Knowledge and Information Systems (2024) 66:4497–4525
https://doi.org/10.1007/s10115-024-02095-6

REGULAR PAPER

Forecasting financial market structure from network features using machine learning

Douglas Castilho[1,5] · **Thársis T. P. Souza**[2] · **Soong Moon Kang**[3] · **João Gama**[4] · **André C. P. L. F. de Carvalho**[1]

Received: 20 December 2022 / Revised: 15 December 2023 / Accepted: 6 March 2024 /
Published online: 22 April 2024

Abstract

We propose a model that forecasts market correlation structure from link- and node-based financial network features using machine learning. For such, market structure is modeled as a dynamic asset network by quantifying time-dependent co-movement of asset price returns across company constituents of major global market indices. We provide empirical evidence using three different network filtering methods to estimate market structure, namely Dynamic Asset Graph, Dynamic Minimal Spanning Tree and Dynamic Threshold Networks. Experimental results show that the proposed model can forecast market structure with high predictive performance with up to 40% improvement over a time-invariant correlation-based benchmark. Non-pair-wise correlation features showed to be important compared to traditionally used pair-wise correlation measures for all markets studied, particularly in the long-term forecasting of stock market structure. Evidence is provided for stock constituents of the DAX30, EUROSTOXX50, FTSE100, HANGSENG50, NASDAQ100 and NIFTY50 market indices. Findings can be useful to improve portfolio selection and risk management methods, which commonly rely on a backward-looking covariance matrix to estimate portfolio risk.

Keywords Financial networks · Network link prediction · Information filtering networks · Correlation-based networks · Machine learning · Stock markets

✉ Douglas Castilho
douglas.braz@ifsuldeminas.edu.br

[1] Institute of Mathematics and Computer Sciences (ICMC), University of São Paulo (USP), São Carlos, Brazil

[2] Columbia University, New York, NY, USA

[3] School of Management, University College London, Gower Street, London WC1E 6BT, UK

[4] Institute for Systems and Computer Engineering, Technology and Science, University of Porto (UP), Porto, Portugal

[5] Laboratory of Technology and Innovation (LATIN), Federal Institute of South of Minas Gerais (IFSULDEMINAS), Poços de Caldas, Brazil

1 Introduction

Multi-asset financial analyses, particularly optimal portfolio selection and portfolio risk management, traditionally rely on the usage of a covariance matrix representative of market structure, which is commonly assumed to be time invariant. Under this assumption, however, non-stationarity [1, 2] and long-range memory [3] can lead to misleading conclusions and spoil the ability to explain future market structure dynamics.

Empirical analyses of networks in finance have been used successfully to study market structure dynamics, particularly to explain market interconnectedness from high-dimensional data [4–7]. Under this approach, market structure is modeled as a network whose nodes represent different financial assets and edges represent one or many types of relevant relationships among those assets. There is a vast literature applying financial networks to descriptive analysis of market and portfolio dynamics, including market stability [8], information extraction [9], asset allocation [10, 11] and dependency structure [4, 12–15]. However, there is little research on the application of financial networks in market structure forecasting. Recent research on market structure inference makes use of information filtering networks to produce a robust estimate of the global sparse inverse covariance matrix [16], achieving computationally efficient results. In a later study [17], the authors forecast market structure based on a model that uses a principle of link formation by triadic closure in stock market networks. Spelta [18] proposed a method to predict abrupt market changes, inferring the future dynamics of stock prices by predicting future distances between them, using a tensor decomposition technique. Musmeci et al. [19] proposed a new tool to predict future market volatility using correlation-based stock networks, meta-correlation and logistic regression. Park et al. [20] analyzed the evolution of Granger causality network of global currencies and proposed a link prediction method incorporating the squared eta of the causality directions of two nodes as the weight of future edges. To build the causality network, they used the effective exchange rate of 61 countries and showed that the predictive capacity of their model outperforms other static methods for predicting links. Other related work [21] proposed a model for predicting links in weighted financial networks, used to define input variables for the portfolio management problem, increasing the financial return of the investment.

