.S. Energy Information Administration
24	OPEC Oil Price	OPEC	2	Organization of Petroleum Exporting Countries Official Website
25	Gold Fixing Price of London	Gold	2	Federal Reserve Bank of St. Louis
26	Russia Trading System	RTS	3	Yahoo Finance
27	Tehran Stock Exchange	TEPIX	3	Tehran Stock Exchange Official Website
28	Bombay Stock Exchange	BSE	3	Yahoo Finance
29	CNX Nifty 50	CNX	3	Yahoo Finance
30	Hang Seng Index	HSI	4	Yahoo Finance
31	SSE Composite Index	SSE	4	Yahoo Finance
32	Shenzen Composite Index	Shen	4	Yahoo Finance
33	Taiwan Capitalization Weighted Stock Index	TAIEX	4	Yahoo Finance
34	Korea Composite Stock Price Index	KOSPI	4	Yahoo Finance
35	NIKKEI 225	NIK	4	Yahoo Finance
36	Australian Securities Exchange	ASX	4	Yahoo Finance

Table B7
Some statistical information about the dataset: mean and standard deviations of the real market data in the time periods of train, validation and test sets are presented. Percentage of up and down classes are also shown in Inc% and Dec% columns. At last the imbalance ratio (IR) is calculated for each market and also for the whole dataset.

