Engineering Applications of Artificial Intelligence 88 (2020) 103340

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Network-based direction of movement prediction in financial markets ☆

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

[a] Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran
[b] Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Keywords:
Graph-based semi-supervised learning
Mixture of experts
Network modeling
Time series prediction

ABSTRACT

Market prediction has been an important research problem for decades. Having better predictive models that are both more accurate and faster has been attractive for both researchers and traders. Among many approaches, semi-supervised graph-based prediction has been used as a solution in recent researches. Based on this approach, we present two prediction models. In the first model, a new network structure is introduced that can capture more information about markets' direction of movements compared to the previous state of the art methods. Based on this novel network, a new algorithm for semi-supervised label propagation is designed that is able to prediction the direction of movement faster and more accurately.

The second model is a mixture of experts system that decides between supervised or semi-supervised approaches. Besides this, the model gives us the ability to identify the markets that their data are helpful in constructing the network. Our models are shown to be both faster regarding computational complexity and running time and more accurate in prediction comparing to best rival models in literature of graph-based semi-supervised prediction. The results are also tested to be statistically significant.

1. Introduction

Direction of movement prediction is an important research topic in financial time series studies. Many traders prefer to know about the direction of movement of a market rather than a precise value for the prices or indices (Yao and Tan, 2000). A traders' decision of hold, buy, or sell comes from his/her perception of future direction rather than exact values of price or index. That is the reason for development of several direction of movement prediction models in many researches like Kia et al. (2018), Li and Liao (2017), Patel et al. (2015), Imandoust and Bolandraftar (2014), Kara et al. (2011) and Huang et al. (2005).

Efficient market hypothesis (EMH) indicates that all information revealed and related, immediately affect the price of a financial time series (Fama et al., 1969). One of the results from the EMH is that predicting prices and stock indices from historical data will not provide profit for the traders (Timmermann and Granger, 2004). EMH has many critics and measuring the level of efficiency in markets has been a challenging topic of research for the decades. There are also many researches in the field of finance and economic that bring evidences in favor of using prediction models to gain profit and show anomalies in EMH (Naseer and Bin Tariq, 2015; Malkiel, 2003; Lo and MacKinlay, 1988). Many researches in the field of financial prediction along with different methods as well as the success of several algorithmic trading companies are evidences that show the traders have found predictions valuable.

There are different categories of methods in the literature for financial time series prediction. The oldest ones are technical analysis methods which use traditional statistical methods to calculate some indicators for prediction. A good survey on these methods can be found in the work of Atsalakis and Valavanis (2013). Some other researches use the financial statements of markets and companies to predict their future. This approach that is called fundamental analysis method has been used in many other researches like Yan and Zheng (2017), Chen et al. (2017), Shen and Tzeng (2015) and Abarbanell and Bushee (1997). Both technical and fundamental analysis are conventional methods used by the traders for decades. With emergence of the machine learning and data mining science, many researchers turned into using these algorithms and techniques for financial time series forecasting. These methods and algorithms have shown to outperform the conventional methods mentioned before in different studies (Atsalakis and Valavanis, 2009). Several surveys have compared machine learning and hybrid methods that use machine learning with conventional methods for market prediction (Rather et al., 2017; Cavalcante et al., 2016; Atsalakis and Valavanis, 2009; Preethi and Santhi, 2012; Soni, 2011).

From one perspective, machine learning algorithms are divided into three categories of supervised, unsupervised, and semi-supervised

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.engappai.2019.103340.

* 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).

https://doi.org/10.1016/j.engappai.2019.103340
Received 25 October 2018; Received in revised form 5 August 2019; Accepted 29 October 2019
Available online xxxx
0952-1976/

learning (Zhu, 2006). Supervised learning has been used mostly for regression and classification models. In supervised prediction, the model is learned by sample instances that have known class labels, then the learned model is used to predict the label of new instances with unknown labels. The labels can be direction of movement of markets time series. In semi-supervised learning there are a few samples with known labels and many samples without labels. Semi-supervised learning tries to label the unlabeled samples using both the labeled ones as well as the underlying structure of all training samples. A network-based approach to find out this underlying structure is called graph-based semi-supervised learning (Goldberg and Zhu, 2006). Using supervised learning models trained with historical data of the market in prediction means that we assume past patterns in the market data can be helpful in prediction of its future. As mentioned before EMH challenges this assumption by reminding us that efficient markets behave as a random-walk processes. In our experiments we found out that many financial time series cannot be predicted better than a fair coin model with their own historical data and this may be an evidence of their efficiency. But some other financial time series were predictable with their historical data. We also observed that many of the stock market indices were predictable using the data from other stock markets and commodity prices. This led us to use semi-supervised learning models where we knew the direction of movements for some markets and some other markets had to be predicted.

The interrelationship among markets with known and unknown labels can be modeled using graph structures (a network). The network models have shown their superiority in representing financial markets interrelations in the past researches (Pereira et al., 2017; Jovanovic and Schinckus, 2017; McCauley, 2004; Mantegna and Stanley, 1999). Some researches like Kia et al. (2018), Saumya et al. (2016), Skabar (2013), Shin et al. (2013), Park and Shin (2013) and Upstill et al. (2003) have tried to forecast financial time series using network models. Different network structures have been used and different semi-supervised learning algorithm over the network have been applied in these studies.

In the first part of this research, we introduce a new approach for modeling the financial markets data as a network structure that captures the information that is relevant and important regarding the desired prediction task. Also we show a novel approach for analyzing the suggested network structure in order to forecast different markets faster and more accurately. As we will see, the experiments shows the superiority of the suggested approach in capturing relevant movement relationship between different markets and using it for movement prediction, compared to previous networks.

The new network structure used in our research is built upon the information of association rules such as $if\ market\ A\ goes\ up \implies market\ B\ goes\ down$ that are extracted from the dataset. A novel weighting method is used for weighting the links between markets in the network. The weighting of the links or edges of the network is important in graph-based semi-supervised learning. Our idea of weighting the edges is taken from the concept of lift of rules in association rules mining domain. The nodes of the network represent up and down directions of each market. Then known information of the markets in time zones that their markets are open, are used to predict the direction of markets in the closed time zones.

In this study we are going to predict next time-step direction of movements in financial time series of stock markets and commodity prices with a novel semi-supervised label propagation approach using a modified personalized PageRank-based algorithm on a network structure of different time series direction of movements interrelations. A novel mixture of experts model is then designed to decide between supervised or semi-supervised prediction approach. The ability of a hybrid model that uses both the benefits of supervised and semi-supervised methods, for movement prediction has been shown in previous studies (Kia et al., 2018). In this study we suggest a new model that achieves better results using a new semi-supervised approach for analyzing a network model with a novel design. But again if in some cases a market is predicted better with its own historical data, the mixture of experts model guides us to use a fully supervised model. This model also helps us find out which markets should be in the network structure. We will show the superiority of our models over other models in terms of prediction performance, computational complexity, and running time in the next sections of the paper.

We will name the proposed models DiMexRank and MixDiMex. DiMexRank (**Di**rection of **M**ovement prediction with **ex**tended lift network using modified Page**Rank**) and MixDiMex, (**Mix**ture of experts model using supervised learning and **DiMex**Rank) are faster, and more accurate comparing to their best rival models in the literature. MixDiMex is also a way to find out whether a market's data is helpful in prediction of other markets future direction or not.

In the next section, we will explain the related works and baseline rival models to our work. In Section 3, the network construction and the prediction algorithms are explained in details. Section 4 compares computational complexity of our proposed models to the rival baseline models. Section 5 evaluates our models, compares the prediction results with other baseline models, and has discussions and interpretations of the results. At the end, Section 6 concludes and suggests some paths for the future research in semi-supervised graph-based financial prediction. Appendices A and B present information about the dataset that is used in experiments.

2. Related works

Different learning methods for information propagation are used in the financial time series prediction with networks. The label propagation and label spreading algorithms have been described in works of Zhou et al. (2004) and Zhu and Ghahramani (2002) and were used for financial prediction in works of Kia et al. (2018), Shin et al. (2013), Park and Shin (2013) and Skabar (2013). These algorithms try to propagate known labels of nodes (directions of markets) in a graph to nodes with unknown label. They work under the condition of the underlying graph structure of markets being undirected. Shin et al. (2013) and Park and Shin (2013) made a network of each node as a vector of some technical indicators of each market, and edges as the Euclidean distance between the nodes. They used label propagation algorithm for prediction. Kia et al. (2018) tried to make a correlation network from the time series of markets and then used label spreading algorithm to predict direction of unknown markets. Both correlation distance and Euclidean distance are symmetric measures and therefore the underlying graph of the problem they used was undirected. As we mentioned before, modeling the interrelations as undirected graphs may be misleading in situations in which a market affecting another market is not a symmetric relation.

Contrary to many network-based prediction researches (that will be explained later), many patterns and information between markets are not symmetric. For example a market's direction going up may be the cause of another market going down while the other direction is not true. This leads us to the idea of using a directed network for capturing the behavioral patterns among markets.

In this research we use a directed graph structure that is made to capture more patterns of the dataset which consists of different markets time series. In our work, each node represents the state of a market going up or down in the direction and the edges' weights represent a measure like lift which tries to show how probable is that markets A and B will be in a specific direction state together regardless of them being in that direction independently. We will call this similarity measure, extended lift or exLift. Extended lift and our network structure are explained in detail in the future sections. At the end a modified personalized PageRanK is used as a novel semi-supervised algorithm to calculate the ranks of all nodes and predict the direction of each market. By subtracting the predicted ranks of down nodes (negative nodes) from up nodes (positive nodes) we will reach a positive or negative label meaning upward/downward movement for each market with unknown direction.