In this article, financial market structure forecasting is formulated as a link prediction problem where we estimate the probability of adding or removing links in future networks. To tackle this problem, we developed a machine learning-based model that uses node- and link-specific financial network features to forecast stock to stock links based on past market structure. Applying machine learning algorithms in the decision-making process on stock markets is not a recent task [22]. An increasing number of applications have been created using machine learning-based models to predict the behavior of price time series [23], volatility forecasting [24], sentiment analysis for investment [25] and automatic trading rules [26]. This paper provides a set of empirical experiments designed to address the following research questions:

1. To what extent can dynamic financial networks help forecast stock market correlation structure?
2. How do financial network topology features perform relative to traditionally used pairwise correlation data to forecast stock market structure?
3. How does the predictability of market structure vary across multiple financial markets for the proposed models?

Findings can be particularly useful to improve portfolio selection and risk management, which commonly rely on a backward-looking correlation matrix to estimate portfolio risk. To the best of our knowledge, this is the first study that combines financial network features and machine learning to forecast stock market structure. The remainder of this paper is organized as follows: Sect. 3 describes the Materials and Methods used to provide the experiments; Sect. 4, which is the Results and Discussion, presents a descriptive analysis of the temporal stock networks and predictive analysis of market structure forecasting, and Sect. 5 draws the Conclusions.

2 Stock market structure and network prediction

We addressed the problem of market structure prediction as a link prediction problem. In order to do this, previously known network information was used to find connections that may appear or disappear in the future. This predictive task was investigated in many real problems, mainly involving social networks [27, 28]. Mantegna et al. [4] introduced a method to perform structure and topological analysis of financial markets, where nodes represent assets and edges represent the relationship between them. This method was adopted in several studies [29–34]. Wang et al. [35] presented a literature review to predict links in social networks. The paper proposed an arrangement of the methods for link prediction in two high-level groups: similarity based and learning based. In addition to this arrangement, the authors described techniques for similarity-based link prediction, which used information from nodes and topology to calculate the similarity between pairs of nodes, and learning-based methods, whose features are derived from node information, network, network topology and non-topological information. Al Hasan et al. [36] studied link prediction as a supervised machine learning problem. They proposed using of three input sets for the machine learning model: proximity features, which described the proximity between two nodes; aggregated features, which aggregated attributes related to nodes; and topological features, related to the network topology. In addition, the authors made a comparison between some supervised machine learning algorithms and analyzed the most important features for link prediction. Lichtenwalter et al. [37] examined important factors for link prediction using a supervised approach. The authors presented a link prediction algorithm that uses supervised learning and proposed a set of features based on path-information, random walk and node neighborhood (called unsupervised methods). Comparative results showed significant improvement in the results of the two databases compared with baseline algorithms. Aouay et al. [38] studied link prediction as a supervised learning task, combining several features as input data for classification. To improve the accuracy, the authors applied a feature selection algorithm. Experiments were performed on two co-authored data sets and the results showed that Random Forest, k-NN and PCA produced the best performances. Fire et al. [39, 40] proposed a set of structural features at the node and link level to identify missed links using supervised machine learning algorithm. Zhu et al. [41] proposed a method to estimate the probability of a link using supervised machine learning. They proposed a method that combines information from the network structure and user-generated content as input to machine learning algorithms. They performed the comparison between three different machine learning algorithms and compared them with thirteen baseline methods. Tan et al. [42] investigated the importance of network topology in link prediction using information theory. In addition to the analysis, the authors proposed a general method for predicting links based on mutual information from the network topology, which represents the reduction in prediction uncertainty due to another variable. The proposed method presented better results in ten databases when compared to six baseline methods used in the literature. Malhotra et al. [43] propose three different link prediction algorithms based on different structural features of networks combined with information theory analyses. The proposed methods presented better and more robust performances in general cases. Bu et al. [44] analyzed the link prediction in temporal networks through a semi-supervised machine learning method. The method uses a sequence of adjacency matrices (time sequence) and the Cox Proportional Hazard Model (Cox PHM) to study the relative risk associated with each link to estimate the coefficients of covariates, which are defined as a set of neighborhoods based on proximity characteristics. Furthermore, the authors proposed a bidirectional selection mechanism based on game theory to predict the future topology of the network. Ma et al. [45] evaluated the performance of similarity-based methods for link prediction and showed that the performance of these methods is not always good in all cases, as each network has its structural characteristics. They analyzed different real networks and showed that these structural features are remarkably different (even in the same network). Thus, the authors proposed to apply several features and similarity indices as input to the proposed method, called adaptive fusion, which combines these features using a logistic function. The performance of this model is better than many similarity indices.