Market abr.	Train (1409 days)				Validation (402 days)				Test (202 days)				Whole dataset (2013 days)				
	Inc%	Dec%	Mean	Std	Inc%	Dec%	Mean	Std	Inc%	Dec%	Mean	Std	Inc#	Dec#	Inc%	Dec%	IR
NYSE	50.8	49.2	7607.1	1197.4	54.7	45.3	9959.0	648.8	49.0	51.0	10885.8	232.4	1035.0	978.0	51.4	48.6	1.1
Nasdaq	51.7	48.3	2429.7	448.4	57.0	43.0	3869.9	407.5	53.5	46.5	4761.9	212.9	1066.0	947.0	53.0	47.0	1.1
SP	52.1	47.9	1213.3	191.3	56.5	43.5	1758.1	137.0	47.5	52.5	2046.9	58.0	1057.0	956.0	52.5	47.5	1.1
DJIA	51.0	49.0	11323.1	1694.4	54.5	45.5	15727.6	834.2	50.0	50.0	17625.9	486.7	1038.0	975.0	51.6	48.4	1.1
Russel	49.5	50.5	695.1	121.6	54.5	45.5	1076.8	87.2	51.0	49.0	1194.0	51.8	1019.0	994.0	50.6	49.4	1.0
TSX	51.8	48.2	12075.8	1525.0	57.2	42.8	13514.4	1008.0	49.5	50.5	14894.8	425.5	1060.0	953.0	52.7	47.3	1.1
IBOVES	49.0	51.0	59439.0	8879.1	48.0	52.0	52390.7	3463.6	43.1	56.9	53002.2	3522.3	970.0	1043.0	48.2	51.8	1.1
MERVAL	49.0	51.0	1737.4	612.2	53.2	46.8	4110.7	1583.3	50.5	49.5	9467.5	1302.3	1007.0	1006.0	50.0	50.0	1.0
WTI	49.4	50.6	86.3	19.9	50.0	50.0	99.4	5.1	42.6	57.4	64.3	16.6	983.0	1030.0	48.8	51.2	1.0
Brent	50.4	49.6	92.3	23.9	47.8	52.2	108.0	3.9	41.6	58.4	68.6	16.1	986.0	1027.0	49.0	51.0	1.0
JSE	45.8	54.2	6609.2	1212.3	48.3	51.7	8527.0	930.1	52.0	48.0	11719.3	1121.7	945.0	1068.0	46.9	53.1	1.1
BIST	50.9	49.1	52580.1	13864.8	49.0	51.0	75524.4	6918.7	51.5	48.5	82522.4	4046.8	1018.0	995.0	50.6	49.4	1.0
ATX	47.8	52.2	2632.4	793.9	47.8	52.2	2473.2	111.1	52.5	47.5	2348.4	186.6	972.0	1041.0	48.3	51.7	1.1
OPEC	53.2	46.8	89.9	23.1	45.3	54.7	105.2	3.2	40.1	59.9	65.8	16.7	1013.0	1000.0	50.3	49.7	1.0
OMX	50.3	49.7	947.6	175.3	52.7	47.3	1235.6	78.0	54.5	45.5	1478.6	132.5	1031.0	982.0	51.2	48.8	1.0
IBEX	48.9	51.1	10161.7	2181.2	53.5	46.5	9431.8	1021.3	55.4	44.6	10790.4	520.9	1016.0	997.0	50.5	49.5	1.0
DAX	50.7	49.3	6294.5	970.5	53.5	46.5	8878.7	729.8	55.4	44.6	10474.3	1075.4	1041.0	972.0	51.7	48.3	1.1
FTSE	49.1	50.9	5440.8	646.9	52.2	47.8	6599.4	170.6	52.0	48.0	6749.1	217.7	1007.0	1006.0	50.0	50.0	1.0
Euronext	49.7	50.3	678.6	121.2	52.5	47.5	781.9	48.6	55.0	45.0	897.4	78.6	1022.0	991.0	50.8	49.2	1.0
SMI	50.3	49.7	6440.7	855.5	52.0	48.0	8145.2	332.2	52.0	48.0	8912.0	323.7	1023.0	990.0	50.8	49.2	1.0
Eurostox	47.8	52.2	2818.4	575.7	50.7	49.3	2950.0	214.7	53.0	47.0	3355.4	255.3	985.0	1028.0	48.9	51.1	1.0
GOLD	51.4	48.6	1243.3	341.2	47.8	52.2	1343.2	105.8	42.6	57.4	1211.2	33.6	1002.0	1011.0	49.8	50.2	1.0
CAC	49.2	50.8	3783.2	697.0	52.2	47.8	4142.9	259.0	53.5	46.5	4605.1	384.4	1011.0	1002.0	50.2	49.8	1.0
AEX	49.1	50.9	340.6	71.9	52.2	47.8	381.5	22.5	53.5	46.5	448.0	38.1	1010.0	1003.0	50.2	49.8	1.0
BEL	49.3	50.7	2598.8	627.0	51.7	48.3	2864.8	211.8	53.5	46.5	3433.3	266.9	1010.0	1003.0	50.2	49.8	1.0
TEPIX	32.3	67.7	17542.3	7894.1	33.8	66.2	65486.5	16008.4	25.7	74.3	68541.0	4200.9	643.0	1370.0	31.9	68.1	2.1
BSE	47.9	52.1	16535.9	2898.3	52.2	47.8	21196.3	2289.7	45.0	55.0	27792.7	896.8	976.0	1037.0	48.5	51.5	1.1
CNX	48.6	51.4	4970.1	860.7	49.8	50.2	6336.0	678.4	47.0	53.0	8372.6	301.5	980.0	1033.0	48.7	51.3	1.1
RTS	50.5	49.5	1527.0	445.4	46.0	54.0	1352.3	108.5	40.1	59.9	967.0	139.7	977.0	1036.0	48.5	51.5	1.1
SSE	47.6	52.4	2840.9	839.0	45.8	54.2	2135.5	105.1	60.4	39.6	3206.7	769.0	977.0	1036.0	48.5	51.5	1.1
Shen	46.7	53.3	11153.1	2726.7	44.0	56.0	8139.0	713.1	48.5	51.5	10962.2	2437.5	933.0	1080.0	46.3	53.7	1.2
TAIEX	50.3	49.7	7500.9	1182.4	52.7	47.3	8463.1	483.4	43.6	56.4	9332.0	316.9	1009.0	1004.0	50.1	49.9	1.0
KOSPI	49.0	51.0	1744.2	273.3	49.8	50.2	1970.8	53.6	46.0	54.0	2003.6	73.7	984.0	1029.0	48.9	51.1	1.0
NIK	47.6	52.4	10355.2	2068.5	50.5	49.5	14334.9	1099.9	55.9	44.1	17793.8	1603.3	987.0	1026.0	49.0	51.0	1.0
ASX	49.0	51.0	4616.7	692.1	51.7	48.3	5239.1	220.4	46.5	53.5	5587.7	241.0	993.0	1020.0	49.3	50.7	1.0
HSI	48.3	51.7	20937.5	3426.5	49.3	50.7	22683.5	902.2	47.5	52.5	24798.3	1602.9	975.0	1038.0	48.4	51.6	1.1
Average	49.1	50.9	8307.9	1699.4	50.6	49.4	10926.0	1166.6	48.9	51.1	12288.3	766.8	**993.4**	**1019.6**	**49.3**	**50.7**	**1.1**