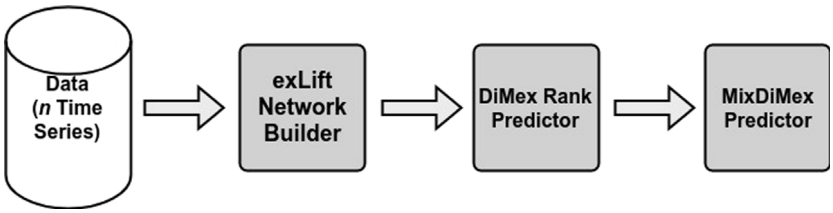

Fig. 1. General steps in prediction model construction.

We compare our proposed models to some baseline models which are described briefly in the following. For supervised prediction different supervised models from simple data mining algorithms to deep neural networks were tested in validation set and random forest achieved the best result. Therefore, for a fair comparison between supervised and semi-supervised models and for a better performance in the mixture of experts model, we used random forest. Random forest is an state-of-the-art ensemble supervised model that uses random sub-sampling of the dataset making sub-datasets with different random features to construct different decision trees (Verikas et al., 2011). Each decision tree predicts the time series for itself and majority voting among the decision trees makes the final prediction of random forest. For more knowledge on the random forest algorithm one can refer to Ho (1995). There are other researches that show the superiority of random forest in direction prediction comparing to other supervised models (Ballings et al., 2015).

We will describe the rival models of our proposed models in this section in brief. For more detailed explanation of these models the researchers can read the original papers referenced at each section. In evaluation Section 5.2 we compare our models with these baseline models. In Section 3.3 we present supervised models as rivals of network-based models. These supervised models are compared to the networks extracted and are used whenever they gain better performances with a mechanism designed in our presented mixture of experts model. The random forest model is selected as the best supervised performer in our dataset. This model has also shown superiority in other recent works like Zhang et al. (2018) when predicting with supervised learners.

Top four models in graph-based semi-supervised learning and the best performance supervised model in our dataset are selected for the comparison.

The first model that is used for comparison is Park and Shin graph-based semi-supervised prediction model. Park and Shin (2013) presented a network structure for modeling stock markets to predict Korean Stock Exchange index using other stock markets and some commodity prices data. They made vectors of some technical analysis indicators for each time series and used the Euclidean distance between these vectors to find the weight of edges in the proposed network. The graph was filtered by removing low-weight edges (Shin et al., 2013). Finally a label propagation algorithm is used to find the direction of movement in the target market by the known direction of movements of other markets (Zhu and Ghahramani, 2002). The enhanced Park and Shin model is the second model used to be compared with our presented models. Kia et al. (2018) used the label spreading algorithm introduced by Zhou et al. (2004) in the network structure provided by Park and Shin (2013) and achieved significantly better prediction performance.

The ConKruG algorithm for network construction was used along with Park and Shin models to make better prediction results and be a stronger rival to our models when the results where compared. Kia et al. (2018) provided an algorithm to construct a network between market time series using a modified Kruskal algorithm that finds maximum spanning trees (Kruskal, 1956). In ConKruG first a maximum spanning tree of the complete correlation-based graph of markets is made. Then other edges of the complete graph are added sequentially, according to their weight in a descending order. The edge adding process tries to only add edges that are helpful in improving the semi-supervised prediction accuracy. Finally this network is used in label spreading algorithm to predict direction of movements for markets.

Kia et al. (2018) provided an algorithm that injected the probabilities of up or down direction of movement extracted from supervised prediction models to the initial matrix of the label spearing algorithm. The label spreading algorithm provided by Zhou et al. (2004) has an initial matrix with n rows for n nodes and c columns for c classes of different levels of direction of movement. In HyS3, the probabilities extracted from a support vector machine is injected to the rows of the initial label spreading matrix for some nodes. That process tries to find which nodes are helpful in semi-supervised prediction when information from their own historical data is used in the HyS3 prediction process. So both historical data of the market and previous data of other markets were used in this prediction approach. At the end each rows of the final converged matrix of the results is checked to see which column has higher value. The column with the higher value corresponds the predicted direction or class.

3. Prediction model

Fig. 1 shows the general procedure of the suggested framework. First, the dataset of raw value time series of stock indices and commodity prices are converted to return series.

The return is a popular measure in finance and shows the degree of variation of a variable in a logarithmic scale. An element in the return series of markets is calculated by Eq. (2) in which $series_{i,t}$ is the value of the market index i in time t. A Positive or a negative return means an upward or downward movement in the direction of the corresponding market. After converting the raw prices or indices to the return value, the return data of all time series are used to construct a network of interrelationships among markets' movements in exLift network builder part of the procedure. The network construction is the learning part of the DiMexRank algorithm while the prediction part is shown in the DiMexRank predictor box. In this step a modified personalized PageRanK is used to propagate the ranks of known markets (known up and down movements) to the markets that are going to be predicted. The module represented in the last box, MixDiMex Predictor, tries to find out if any market is better predicted by supervised learning compared to network-based model. If any market is predicted more accurately with supervised models, then it will be checked if existence of this market is helpful for predicting other markets in the network or not. At the end, based on the resulting model, the final prediction for each market is made either by supervised or semi-supervised graph-based model. Also each series is tested if its existence is useful in the network learning phase or not (in MixDiMex Phase). In next sections we will explain all the parts of Fig. 1 in details.

3.1. exLift network structure

In this section, a network called exLift, is created to model the interrelationships among markets' direction of movement. exLift is a graph $G = (V, E)$ in which V is the set of nodes and E the set of edges. We assume that there are n market time series in the dataset and M is the set of all markets: $M = \{m_1, m_2, \ldots, m_n\}$. For each market m_i, we will put two nodes of m_i^+ and m_i^- in the graph G, that respectively

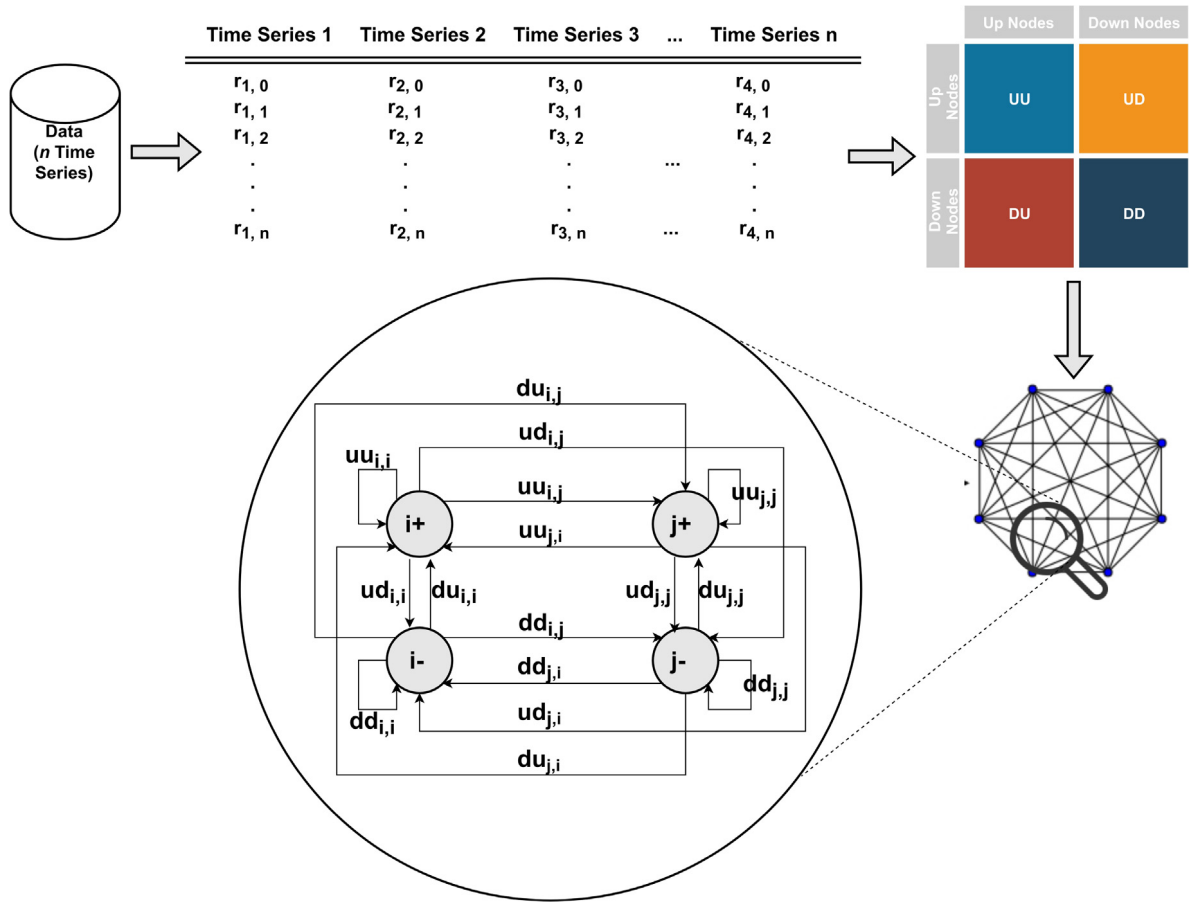

Fig. 2. General steps of extended-Lift network construction.

represent up and down directions in the market m_i. Therefore the set of nodes will be $V = \{m_1^+, m_1^-, m_2^+, m_2^-, \ldots, m_n^+, m_n^-\}$. For each pair of nodes $(m_i^{(+/-)}, m_j^{(+/-)})$ there is an edge whose weight is calculated from a proposed weighting function called extended lift or simply exLift. The exLift has come from the concept of lift measure in the association rules mining in data mining literature which is calculated according to Eq. (1) (Brin et al., 1997). Lift measures how much the occurrence of event A in $A \implies B$ affects the chance of occurrence for B. As formulated in Eq. (3), for calculating the exLift measure, the magnitude of upward/downward movements are taken into account as well. The function δ is used to check if the conditions of two markets match a rule in a certain time unit.