Considering stock networks, Yao et al. [46] analyzed the 180 most important stocks on the Shanghai Stock Exchange (SSE) through stock networks using the log-return of the closing price of these stocks. Networks were created with different thresholds. According to different networks under different thresholds, they found the actions with the greatest potential for influence based on local structural centrality. Finally, they analyzed the link prediction in stock networks using different similarity and path-based indices and showed that there are better similarity indices to predict the probability of node connections in different stock networks.

The link prediction problem is also applied to other research areas. Lu et al. [47] investigated the prediction of Drug–Target Interaction (DTI), which is the discovery of new uses for existing drugs through network-based prediction. In this work, the authors proposed a new method for DTI prediction that uses only network topology information. Wang et al. [48] proposed a method for predicting Drug-Protein Interactions (DPIs), important for drug repositioning, drug discovery and clinical medicine, by predicting bipartite links and networks. The method uses node similarity approaches to extract information from the network structure in order to predict hidden links. Lim et al.[49] presented an analysis of link prediction in criminal networks using Deep Reinforcement learning and a set of features based on node similarity. To do this, the authors developed a network model for link prediction by reconstructing a corrupted criminal network database.

3 Materials and methods

In this section, we describe the main steps of the proposed method to forecast market structure from financial network features using machine learning. Figure 1 presents the methodology.

Initially, we calculate the pairwise correlation matrix based on daily closing price series of assets. Given the correlation matrix, the market structure is modeled as a financial network by calculating the assets' distance matrix and applying a network filtering method. In this article, we evaluated three different network filtering methods to model financial market structure, described in Sect. 3.1. We then extract a set of network features, used as input attributes for the machine learning model, by calculating node- and link-level network features, as described in Sect. 3.2.1. Finally, we applied a machine learning model, described in Sect. 3.2, to forecast financial networks using network information itself as input.

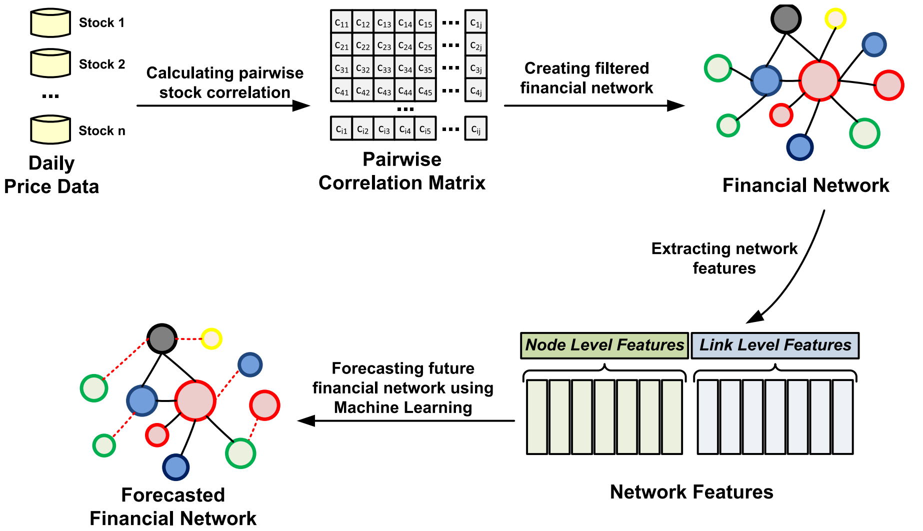

Stock 1
Stock 2
...
Stock n
Daily Price Data
Calculating pairwise stock correlation
Pairwise Correlation Matrix
Creating filtered financial network
Financial Network
Extracting network features
Node Level Features
Link Level Features
Network Features
Forecasting future financial network using Machine Learning
Forecasted Financial Network