References

Anderson, J. R. (2013). *The architecture of cognition*. Psychology Press.

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

Baghshah, M. S., Afsari, F., Shouraki, S. B., & Eslami, E. (2014). Scalable semi-supervised clustering by spectral kernel learning. *Pattern Recognition Letters, 45*, 161–171.

Boyacioglu, M. A., & Avci, D. (2010). An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: The case of the istanbul stock exchange. *Expert Systems with Applications, 37*(12), 7908–7912.

Cawley, G. C., & Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research, 11*(Jul), 2079–2107.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST), 2*(3), 27.

Chapelle, O., Scholkopf, B., & Zien, A. (2009). Semi-supervised learning (Chapelle, O. et al., Eds.; 2006) [book reviews]. *IEEE Transactions on Neural Networks, 20*(3), 542.

Cormen, T. H. (2009). *Introduction to algorithms*. MIT Press.

Cortes, C., & Vapnik, V. (1995). Support vector machine. *Machine Learning, 20*(3), 273–297.

Fathian, M., & Kia, A. (2012). Exchange rate prediction with multilayer perceptron neural network using gold price as external factor. *Management Science Letters, 2*(2), 561–570.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861–874.

Fernández, A., García, S., del Jesus, M. J., & Herrera, F. (2008). A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems, 159*(18), 2378–2398.

Gao, X., An, H., & Zhong, W. (2013). Features of the correlation structure of price indices. *PLoS One, 8*(4), e61091.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(8), 832–844.

Homm, U., & Breitung, J. (2012). Testing for speculative bubbles in stock markets: A comparison of alternative methods. *Journal of Financial Econometrics, 10*(1), 198–231.

Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research, 32*(10), 2513–2522.

Jaruszewicz, M., & Mańdziuk, J. (2004). One day prediction of Nikkei index considering information from other stock markets. In Proceedings of the 2004 artificial intelligence and soft computing – ICAISC 2004, (pp. 1130–1135).

Kara, Y., Boyacioglu, M. A., & Baykan, Ö. K. (2011). 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, 38*(5), 5311–5319.

Kia, A. N. (2018). 36 stock indices and commodity prices time series, Mendeley data, v1. 10.17632/x744mgjpkv.1 .

Kia, A. N., Fathian, M., & Gholamian, M. (2012). Using MLP and RBF neural networks to improve the prediction of exchange rate time series with ARIMA. *International Journal of Information and Electronics Engineering, 2*(4), 543.

Kia, A. N., Haratizadeh, S., & Heshmati, Z. (2016). Analysis and prediction of fluctuations for sector price indices with cross correlation and association based networks: Tehran stock exchange case. *Bonfring International Journal of Industrial Engineering and Management Science, 37*(4), 95.

Kia, A. N., Haratizadeh, S., & Shouraki, S. B. (2018). "Hys3-ConKrug prediction model", Github, v1.0. URL:https://github.com/ConKruG/HyS3/tree/v1.0.

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing, 55*(1), 307–319.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society, 7*(1), 48–50.

Kullmann, L., Kertész, J., & Kaski, K. (2002). Time-dependent cross-correlations between different stock returns: A directed network of influence. *Physical Review E, 66*(2), 026125.