Fig. 2 shows a general schema of the network construction. A sample view of all the edges between four nodes of two markets of i and j is also shown in 2. It can be seen in Fig. 2 that for a network that only considers two markets, there will be four nodes and 16 edges.

While it is obvious from the Eqs. (1) that lift is a symmetric measure, is not a symmetric measure according to its definition (3). That means that the exLift network structure should be a directed graph.

$$Lift(A \implies B) = \frac{P(A \wedge B)}{P(A)P(B)} \qquad (1)$$
$$Lift(A \implies B) = Lift(B \implies A)$$

$$return_{i,t} = log\frac{series_{i,t}}{series_{i,t-1}} \qquad (2)$$

$$Direction = \{up : +1, down : -1\}, \quad direction_i, direction_j \in Direction$$

$$\delta(series_{i,t,direction}) = \begin{cases} 1 & if\ sign(return_{i,t}) = sign(direction) \\ 0 & otherwise \end{cases}$$

$$exLift(i^{(+/-)} \xrightarrow{1} j^{(+/-)}) =$$

$$= \frac{\sum_t \delta(series_{i,t,direction_i})\delta(series_{j,t+1,direction_j})|return_{j,t+1}|}{\sum_t \delta(series_{i,t,direction_i})|return_{i,t}| \sum_t \delta(series_{j,t+1,direction_j})|return_{j,t+1}|}$$

$$(i^+, i^-, j^+, j^- means : direction_i = +1, direction_i = -1, direction_j = +1, direction_j = -1)$$

$$(\xrightarrow{1} means : after\ one\ time - step) \qquad (3)$$

$$exLift(i^{(+/-)} \xrightarrow{1} j^{(+/-)}) \neq exLift(j^{(+/-)} \xrightarrow{1} i^{(+/-)}) \qquad (4)$$

After calculating exLift for every edge in E, the adjacency matrix of the complete exLift graph is constructed according to Eq. (5). Here, the adjacency matrix of the exLift graph is denoted by Ψ. This matrix is made of four block sub-matrices called UU, UD, DU, and DD. UU is the block that is associated with rules like $i^+ \implies j^+$. The exLifts of such rules are the weights of edges between nodes representing upward movements of the markets. UD block is associated with rules like $i^+ \implies j^-$. Likewise, DU and DD blocks of the adjacency matrix Ψ correspond to $i^+ \implies j^-$ and $i^- \implies j^-$ rules respectively, for each $i, j \in V$. A sample part of the graph between two markets of i and j is presented in Fig. 2.

$$\Psi = \begin{pmatrix} UU & UD \\ DU & DD \end{pmatrix},$$
$$UU_{i,j} = exLift(i^+ \xrightarrow{1} j^+),\ UD_{i,j} = exLift(i^+ \xrightarrow{1} j^-), \qquad (5)$$
$$DU_{i,j} = exLift(i^- \xrightarrow{1} j^+),\ DD_{i,j} = exLift(i^- \xrightarrow{1} j^-)$$

The pseudo-code of exLift network construction algorithm is presented in algorithm 1. Dataset of n time series return values and time zones of each time series are the inputs of the algorithm. The time zones and total number of series that is used in our experiment Section 5 is presented in Fig. 3. We will use daily time unit in our experiments.

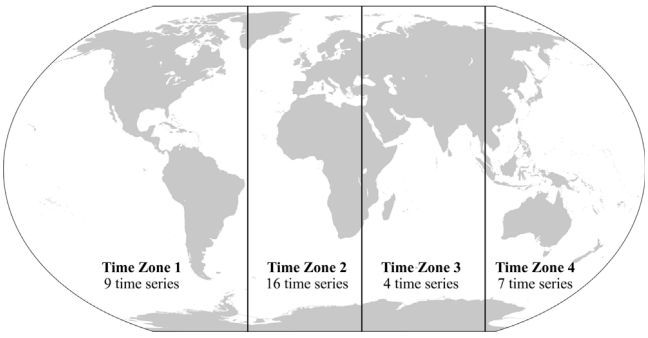

Fig. 3. Time zones and number of time series in each zone in our dataset.

For predicting one day ahead of markets in a time zone with known data from markets in another zone, the time zones are important and should be considered. For example, if an Australian market data is used to predict one time step ahead of a market in America zone, both data must be in the same day calendar. But if we want to predict Australian market using an American market data for one day ahead, if the American market is in day t, the Australian market data used, should be in day $t+1$. This phenomenon happens in daily time unit for time series due to the international date line which goes from North to South Pole in Pacific Ocean. The time zones of each two markets are checked before the exLift calculation for each two markets i and j in line 7 of the algorithm one. Adder variables are defined in line 4 to calculate exLift in Eq. (3) in two parts of loop. Time zones of two markets are the parameters to choose which loop part should be executed for exLift calculation in daily return time series. If the time unit is bigger than day (for example week), then one loop from lines 8 to 16 will be enough. At the end, four blocks of UU, UD, DU, and DD are put together to construct Ψ adjacency matrix of exLift network in line 21.

It is worth mentioning that in case of large networks with too many nodes it may be a good idea to filter the links that import low quality or may be noisy information to the prediction network. This can be done using filtering methods like threshold cutting of the links with lower weights or extracting maximum spanning tree out of the network. A good approach for filtering the edges has been presented in our previous work in Kia et al. (2018) called ConKruG. A way to filter the nodes is also presented in Section 3.3.

3.2. Prediction with DiMexRank

For prediction we modify and customize the personalized PageRank algorithm. In a directed network, label propagation can be done using a personalized PageRank method with an initial personalization vector that have node labels for corresponding known nodes elements in the vector. PageRank was presented in Page et al. (1999) and is widely used in search engines. It tries to estimate the probability of presence of a random walker in a node of a graph, given that the random walker hops through the edges of the graph from one node to another with probability that is proportional to the number and weights of the out-links of each node. It may also go out of a node to a random node with a defined probability called damping factor. PageRank formula is described in Eq. (6). In Eq. (6) A is the adjacency matrix of the network and *sumout* is the diagonal matrix of sum of outdegrees. It is defined in an iterative method where P_t is the PageRanK of the nodes at time t. d is the damping factor and is set to 0.85 that is the probability of the random walker through the out-links and by what probability $(1-d)$ it teleports from the node it is now on to any different random node. The teleportation vector or P_0 can be set to all ones or to the initial preferred ranks of the nodes in our case. Having initial ranks in the P_0 lets the PageRanK to propagate the initial rank of the nodes to other nodes in the network. In Eq. (6) M is the stochastic matrix of transitions in network which is calculated from adjacency matrix divided to the sum of rows or columns (depending on the way the PageRank equation is written).

The solution to the PageRanK can be found with algebraic method in small networks, and can also be found with an iterative solution. Both these solutions are presented in Eq. (7). The algebraic equation is solvable for M matrices that can efficiently be inverted. Matrices of graphs with too many nodes and edges cannot be solved with algebraic method in an efficient time. Therefore the iterative method is used. In iterative method the iterative PageRank formula in Eq. (6) repeats until a tolerable error is reached in Eq. (7) like ε. This iterative method has been proved to converge to a unique solution. Page et al. (1999) showed that in a network with millions of edges, the PageRank algorithm converges to the unique solution in about 52 iterations.

PageRank has been applied in many different fields to analyze directed networks. By the novel exLift network design and a modification to the personalized PageRank we found a way to use it for prediction of movement direction in markets.

$$P_t = d\,M P_{t-1} + \frac{1-d}{n}\vec{P}_0 \qquad where:\ M = (sumout^{-1}A)^T \tag{6}$$

$$P = (I - d\,M)^{-1}\frac{1-d}{n}P_0 \qquad P_{t+1} - P_t < \varepsilon \tag{7}$$

For prediction of direction of movement in financial markets we constructed a network called exLift in Section 3.1. To use PageRank for prediction we need to use a modified personalized vector with our proposed exLift network. We call this method of prediction DiMexRank. The DiMexRank personalization vector has two parts: one for nodes that represent upward movement and the other part for downward movement nodes. The DiMexRank personalization vector is presented in Eq. (8). It is shown that the personalized vector should be set to one in each element of up direction when a market has upward direction and its movement is known in our model. If the market's direction is downward then the downward section of the personalized vector (the second part in Eq. (8)) should be set to one for each element that its corresponding market direction is known to us. In our model we assume that we know all the directions of the America time zone markets and we will predict other markets with DiMexRank using these known data.

$$Init = [\overbrace{\underbrace{m_1^+, m_2^+, \ldots, m_k^+,}_{\text{(k) known directions}}\ \underbrace{0, 0, \ldots, 0,}_{\text{(n - k) unknown directions}}}^{\text{up direction nodes}}$$
$$\overbrace{\underbrace{m_1^-, m_2^-, \ldots, m_k^-,}_{\text{(k) known directions}}\ \underbrace{0, 0, \ldots, 0}_{\text{(n- k) unknown directions}}}^{\text{down direction nodes}}] \tag{8}$$
$$(m_i^{(+/-)}\ means:\ rank\ of\ market\ i)$$
$$\textbf{At first step}: if\ direction(market_i) == UP\ then\ m_i^+ = 1,\ m_i^- = 0\ else\ m_i^+ = 0, m_i^- = 1$$

In Eq. (9), the correspondence could be seen between DiMexRank and PageRank in Eq. (6). It implies that the DiMexRank solution also converges like PageRank.

$$P_t = d\,D P_{t-1} + \frac{1-d}{n}Init \qquad M = D,\ P_0 = Init,\quad d: damping\ factor \tag{9}$$

After DiMexRank converges to the solution as stated in Eq. (10), the final ranks will be presented in P. As it is shown, P has two equal parts for upward and downward movement of the markets same as

Pseudo-code of exLift network construction algorithm (Training Phase of DiMexRank).