Fig. 1 Main steps of the methodology used in this work. Based on daily asset closing prices of stocks constituents of a target stock market index, we calculate a pairwise correlation matrix and create a filtered financial network using three different network filtering algorithms. Given the financial network, we create a graph embedding by extracting network derived features at node and link levels. These features are used as input for a machine learning algorithm to forecast future financial networks

3.1 Dynamic financial networks

There are many methods in the literature to model financial market structure. Some of the most commonly used methods include correlation based networks and network filtering methods [7]. Network filtering methods allow prompt and temporal analysis of the market structure by exploring market data snapshots to model financial networks that represent the topology and the structure of the market. Using a rolling window approach, we can take snapshots in each time window of arbitrary length, allowing to explore temporal analysis of the market evolution [13], also called as dynamic or temporal networks. Some examples of the most common methods include Minimal Spanning Tree approach [4], the Planar Maximally Filtered Graph [50], the Directed Bubble Hierarchical Tree [14], asset graphs [51] and other approaches based on the threshold networks [52].

In this study, we investigate three different network filtering methods to estimate financial market structure: *(i)* Dynamic Asset Graph; *(ii)* Dynamic Threshold Networks; and *(iii)* Dynamic Minimal Spanning Tree. We explore these three methods due to their importance for financial analysis, considering that there is a vast literature [30, 51–56] that uses these methods to study different characteristics of the structure of financial networks.

These methods estimate an asset distance matrix through co-movement metrics of daily return prices. Let $P(t)$ be the closing price of an asset at day t. We consider assets' daily log-returns $R(t) = \log P(t) - \log P(t-1)$ that are calculated at time t. First, we calculate a distance matrix that measures the co-movement of daily log-returns [4], defined as

$$D_{i,j}(t) = \sqrt{2(1 - \rho_t(i, j))}, \tag{1}$$

where $\rho_t(i, j)$ is the Pearson's correlation coefficient between the time series of log-returns of assets i and j at time t, $\forall i, j \in V$, where V is the set of assets. The distance matrix is constructed by dividing the returns time-series $R(t)$ into rolling windows of size L trading days with δT trading days between two consecutive windows (time-step). The choice of window width L and window time-step δT is arbitrary, and it is a trade-off between having an analysis that is either too dynamic or too smooth [57]. The smaller the window width and the larger the window steps, the more dynamic the data are. We report results for $L \in \{126, 252, 504\}$ and $\delta T = 5$ trading days. A dynamic financial network is defined as a temporal network

$$W = \langle V, E_1, \ldots, E_T : E_t \subseteq V \times V, \forall t \in \{1, \ldots, T\}\rangle, \tag{2}$$

where vertices $i \in V$ correspond to assets of interest. For every pair $\langle i, j\rangle$ at time-window $t, \forall i, j \in V \mid i \neq j$, there is a corresponding edge $(i, j)_t \in E_t$ and every edge has a weight $w_{i,j}(t) = D_{i,j}(t)$. Considering the distance matrix $D_{i,j}(t)$ previously defined, we can apply a network filtering method in order to create dynamic networks. The three evaluated methods in this work are described in the next sections.

3.1.1 Dynamic Asset Graph (DAG)

A Dynamic Asset Graph [51] is a type of filtered financial network modeled by first ranking edges in ascending order of weights $w_1(t), w_2(t), ..., w_{N(N-1)/2}(t)$. The resulting graph is obtained by selecting the edges with the strongest connections. The number of edges are, of course, arbitrary. Here, we select edges with weights in the top quartile, i.e., $w_1(t), w_2(t), ..., w_{\lfloor N(N-1)/8 \rfloor}(t)$, as proposed in Souza et al. [17]. The main idea of this method is to identify the smallest distances in the stock market.