Lin, F., Yeh, C. C., & Lee, M. Y. (2013). A hybrid business failure prediction model using locally linear embedding and support vector machines. *Romanian Journal of Economic Forecasting, 16*(1), 82–97.

Liu, X. F., & Tse, C. K. (2012). A complex network perspective of world stock markets: Synchronization and volatility. *International Journal of Bifurcation and Chaos, 22*(06), 1250142.

Mantegna, R. N. (1999). Hierarchical structure in financial markets. *The European Physical Journal B-Condensed Matter and Complex Systems, 11*(1), 193–197.

Mantegna, R. N., & Stanley, H. E. (1999). *Introduction to econophysics: Correlations and complexity in finance*. Cambridge University Press.

Narayan, P. K., Sharma, S. S., & Thuraisamy, K. S. (2015). Can governance quality predict stock market returns? New global evidence. *Pacific-Basin Finance Journal, 35*, 367–380.

Papana, A., Kyrtsou, C., Kugiumtzis, D., & Diks, C. (2017). Financial networks based on Granger causality: A case study. *Physica A: Statistical Mechanics and its Applications, 482*, 65–73.

Park, K., & Shin, H. (2013). Stock price prediction based on a complex interrelation network of economic factors. *Engineering Applications of Artificial Intelligence, 26*(5), 1550–1561.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications, 42*(1), 259–268.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers, 10*(3), 61–74.

Rather, A. M., Sastry, V., & Agarwal, A. (2017). Stock market prediction and portfolio selection models: A survey. *OPSEARCH, 54*(3), 558–579.

Rice, J. (2006). *Mathematical statistics and data analysis*. Nelson Education.

Roy, R. B., & Sarkar, U. K. (2013). A social network approach to change detection in the interdependence structure of global stock markets. *Social Network Analysis and Mining, 3*(3), 269–283.

Sapankevych, N. I., & Sankar, R. (2009). Time series prediction using support vector machines: a survey. *IEEE Computational Intelligence Magazine, 4*(2), 24–38.

Shin, H., Hou, T., Park, K., Park, C.-K., & Choi, S. (2013). Prediction of movement direction in crude oil prices based on semi-supervised learning. *Decision Support Systems, 55*(1), 348–358.

Shin, K.-S., Lee, T. S., & Kim, H.-J. (2005). An application of support vector machines in bankruptcy prediction model. *Expert Systems with Applications, 28*(1), 127–135.

Shrager, J., Hogg, T., & Huberman, B. A. (1987). Observation of phase transitions in spreading activation networks. *Science, 236*(4805), 1092–1094.

Soni, S. (2011). Applications of ANNs in stock market prediction: A survey. *International Journal of Computer Science & Engineering Technology, 2*(3), 71–83.

Sui, X., Hu, Q., Yu, D., Xie, Z., & Qi, Z. (2007). A hybrid method for forecasting stock market trend using soft-thresholding de-noise model and SVM. In *International workshop on rough sets, fuzzy sets, data mining, and granular-soft computing* (pp. 387–394). Heidelberg: Springer, Berlin.

Thawornwong, S., & Enke, D. (2004). The adaptive selection of financial and economic variables for use with artificial neural networks. *Neurocomputing, 56*, 205–232.

Tsai, C.-F., & Hsiao, Y.-C. (2010). Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. *Decision Support Systems, 50*(1), 258–269.

Weng, B., Ahmed, M. A., & Megahed, F. M. (2017). Stock market one-day ahead movement prediction using disparate data sources. *Expert Systems with Applications, 79*, 153–163.

Yao, J., & Tan, C. L. (2000). A case study on using neural networks to perform technical forecasting of Forex. *Neurocomputing, 34*(1), 79–98.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schölkopf, B. (2004). Learning with local and global consistency. In *Proceedings of the advances in neural information processing systems* (pp. 321–328).

Zhu X. Semi-supervised learning literature survey. Tech rep 1530, computer sciences. University of Wisconsin-Madison; 2005.

Zhu, X., Ghahramani, Z., & Lafferty, J. D. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the twentieth international conference on machine learning (ICML-03)* (pp. 912–919).