Algorithm 1. PageRank-Based **Di**rection of **M**ovement prediction with **ex**tended Lift Network
DiMexRank (Train Phase)

Inputs: 1. Train Dataset (A Table of d rows and n columns)
d : $days$, n : $number\ of\ time\ series$
2. Time zones of time series in the dataset
1 : $America\ Continent$, 2 : $Europe\ and\ Africa$, 3 : $Russia,\ Iran,\ and\ India$, 4 : $East\ Asia\ and\ Australia$
Output: Ψ: Adjacency matrix of complete directed graph (exLift Network)
Ψ is constructed from four block matrices of $UU, UD, DU, and\ DD$

1: Assume UU, UD, DU, DD as $n \times n$ empty matrices

2: for i in time series:
3: for j in time series:

4: $uu, ud, du, dd = 0$
5: $ups_i = \sum_t \delta(series_{i,t,+1})|return_{i,t}|,\ downs_i = \sum_t \delta(series_{i,t,-1})|return_{i,t}|$
6: $ups_j = \sum_t \delta(series_{j,t,+1})|return_{j,t}|,\ downs_j = \sum_t \delta(series_{j,t,-1})|return_{j,t}|$

7: if $timezone[series_i] \leq timezone[series_j]$:
8: for day in $days - 1$: // $days$ is the number of days (rows) in dataset
9: if $(return_{i,day} > 0) \wedge (return_{j,day+1} > 0)$:
10: $uu = uu + |return_{j,day+1}|$
11: if $(return_{i,day} > 0) \wedge (return_{j,day+1} < 0)$:
12: $ud = ud + |return_{j,day+1}|$
13: if $(return_{i,day} < 0) \wedge (return_{j,day+1} > 0)$:
14: $du = du + |return_{j,day+1}|$
15: if $(return_{i,day} < 0) \wedge (return_{j,day+1} < 0)$:
16: $dd = dd + |return_{j,day+1}|$
17: else:
18: for day in $days - 1$:
19: if $(return_{i,day} > 0) \wedge (return_{j,day} > 0)$:
20: $uu = uu + |return_{j,day}|$
21: if $(return_{i,day} > 0) \wedge (return_{j,day} < 0)$:
22: $ud = ud + |return_{j,day}|$
23: if $(return_{i,day} < 0) \wedge (return_{j,day} > 0)$:
24: $du = du + |return_{j,day}|$
25: if $(return_{i,day} < 0) \wedge (return_{j,day} < 0)$:
26: $dd = dd + |return_{j,day}|$

27: $UU_{i,j} = \frac{uu}{ups_i \times ups_j},\quad UD_{i,j} = \frac{ud}{ups_i \times downs_j}$
28: $DU_{i,j} = \frac{du}{downs_i \times ups_j},\quad DD_{i,j} = \frac{dd}{downs_i \times downs_j}$

29: $\Psi = \begin{pmatrix} UU & UD \\ DU & DD \end{pmatrix}$

30: return Ψ

initial personalized vector in Eq. (8). For the final prediction, the rank of downward movements or the right part of the P should be subtracted from the left part of P, the rank of upward movement of markets. Then the sign of this subtraction shows the predicted label. If the sign is positive the corresponding market is predicted to have an upward movement in next time step, else it is predicted to have a downward direction of movement.

$$P = [\overbrace{\underbrace{m_1^+, m_2^+, \ldots, m_k^+}_{\text{up known ranks}}, \underbrace{m_{k+1}^+, \ldots, m_n^+}_{\text{UP: up unknown ranks}}}^{\text{up direction ranks}}, \overbrace{\underbrace{m_1^-, m_2^-, \ldots, m_k^-}_{\text{down known ranks}}, \underbrace{m_{k+1}^-, \ldots, m_n^-}_{\text{DOWN: unknown directions ranks}}}^{\text{down direction ranks}}] \tag{10}$$

$$Result = sign(UP - DOWN)$$

$$if\ Result = +1 \implies Class\ Label = UP\ otherwise: Class\ Label = DOWN$$

Algorithm 2 presents the DiMexRanK prediction phase in detail. Inputs of the algorithm are a test set with n time series for d continuous time steps, the exLift network that was the output of algorithm 1 (the DiMexRank training phase), and k, the number of markets, of a certain time zone, whose directions are known and are going to be used to predict other markets directions. The output of the algorithm will be predictions and performance of the prediction in each day of the test set for markets with unknown direction of movements. In line 1 to 9 of the algorithm, the prediction method using DiMexRank is done for each day as explained before. In line 10 the output prediction results matrix is constructed. The hit score or any other performance evaluation method can be calculated in the next steps of the algorithm. In line 11, the hit score or accuracy of the whole model is calculated. In line 12, the hit score for each market is calculated and returned in line 13.

3.3. Mixture of experts model

The mixture of experts or system of systems approach in financial prediction has been studied recently in some researches (Nguyen and Chamroukhi, 2018; Curry et al., 2018; Masoudnia and Ebrahimpour, 2014). There is a philosophy behind using a mixture of models in market prediction: according to all the surveys and literature reviews that many of them were mentioned in introduction section of this paper, there are different models with different structures that show superiority in some markets and not in all markets. In semi-supervised learning like the model we introduced in DiMexRank, the emphasis is on the environmental data or the data of other markets to predict any market. This will be shown in experiments section to be a good approach most of the times. But in some markets, using a market's own historical data achieves better performance in prediction rather

Pseudo-code of DiMexRank algorithm (Prediction Phase).

Algorithm 2. PageRank-Based **Di**rection of **M**ovement prediction with **ex**tended Lift Network
DiMexRank (Prediction Phase)

Inputs: 1) Test Dataset (A Table of d rows and n columns)
d : $days$, n : $number\ of\ time\ series$
2) exLift Network //from DiMexRank Training Phase
3) k: number of time series with known direction of movement
Output: $performance$**: A list of $n-k$ length to store accuracies for predicted time series in test set**

1: for day in Test Dataset:

2: $Init_{left} = int(Test_{day} > 0)$ //Changes the return value to +1 for positives and 0 for non-positives
3: $Init_{left}[k\ to\ n] = 0$ //Zero being the first index of the vector, the first series that should be predicted will be in k index
4: $Init_{right} = int(Test_{day} \leq 0)$
5: $Init_{right}[k\ to\ n] = 0$
6: $Init = [Init_{left}, Init_{right}]$ //Personalization vector for the DiMexRank
7: $P = pagerank(exLift, Init)$ //Standard PageRanK implementation using exLift adjacency matrix and DiMexRank modified vector as input
8: $R = sign(P[0\ to\ n] - P[n\ to\ 2n])$ //sign() function used here outputs 1/0 for positives and non-positives
9: $Result_{day} = R[k\ to\ n]$ //Each element of $Result$ vector is +1 or 0 which shows the UP or DOWN prediction for the series

10: $Output = \begin{pmatrix} Result_1 \\ Result_2 \\ \dots \\ Result_n \end{pmatrix}$, $Target = sign(Test > 0)$

11: $Hits = int(Output == Target)$ //Hits shows in what $(day, series)$ our model predicts correctly
12: $performance_i = average\ in\ column\ i\ of\ Hits$

13: return $performance = [performance_k, performance_{k+1}, \dots performance_{n-1}]$ //$(n-k)$ accuracies for $(n-k)$ series predicted by DiMexRank

than using the environmental data of other markets. This should be examined in validation dataset and the markets that are predicted better with supervised learners should be discovered. These markets will be called $candidates$. These candidates are discovered in phase one of the **Mix**ture of experts model using **DiMex**Rank and another supervised predictor (MixDiMex).

The MixDiMex has three phases. Phase one is the model selection. Phase two is for feature selection and phase three is for testing the model. An overall look of the all phases is presented in 4.

Considering the phase one of the MixDiMex that is presented in Fig. 4, performance comparator is the gating network that decides which expert (supervised or DiMexRank semi-supervised) to use in the next phases for each market prediction. This phase of the algorithm is the model selection phase.

After discovering the $candidates$ (markets that are predicted better with a supervised model rather than DiMexRank in validation set), the next phase is to check if their presence in the exLift network is useful for prediction of other markets or not. This phase of the algorithm is the wrapper feature selection phase. In wrapper feature selection methods, features that can help the performance of a model remain in the data and other features are eliminated. In this phase (phase two) predictions will be done using DiMexRank both with and without the candidate markets. The performances are compared at the end and the decision of having candidate markets in exLift network will be decided. The flowchart of the phase two is presented in 4. This phase can also be seen as a useful tool to see whether a feature (a market's data) is helpful in prediction of some other markets or not.

Final phase of the MixDiMex is to predict each market using the best model (supervised or semi-supervised) and calculating the prediction performances one by one and for the whole mixture of experts model. The flowchart of the phase three is also presented in 4. Both phase one and two were done in validation set but final phase is done in test set. The DiMexRank and exLift network are learned with or without the candidates according to the decision made in phase two and then the prediction is made.

The algorithm 3 explains the MixDiMex in detail. The inputs are the whole dataset including train, validation, and test parts. The parameter k, like before, shows how many of the markets are in the known time zone and their direction of movement is known to us so that other networks can be predicted using their direction. The final result of the MixDiMex will be the predictions of all markets with the performance of the models both for the whole markets and for each market alone.

line 1 to 6 of algorithm 3, represent the first phase of MixDiMex. Line 6 is the gating network that decides which model (supervised or semi-supervised) should be used in the next steps for each market after checking which model performs better. Line 7 to 13 describes the second phase of MixDiMex algorithm. In line 10 a flag is defined to be set to true if there is no increase in prediction performance using the data of candidate markets (markets that are better predicted by supervised models). When the flag is true the exLift network excludes the data of the candidate markets and a knowledge is also extracted that tells the market researchers the data of candidate markets are not useful in predicting other non-candidate markets at least in the DiMexRanK semi-supervised prediction. From line 13 till the end of algorithm 3, the final predictions in test set are made using the best model.