3.1.2 Dynamic Threshold Networks (DTN)

Considering the distance matrix $D(t)$ defined in Equation (1), we create a filtered adjacency matrix A to construct the financial network using the following rules [52, 56]:

$$A_{i,j}(t) = \begin{cases} 1, & |D_{i,j}(t)| \geq r_c \\ 0, & |D_{i,j}(t)| < r_c \end{cases} \tag{3}$$

where assets $i, j \in V$ and $\forall (i, j)_t \in E_t$. The critical value r_c converts the matrix D into an undirected network, whereby $A_{ij}(t) = 1$ and $A_{ij}(t) = 0$ represents the existence and absence of edges between i and j at time window t, respectively. We fixed the r_c value in 0.65 because for $r_c \leq 0.65$ the network characteristics are submerged in large fluctuations [56]. It is important to observe that the DTN method can produce disconnected graphs and the number of edges is dynamic. In general, the main goal of this method is to identify pairs of assets that are highly correlated and above the threshold r_c. This is different from DAG, where pairs with a correlation value lower than r_c can be added to the network.

3.1.3 Dynamic Minimal Spanning Tree (DMST)

We create a Dynamic Minimal Spanning Tree [4] based on the smallest asset distance in the previous defined matrix $D(t)$. We use the Kruskal's Algorithm to identify the Minimal Spanning Tree (MST) in the fully connected graph D at time t. The number of edges is fixed and calculated as $N - 1$, where N is the number of assets. This method provides the smallest distance to interconnect the market, producing the minimal market structure to connect all assets.

3.2 Machine learning-based approach

In this section, we describe the proposed machine learning based approach to forecast stock market structure for a given market index. In this study, we address market structure forecasting as a network link prediction problem. Given snapshots of financial networks up to time t, we want to accurately predict the edges that will be present in the network at a given future time t'. We choose three times $t_0 < t < t'$ and provide an algorithm that accesses $W[t_0, t] = \langle V, E_{t_0}, \dots, E_t \rangle$ to estimate the likelihood of edges to be present in $W[t']$, where $t' = t + h$ and $h = \{1, 2, \dots, 20\}$ trading weeks.

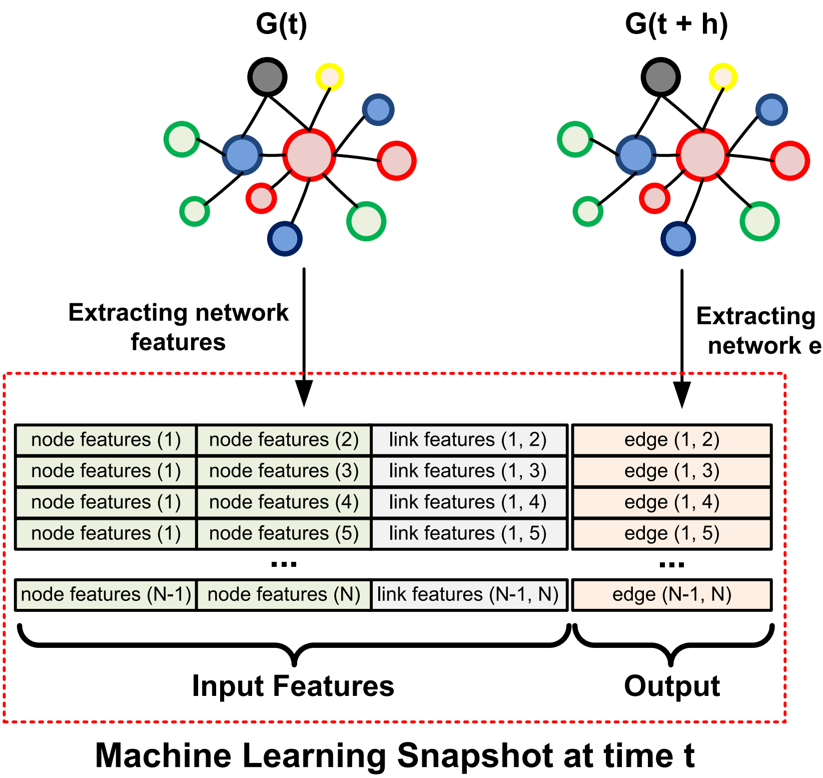
Similarity-based methods and classifier-based methods are two of the most common approaches for link prediction [58]. In similarity-based methods [59], the algorithm assigns a connection weight $score(x, y)$ to pairs of nodes $\langle x, y \rangle$, based on the input graph G, and then produces a ranked list in decreasing order of $score(x, y)$. These algorithms can be viewed as computing a measure of proximity or "similarity" between nodes x and y. Common Neighbors, Jaccard Coefficient, Preferential Attachment, Adamic Adar and Resource Allocation are among the most popular local indices (node-based). Katz, Leicht–Holme–Newman, Average Commute Time, Random Walk and Local Path represent global indices (path based). While the local indices are simple in computation, the global indices may provide more accurate predictions.