It should be mentioned that the training, validation, and test sets are the same as those in exLift network and supervised prediction methods: If the dataset is presented as in Fig. 1, where all the time series are columns of a matrix and the rows representing days, the first α% of the rows of the dataset matrix will be used as training set, the second β% as the validation set and the third $100--\alpha--\beta$% as the test. The validation set which is obviously a sub-block of the dataset matrix was shown to be used as the test set for the phase one and two of the MixDiMex model to compare between different supervised and semi-supervised models and to find the nodes (markets) that were helpful in prediction performance of the network model. In our work α and β were set to 0.7 and 0.2 of the whole dataset rows like Kia et al. (2018).

4. Complexity analysis and comparison

In this section we show that our presented models, DiMexRank and MixDiMex are faster than the baseline models regarding computational complexity.

We have discussed the running time performance in the evaluation Section 5.2 because unlike the complexity, the running time of the algorithms are dependent to the dataset volume.

All models have two computational complexities: one for learning (training) phase and one for prediction (test) phase. DiMexRank's learning phase or exLift network construction in algorithm 1 has two nested

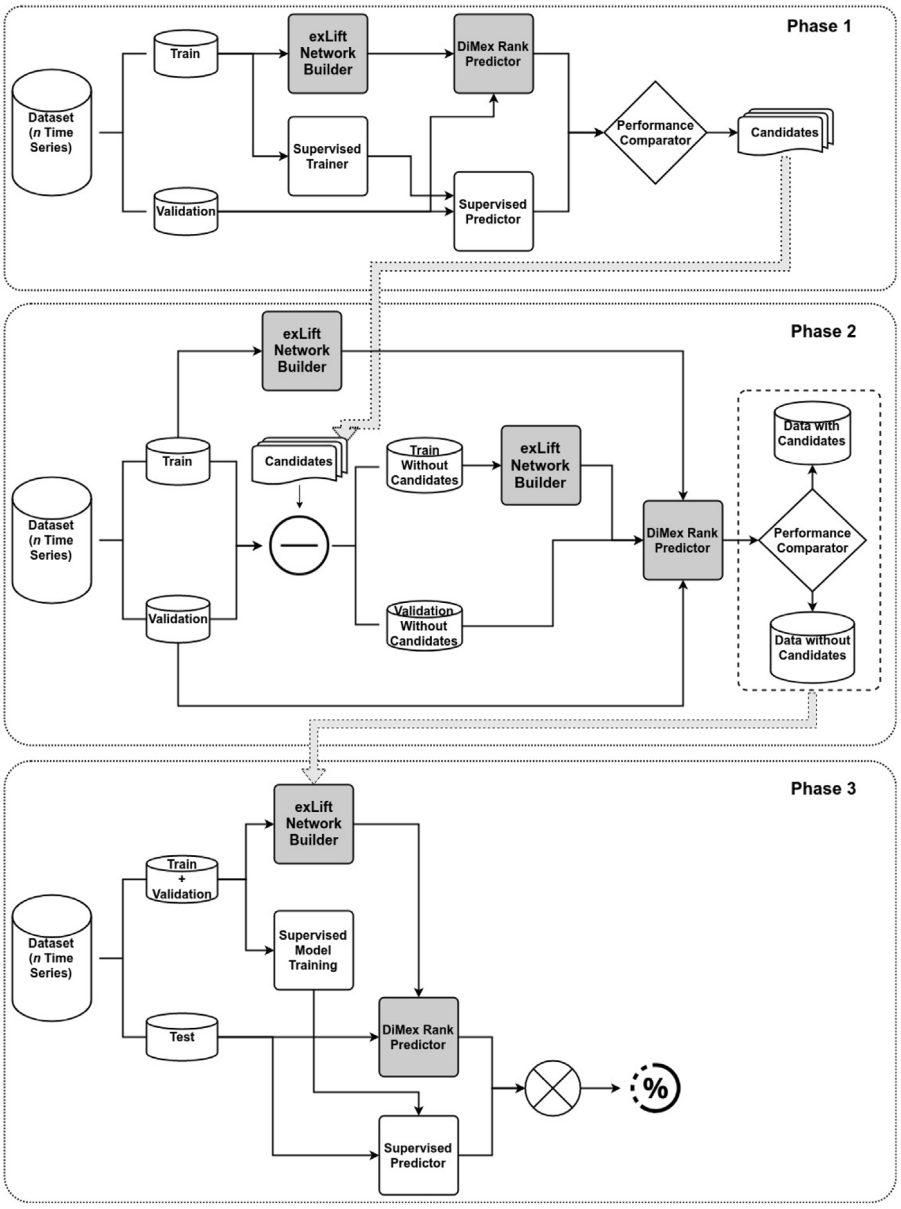

Fig. 4. Mixture of experts with Direction of Movement in extend Lift Network model and supervised modeling procedure (MixDiMex).

loops for calculating the exLift of rules like $i^{(+/-)} \implies j^{(+/-)}$ for each pair of $(i^{(+/-)}, j^{(+/-)})$. Therefore the complexity order for learning phase of DiMexRank will be $O(n^2)$ if n is the number of markets.

The computational complexity for testing phase of the DiMexRank will be the same as a personalized PageRanK because modified parts of the DiMexRank that differs from the PageRank have $O(1)$. The order of personalized PageRank, when solved with iterative method like Eq. (7) will be $O(te)$ where t is the number of iterations and e the number edges in the network. As discussed in Section 3.2 the number of iterations is about 50 for a huge graph of internet web pages. Therefore t can be ignored in the computational complexity calculation process. e or the number of edges in the exLift network of markets will be $2n(2n-1)/2$ when n is the number of networks because we have 2 nodes for upward/downward movement of each market and the graph is complete. Therefore the computational complexity order of DiMexRank test phase will be $O(n^2)$.

Now for finding the MixDiMex algorithm's computational complexity we should get the maximum of complexity orders in both train and test phases between the supervised model and the DiMexRank model. Therefore the complexity order of both training and testing phase in MixDiMex will be $Max(O(n^2), O(Supervised_{train/test}))$.

It will be seen that the best rival models in terms of prediction performance comparing to our presented models in this paper are HyS3 and ConKruG models. Their computational complexity is calculated in Kia et al. (2018). The complexity order of ConKruG in train and test phase are respectively $O(n^4)$, and $O(n^2)$. The complexity order of HyS3 in train and test phase are respectively $Max(O(train_{ConKruG}/test_{ConKruG}), O(Supervised_{train/test}))$. It is clear now that our MixDiMex and DiMexRank proposed algorithms in this paper are n^2 times faster in training phase and same in test phase comparing to their best rival models HyS3 and ConKruG.

5. Experiments

In this section final prediction performance results are presented. First we describe the dataset the data preparation phase. After that, the evaluation methods are explained and finally the results of our models and baseline models are presented, explained, and discussed.

Pseudo-code of MixDiMex algorithm.

Algorithm 3. Mixture of experts: **DiMex**Rank and Supervised Prediction, **(MixDiMex)**

Inputs: 1) Dataset (Train, Validation, and Test)
2) k: number of time series with known direction of movement
Output: $performance$: A list of $n-k$ size to store accuracies for predicted time series in test set

1: //**Phase One of the MixDiMex** Finding which series are better predicted with their own historical data
2: $network = exLiftNet(Train, TimeZones)$
3: $supervisedModel = supervisedTrainer(Train, [Supervised\ \ model\ \ parameters])$
4: $rankPerformance = DiMexRank(Validation, network, k)$
5: $supervisedPerformance = supervisedModel(Validation)$
6: $candidates = where\ \ supervisedPerformance > rankPerformance$

7: //**Phase Two of the MixDiMex** Checking if candidate series are helpful in prediction of other series or should be eliminated
8: $network_{without\ \ candidates} = exLiftNet(Train_{without\ \ candidates}, TimeZones_{without\ \ candidates})$
9: $rankPerformance_{without\ \ candidates} = DiMexRank(Validation_{without\ \ candidates}, network_{without\ \ candidates}, k)$

10: $flag = False$
11: $If\ \ mean(rankPerformance_{without\ \ candidates}) > mean(rankPerformance)$:
12: $flag = True$

13: //**Phase Three of the MixDiMex** Producing the output of prediction and calculating performance for the mixture of experts model
14: $TrainSet = [Train + Validation]$
15: $if\ \ flag$:
16: $TrainSet_{exLiftNet} = TrainSet - [candidates]$
17: $TestSet_{DiMexRank} = Test - [candidates]$
18: $TimeZones_{exLiftNet} = TimeZones - [candidates]$
19: $else$:
20: $TrainSet_{exLiftNet} = TrainSet$
21: $TestSet_{DiMexRank} = Test$
22: $TimeZones_{exLiftNet} = TimeZones$

23: $network = exLiftNet(TrainSet_{exLiftNet}, TimeZones_{exLiftNet})$
24: $supervisedModel = supervisedTrainer(TrainSet, [Supervised\ \ model\ \ parameteres])$
25: $performance_{DiMexRank} = DiMexRank(TestSet_{DiMexRankK}, network, k)$
26: $performance_{supervised} = supervisedModel(Test)$

27: $performance_{MixDiMex} = [candidates\ \ performance\ \ from\ \ performance_{supervised} + other\ \ series\ \ results\ \ from\ \ performance_{DiMexRank}]$

28: return $performance_{MixDiMex}, flag$ //flag shows if the candidate series are helpful in prediction of other series or not

5.1. Data gathering and preparation

All the time series sources are presented in Appendix A. All these data are gathered from open-source resources in the Internet. The dataset consists of three of most famous oil prices of West Texas Intermediate, Brent, and OPEC, gold price of 10AM London, and 32 famous stock market indices all over the world. The time series are transformed to return series using the formula presented in Eq. (2). The return series are shown in Fig. 5 in Appendix B.

5.2. Evaluation method

The first superiority of our presented models, DiMexRank and MixDiMex are in computational complexity that was discussed in Section 4. This superiority has made the running time of our models lower than their best rival models with the highest prediction performance. With a core i7 system with 4MG CPU cache, 12GB of DDR3 ram, 512 SSD hard drive, and Ubuntu 16.04 installed, the DiMexRank ran in 12 s. The MixDiMex ran in 1 min and 21 s. The running time of HyS3 and ConKruG as the best rival models with the same mentioned hardware configuration were respectively 20 min and 58 s, and 9 min and 11 s. It is notable that MixDiMex as our best performance proposed algorithm in the graph-based semi-supervised models works about 16 times faster than its best rival HyS3.

For prediction performance we will calculate the hit score as the number of times each model predicts the direction of movement correct divided to all the predictions done by the model. Hit score is like accuracy in machine learning models where all true positives (TP) and true negatives (TN) are divided to total number of samples according to Eq. (11). Hit score has been used as the most important evaluation criterion regarding the literature of direction prediction (Atsalakis and Valavanis, 2009). To be a fair evaluation metric, hit score needs data to be balanced. The data being balanced is measured with a criterion called imbalance ratio (IR) that is calculated by the number of instances in majority class divided by the number of instances in minority class, as in equation ($IR = {}^{\#Majority\ Class}/_{\#Minority\ Class}$). In our direction prediction problem, the two classes are downward movement and upward movement. A dataset with imbalance ratio less than 1.5 is usually considered to be balanced (Fernández et al., 2008). Our dataset imbalance ratio for the whole dataset and each market series are presented in Appendix B. It can be seen that our dataset is balanced. There is also no priority for us between predicting upward or downward directions. Therefore the Hit score or accuracy is the best evaluation parameter in this research. In Appendix B it can be seen that only two markets (TEPIX and JSE) are not balanced. Table 3 shows that those two markets are better predicted both with supervised models and our MixDiMex model. It is notable that we did not claim being explicitly superior in prediction in those two imbalance stock markets.

$$\begin{aligned} Hit\ Score &= \frac{\#Correct\ direction\ prediction}{\#All\ predictions} \\ Accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \\ Hit\ Score &= Accuracy \end{aligned} \tag{11}$$

The prediction performance differences between different models provided in the next section is also checked to be statistically significant with paired T-Test, assuming H_0 hypothesis being performances of two most accurate models are equal to MixDiMex as our best proposed model. The t-test results are presented in Table 5 and discussed in Section 5.3.

Table 1
Search domains for different parameters of models being compared. These parameters are set by search in the validation set to avoid overfitting.

Parameter	Model(s)	Search Domain
Damping factor (d)	PageRank & DiMex Rank	0.85 according to (Page et al., 1999)
Delay	Supervised Models	$\{1, 2, 3, \dots, 30, 31, 60, 62, 90, 93, 365, 366\}$
$\#estimators$	Random Forest	$\{10, 100, 1000, 10000\}$
α	All graph-based models except Park & Shin label propagation method	0.99 according to (Zhou et al., 2004)
σ	All graph-based models except complete graph with all weights = 1	$\{0.1, 0.2, 0.3, \dots, 3.0\}$
μ	Park & Shin model with label propagation	$\{0.01, 0.1, 0.3, 0.5, 0.7, 1, 10, 100\}$ (Park and Shin, 2013)
K	For K-NN in all Park & Shin models	$\{2, 3, 4, 5\}$ (Park and Shin, 2013)

5.3. Results

In this section we compare our model to the best rival models in the literature of graph-based semi-supervised prediction models and random forest as the best performance supervised model in our dataset. These baseline models are described in Section 2. Model parameters are derived from search in validation set to avoid overfitting. The used setting for the parameters are presented in Table 1. Some parameters like damping factor in PageRanK and α coefficient for label spreading are usually chosen from the literature and set to a widely-used value. We did the same in our research. $delay$ is the number of time-steps before that is used in supervised model (random forest) to predict next time step. $\#estimators$ is the number of decision trees used in random forest algorithm. σ is the variance and a weight adjuster in label propagation and spreading algorithms. μ is a parameter like α in label spreading or d in PageRank which tries to establish a ratio between the importance of the network and the initial value of node labels in the network. K is the number of neighbors to be checked in KNN algorithm used for filtering the graph in Park and Shin (2013) work.

The results for DiMexRank, random forest supervised model, and MixDiMex are presented in Table 2 as results of different phases of MixDiMex model. First column shows the abbreviations used for the name of stock market indices and commodity prices time series. The abbreviations and their corresponding full names are presented in Appendix A. The validation columns performances are presented for comparison between supervised and DiMexRank semi-supervised model as the first phase of MixDiMex algorithm presented in Section 3.3. The comparison column of Table 2 is *True* for any market that supervised prediction with historical data of the market has shown a better prediction performance compared to DiMexRank semi-supervised model. In first two columns of test part of Table 2, the results of phase two of MixDiMex algorithm is presented. The first column of the test part in table shows the DiMexRank prediction results when all the markets are used in exLift network building (learning phase of DiMexRank). The second column of test part in Table 2 shows the prediction results of DiMexRank when the candidate markets (the markets with better prediction in supervised model) are eliminated from the exLift network building phase. The average performance of the first and second column of test part shows if the candidates are helpful in predicting other markets or not. In our case the markets had to be eliminated in final phase of MixDiMex because their existence in exLift network was not helpful. The final results of the MixDiMex are the supervised results for the candidate markets and DiMexRank results for other markets.

The comparison information for different baseline models and our proposed models is presented in Table 3. The last column of the table shows the best model for prediction of each market. Table 4 shows how many times each model has outperformed other models in prediction. It is clear in Tables 3 and 4 that both our proposed semi-supervised models, DiMexRank and mixture of experts model (MixDiMex) outperform other models in terms of hit score. We saw in Section 4 that they also outperform their best rivals in terms of computational complexity and running time. Table 3 shows that each of prediction models outperforms others in at least one market. This may be due to the different natures of the markets that result in different features for their time series and different models for the prediction. Again, this is the fact that lad us to the idea of using a mixture of experts or system of systems idea.

Table 2
Hit score prediction performance of models during MixDiMex different phases. In our experiment, random forest was chosen as the best supervised model in our dataset.

Markets	Validation		Comparison	Test			
	DiMexRank	Supervised		DiMexRank	DiMexRank (No Candidates)	Supervised	MixDiMex
FTSE	56.45%	51.58%	FALSE	58.82%	58.40%	52.66%	58.40%
Euronext	56.03%	48.64%	FALSE	55.46%	55.46%	47.34%	55.46%
Eurostox	56.03%	48.42%	FALSE	55.04%	54.62%	49.76%	54.62%
CAC	56.66%	47.96%	FALSE	54.20%	57.98%	45.41%	57.98%
AEX	54.97%	50.00%	FALSE	57.98%	56.72%	48.31%	56.72%
BEL	54.55%	49.55%	FALSE	57.14%	55.88%	50.24%	55.88%
DAX	54.55%	48.42%	FALSE	55.46%	54.62%	49.76%	54.62%
SMI	54.33%	51.36%	FALSE	55.04%	49.58%	57.49%	49.58%
OMX	57.29%	48.42%	FALSE	53.78%	57.14%	48.31%	57.14%
IBEX	55.18%	50.00%	FALSE	48.74%	57.14%	54.11%	57.14%
ATX	54.97%	47.06%	FALSE	57.56%	48.74%	51.21%	48.74%
BIST	59.41%	50.68%	FALSE	56.30%	62.18%	52.66%	62.18%
JSE	57.08%	80.54%	TRUE	50.00%	–	63.29%	63.29%
Brent	55.81%	47.51%	FALSE	57.14%	54.62%	47.34%	54.62%
OPEC	64.06%	51.81%	FALSE	61.76%	58.40%	56.52%	58.40%
GOLD	53.49%	51.81%	FALSE	54.62%	60.50%	56.04%	60.50%
RTS	57.08%	52.04%	FALSE	58.40%	60.50%	52.66%	60.50%
TEPIX	68.50%	76.02%	TRUE	60.50%	–	79.23%	79.23%
BSE	57.72%	48.87%	FALSE	60.50%	62.61%	50.72%	62.61%
CNX	58.77%	49.55%	FALSE	61.34%	58.82%	52.66%	58.82%
HSI	64.06%	45.93%	FALSE	57.98%	55.46%	48.79%	55.46%
SSE	55.18%	54.98%	FALSE	52.52%	51.68%	49.28%	51.68%
Shen	52.85%	54.30%	TRUE	51.68%	–	54.59%	54.59%
TAIEX	65.96%	49.32%	FALSE	59.24%	59.66%	58.94%	59.66%
KOSPI	65.96%	52.71%	FALSE	58.40%	58.40%	49.28%	58.40%
NIK	60.68%	54.30%	FALSE	67.23%	67.23%	50.24%	67.23%
ASX	59.62%	54.98%	FALSE	56.72%	56.30%	51.69%	56.30%
Average	**58.05%**	**52.47%**	-	**56.80%**	**57.20%**	**52.91%**	**58.14%**

Table 3
Comparison of different models in prediction hit score. RF refers to random forest. Enhanced Park & Shin Model is our improvement to Park & Shin model using label spreading algorithm. The proposed models in our paper are DiMexRank and MixDiMex.

Markets	Park & Shin Model	Enhanced Park & Shin	ConKruG	HyS3	DiMexRank	RF	MixDiMex	Best Model
FTSE	54.70%	44.87%	60.08%	60.34%	58.82%	52.66%	58.40%	HyS3
Euronext	46.15%	71.37%	49.58%	64.56%	55.46%	47.34%	55.46%	Enhanced Park & Shin
Eurostox	45.73%	59.83%	58.82%	58.23%	55.04%	49.76%	54.62%	Enhanced Park & Shin
CAC	64.96%	48.72%	50.84%	50.63%	54.20%	45.41%	57.98%	Park & Shin Model
AEX	49.57%	54.27%	67.23%	67.09%	57.98%	48.31%	56.72%	ConKruG
BEL	47.44%	57.69%	53.36%	53.59%	57.14%	50.24%	55.88%	Enhanced Park & Shin
DAX	52.99%	48.72%	54.20%	54.85%	55.46%	49.76%	54.62%	DiMexRank
SMI	53.42%	52.56%	55.46%	55.70%	55.04%	57.49%	49.58%	RF
OMX	50.43%	53.85%	52.94%	52.32%	53.78%	48.31%	57.14%	MixDiMex
IBEX	50.00%	55.13%	60.08%	59.49%	48.74%	54.11%	57.14%	ConKruG
ATX	49.15%	56.41%	55.04%	55.27%	57.56%	51.21%	48.74%	DiMexRank
BIST	54.27%	52.99%	55.04%	55.27%	56.30%	52.66%	62.18%	MixDiMex
JSE	52.14%	53.85%	55.04%	54.85%	50.00%	63.29%	63.29%	MixDiMex, RF
Brent	50.43%	54.27%	55.04%	55.27%	57.14%	47.34%	54.62%	DiMexRank
OPEC	52.56%	58.97%	54.62%	54.01%	61.76%	56.52%	58.40%	DiMexRank
GOLD	49.15%	52.99%	58.82%	58.23%	54.62%	56.04%	60.50%	MixDiMex
RTS	60.26%	50.85%	53.78%	53.16%	58.40%	52.66%	60.50%	MixDiMex
TEPIX	48.29%	49.57%	56.30%	55.70%	60.50%	79.23%	79.23%	MixDiMex, RF
BSE	45.30%	59.83%	49.58%	49.79%	60.50%	50.72%	62.61%	MixDiMex
CNX	47.86%	60.68%	55.46%	55.27%	61.34%	52.66%	58.82%	DiMexRank
HSI	49.57%	55.56%	47.90%	60.34%	57.98%	48.79%	55.46%	HyS3
SSE	46.15%	60.26%	60.50%	60.76%	52.52%	49.28%	51.68%	HyS3
Shen	50.85%	65.81%	60.50%	60.76%	51.68%	54.59%	54.59%	HyS3
TAIEX	44.02%	53.42%	57.56%	57.81%	59.24%	58.94%	59.66%	MixedDiMex
KOSPI	44.44%	55.98%	52.10%	51.90%	58.40%	49.28%	58.40%	MixDiMex, DiMexRank
NIK	50.43%	51.28%	50.42%	50.21%	67.23%	50.24%	67.23%	MixDiMex, DiMexRank
ASX	51.28%	54.70%	58.82%	57.38%	56.72%	51.69%	56.30%	ConKruG
AVERAGE	50.43%	55.35%	55.52%	56.40%	56.80%	52.91%	58.14%	–

Table 4
Number of times a model outperformed other models in different markets time series prediction. The superiority of our proposed models are evident. RF refers to random forest as the best candidate for supervised models in our data. Enhanced Park & Shin is the Park & Shin model with our enhancement using label spreading algorithm in our previous work.

Model	MixDiMex	DiMexRank	HyS3	Enhanced Park & Shin	Supervised (RF)	ConKruG	Park & Shin
Outperformed	10	7	4	3	3	3	1

Table 5
Paired T-test statistics and P-Value showing the statistically significance of our best model results comparing to its best rival models.

Paired T-Test	HyS3	DiMexRank
MixDiMex	Statistic = −2.58, p-value = 0.001 < 0.05	Statistic = −8.19, p-value = 3.14e−16 < 0.05

5.4. Discussion

Table 4 shows that our proposed models outperform other semi-supervised graph-based predictors in the literature and better than the best supervised model found for our dataset in average (random forest). Most of the times (10 times) the MixDiMex had better prediction performance comparing to others and the next rank is for DiMexRank semi-supervised learner.

The difference between the actual values of the hit scores from Table 3 are tested to be statistically significant with the values of prediction for each sample in the test set. Samples in the test sets are used for the paired t-test described in Section 5.2. The t-test statistics and p-values of the comparison between the best model MixDiMex, DiMexRank and HyS3 are presented in Table 5. Both p-values are far less than 0.05 that is an accepted value refusing H_0 hypothesis explained in evaluation Section 5.2.

By using our constructed network and ranking algorithm, we are able to use other markets data to predict a market's change in direction. Our empirical studies show that markets own historical data used in statistical forecasting methods or machine learning supervised algorithms do not reach good results comparing to semi-supervised algorithms that use environmental data rather than historical data of the market itself.

Table 6
Advantages of our proposed models to previous graph-based semi-supervised prediction researches.

	Our Models: (DiMexRank and MixDiMex)	Previous Graph-Based Models
1	Less computational complexity and running time	Previous model with accuracy near our models (both DiMex and MixDiMex) run about **sixteen times slower**.
2	Gaining knowledge about the time series (in MixDiMex). It practically shows if any time series helps future prediction of other time series and if itself can be predicted better by other time series compared to its own historical data.	The closest model in performance to our model (HyS3), only shows if it helps to use the historical data of a time series comparing to usage of environmental data.
3	Both DiMexRank and MixDiMex have **higher accuracies (Hit score)** compared to previous models	The best model in the literature (HyS3) has less accuracy compared to our model. The difference of the performances is statistically significant.

Table 6 shows our superiority against the best models in the literature of semi-supervised graph-based financial prediction.

6. Conclusion and further researches

In this paper, we presented two prediction models: one semi-supervised graph-based model using an innovative network structure and one mixture of experts model that decided when to use historical data of the market itself and when to use historical data of other

Table 7
In this table the abbreviation of the market indices and commodities used in the research are presented with the full names of the time series, the dataset resource, and the time zones. The time zones are according to this set: {1: America, 2: Europe & Africa, 3: Asia & Russia, 4: Australia & East Asia}. The data resource addresses are as follows: Yahoo Finance Service: http://finance.yahoo.com, Google Finance Service: http://finance.google.com, Federal Reserve Bank of St. Luis. Economic Research Center Web Site: https://research.stlouisfed.org/fred2/, U.S. Energy Information Administration Web Site: http://tonto.eia.gov, Organization of Petroleum Exporting Countries Official Site: http://www.opec.org, Tehran Stock Exchange Official Site: http://www.tse.ir.

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

Table 8
Imbalance ratio for the whole dataset (Global IR) and for each market.

Markets	FTSE	Euronext	Eurostox	CAC	AEX	BEL	DAX	SMI	OMX	Global IR
IR	1.03	1.06	1.05	1.01	1.05	1.04	1.10	1.06	1.02	
Markets	IBEX	ATX	BIST	JSE	Brent	OPEC	GOLD	RTS	TEPIX	
IR	1.03	1.01	1.04	4.43	1.04	1.01	1.11	1.02	2.01	1.21
Markets	BSE	CNX	HSI	SSE	Shen	TAIEX	KOSPI	NIK	ASX	
IR	1.02	1.03	1.04	1.02	1.06	1.06	1.02	1.02	1.22	

markets for prediction of each market. We showed the ability of semi-supervised graph-based models in prediction that could beat supervised models and also previous models in the literature of semi-supervised graph-based learning. It was shown that in most of the time series examined, the recent information about other markets are better predictor features than the historical data of the market being predicted itself. Our proposed models in this research outperformed other models both in terms of computational complexity (and also running time), and prediction accuracy. Our MixDiMex model could also be a tool to find out whether a market's data is useful in predicting other markets future or not.

For future research we suggest to use networks that could model historical data of other markets that belong to before one time step ago. In our initial researches we have found out that using more than one step before in network construction, imports intense noise to the model and reduces prediction performance. Filtering the complete exLift graph may help in improving the prediction performance by eliminating the probable noise but using different filtering methods like the methods used in ConKruG and works of Park and Shin (2013) and Shin et al. (2013) has not been successful up to now. These filtering methods only reduced the prediction performance of our models. Adding other time series from different sources of data may also help the prediction of the model. We have tried to use most famous stock market indices and commodity prices but the path is open for practical researches in finding other useful information sources. Text mining has been used in many financial prediction models (Kumar and Ravi, 2016; Ruiz et al., 2012). Web mining is another popular method for finding new sources of information as input for prediction models (Nardo et al., 2016). Information gathered from text and web mining can change our proposed prediction models to online and stream predictors. Learning the turning points rather than direction of movement and changing the structure of the model to a 3-class classifier (for example: support, resistance, neutral classes instead of upward and downward movement classes) for time series can also be an interesting topic for future researches.

Appendix A

See Table 7.

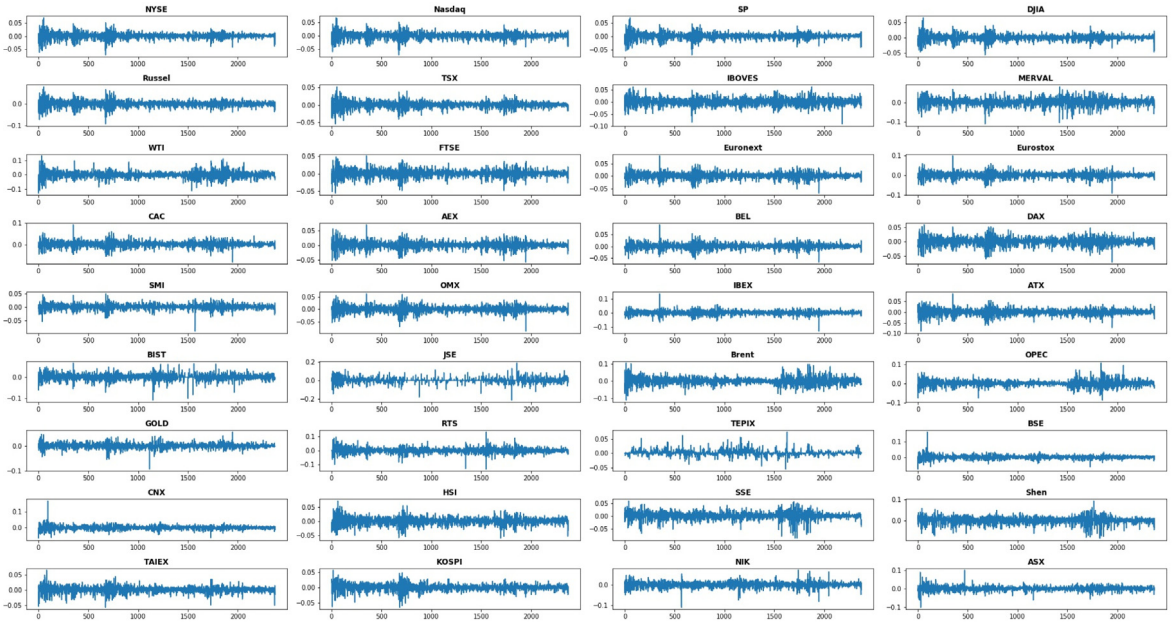

Fig. 5. Return series of all markets used in the experimental results section.

Appendix B. Return series of the markets and their imbalance ratio

See Fig. 5 and Table 8.

References

Abarbanell, J.S., Bushee, B.J., 1997. Fundamental analysis, future earnings, and stock prices. J. Account. Res. 35, 1–24.

Atsalakis, G., Valavanis, K., 2009. Surveying stock market forecasting techniques - part ii: Soft computing methods. Expert Syst. Appl. 36, 5932–5941.

Atsalakis, G., Valavanis, K., 2013. Surveying stock market forecasting techniques-Part I: Conventional methods. In: Zopounidis, C. (Ed.), Computation Optimization in Economics and Finance Research Compendium. Nova Science Publishers, Inc., New York.

Ballings, M., Van den Poel, D., Hespeels, N., Gryp, R., 2015. Evaluating multiple classifiers for stock price direction prediction. Exp. Syst. Appl. 42, 7046–7056.

Brin, S., Motwani, R., Ullman, J.D., Tsur, S., 1997. Dynamic itemset counting and implication rules for market basket data. ACM Sigmod Record 26, 255–264.

Cavalcante, R.C., Brasileiro, R.C., Souza, V.L., Nobrega, J.P., Oliveira, A.L., 2016. Computational intelligence and financial markets: A survey and future directions. Expert Syst. Appl. 55, 194–211.

Chen, Y.-J., Chen, Y.-M., Lu, C.L., 2017. Enhancement of stock market forecasting using an improved fundamental analysis-based approach. Soft Comput. 21, 3735–3757.

Curry, D.M., Beaver, W.W., Dagli, C.H., 2018. A system-of-systems approach to improving intelligent predictions and decisions in a time-series environment. In: 2018 13th Annual Conference on System of Systems Engineering (SoSE). IEEE, pp. 98–105.

Fama, E.F., Fisher, L., Jensen, M.C., Roll, R., 1969. The adjustment of stock prices to new information. Int. Econ. Rev. 10, 1–21.

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, 2378–2398.

Goldberg, A.B., Zhu, X., 2006. Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In: Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing. Association for Computational Linguistics, pp. 45–52.

Ho, T.K., 1995. Random decision forests. In: Document Analysis and Recognition, 1995, Proceedings of the Third International Conference on, Vol. 1. IEEE, pp. 278–282.

Huang, W., Nakamori, Y., Wang, S.-Y., 2005. Forecasting stock market movement direction with support vector machine. Comput. Oper. Res. 32, 2513–2522.

Imandoust, S.B., Bolandraftar, M., 2014. Forecasting the direction of stock market index movement using three data mining techniques: the case of tehran stock exchange. Int. J. Eng. Res. Appl. 4, 106–117.

Jovanovic, F., Schinckus, C., 2017. Econophysics and Financial Economics: An Emerging Dialogue. Oxford University Press.

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. Exp. Syst. Appl. 38, 5311–5319.

Kia, A.N., Haratizadeh, S., Shouraki, S.B., 2018. A hybrid supervised semi-supervised graph-based model to predict one-day ahead movement of global stock markets and commodity prices. Expert Syst. Appl. 105, 159–173.

Kruskal, J.B., 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. Proc. Amer. Math. Soc. 7, 48–50.

Kumar, B.S., Ravi, V., 2016. A survey of the applications of text mining in financial domain. Knowl.-Based Syst. 114, 128–147.

Li, W., Liao, J., 2017. A comparative study on trend forecasting approach for stock price time series. In: Anti-Counterfeiting, Security, and Identification (ASID), 2017 11th IEEE International Conference on. IEEE, pp. 74–78.

Lo, A.W., MacKinlay, A.C., 1988. Stock market prices do not follow random walks: Evidence from a simple specification test. Rev. Financ. Stud. 1, 41–66.

Malkiel, B.G., 2003. The efficient market hypothesis and its critics. J. Econ. Perspectives 17, 59–82.

Mantegna, R.N., Stanley, H.E., 1999. Introduction To Econophysics: Correlations and Complexity in Finance. Cambridge university press.

Masoudnia, S., Ebrahimpour, R., 2014. Mixture of experts: a literature survey. Artif. Intell. Rev. 42, 275–293.

McCauley, J.L., 2004. Dynamics of Markets: Econophysics and Finance. Cambridge University Press.

Nardo, M., Petracco-Giudici, M., Naltsidis, M., 2016. Walking down wall street with a tablet: A survey of stock market predictions using the web. J. Econ. Surv. 30, 356–369.

Naseer, M., Bin Tariq, Y., 2015. The efficient market hypothesis: A critical review of the literature. IUP J. Financ. Risk Manag. 12, 48–63.

Nguyen, H.D., Chamroukhi, F., 2018. Practical and theoretical aspects of mixture-of-experts modeling: An overview. In: Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. p. e1246.

Page, L., Brin, S., Motwani, R., Winograd, T., 1999. The PageRank Citation Ranking: Bringing Order To the Web. Technical Report, Stanford InfoLab.

Park, K., Shin, H., 2013. Stock price prediction based on a complex interrelation network of economic factors. Eng. Appl. Artif. Intell. 26, 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 Syst. Appl. 42, 259–268.

Pereira, E.J.d.A.L., d. Silva, M.F., Pereira, H., 2017. Econophysics: Past and present. Physica A 473, 251–261.

Preethi, G., Santhi, B., 2012. Stock market forecasting techniques: A survey. J. Theor. Appl. Inf. Technol. 46.

Rather, A.M., Sastry, V., Agarwal, A., 2017. Stock market prediction and portfolio selection models: a survey. OPSEARCH 54, 558–579.

Ruiz, E.J., Hristidis, V., Castillo, C., Gionis, A., Jaimes, A., 2012. Correlating financial time series with micro-blogging activity. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining. ACM, pp. 513–522.

Saumya, S., Singh, J.P., Kumar, P., 2016. Predicting stock movements using social network. In: Conference on E-Business, E-Services and E-Society. Springer, pp. 567–572.

Shen, K.-Y., Tzeng, G.-H., 2015. Combined soft computing model for value stock selection based on fundamental analysis. Appl. Soft Comput. 37, 142–155.

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. Decis. Support Syst. 55, 348–358.
Skabar, A., 2013. Direction-of-change financial time series forecasting using a similarity-based classification model. J. Forecast. 32, 409–422.
Soni, S., 2011. Applications of anns in stock market prediction: a survey. Int. J. Comput. Sci. Eng. Technol. 2, 71–83.
Timmermann, A., Granger, C.W., 2004. Efficient market hypothesis and forecasting. Int. J. Forecast. 20, 15–27.
Upstill, T., Craswell, N., Hawking, D., 2003. Predicting fame and fortune: Pagerank or indegree. In: Proceedings of the Australasian Document Computing Symposium. ADCS, pp. 31–40.
Verikas, A., Gelzinis, A., Bacauskiene, M., 2011. Mining data with random forests: A survey and results of new tests. Pattern Recogn. 44, 330–349.
Yan, X., Zheng, L., 2017. Fundamental analysis and the cross-section of stock returns: A data-mining approach. Rev. Financ. Stud. 30, 1382–1423.
Yao, J., Tan, C.L., 2000. A case study on using neural networks to perform technical forecasting of forex. Neurocomputing 34, 79–98.
Zhang, J., Cui, S., Xu, Y., Li, Q., Li, T., 2018. A novel data-driven stock price trend prediction system. Expert Systems with Applications 97, 60–69.
Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B., 2004. Learning with local and global consistency. In: Advances in Neural Information Processing Systems. pp. 321–328.
Zhu, X., 2006. Semi-supervised learning literature survey. Comput. Sci., University of Wisconsin-Madison 2 (4).
Zhu, X., Ghahramani, Z., 2002. Learning from labeled and unlabeled data with label propagation. Tech. Rep., Technical Report CMU-CALD-02–107, Carnegie Mellon University.