In classifier-based methods, the link prediction is defined as a binary classification problem. Here, a feature vector is extracted for each pair of nodes and a $1/0$ label should be assigned based on the existence/not existence of that link in the network. Any similarity-based method could form the required feature vector for a supervised learning method [36]. Afterward, any conventional supervised learning algorithm might be applied to train a supervised link predictor. In this article, we applied a classifier-based method to forecast the financial market structure. Our approach uses financial network features as input to a machine learning model in order to create a link prediction method, as presented in Fig. 2.

Figure 2 presents the process used to create the machine learning database. Assuming i and j as two arbitrary nodes ranging from 1 to N and t as the current time, an instance of the dataset used in the machine learning algorithm has the following predictive attributes: (a) i node-level features; (b) j node-level features; (c) (i, j) link-level features. As previously described, the target of the supervised machine learning model is to forecast the existence of links in a network $G(t + h)$, where $h = 1, 2, \dots, 20$ trading weeks. Figure 2 presents an illustration of how we build instances to the machine learning model, exemplified as the snapshot at time t.

We split the dataset between train and test sets taking into account the temporal sequence of the data. The train set includes data produced in the period from 1 March 2005 to 30 May 2007, and the test set has data from 30 May 2007 to 18 December 2019. Figure 3 presents an illustration explaining how we created the train and test sets. Machine learning models were trained and tested using a rolling window approach. Considering L as the size of the log-return time series, t as current time and $t - k < t < t + h$, we create the train set using network features from $G(t - k)$, where $k = 1, 2, \dots, 30$. The test set contains data from the current network $G(t)$, in which $G(t + h)$ is the target, where $h = 1, 2, \dots, 20$ trading weeks. After training the machine learning model and testing it, we move the rolling window forward taking into account the time-step $\delta T = 5$ trading days (1 trading week) between two consecutive executions (see Supplementary Material, Section S.1 for further details).

To assess the information rate that a machine learning model can extract from the features set, we applied the XGboost [60] algorithm. In this experiment, the algorithm induces a predictive model for stock market structure forecasting. XGboost is a fast, highly effective,


G(t)
G(t + h)
Extracting network features
Extracting target network edges
Input Features
Output
Machine Learning Snapshot at time t

node features (1)	node features (2)	link features (1, 2)	edge (1, 2)
node features (1)	node features (3)	link features (1, 3)	edge (1, 3)
node features (1)	node features (4)	link features (1, 4)	edge (1, 4)
node features (1)	node features (5)	link features (1, 5)	edge (1, 5)
	...		...
node features (N-1)	node features (N)	link features (N-1, N)	edge (N-1, N)

Fig. 2 Building the machine learning dataset. We calculate features for each node ranging from 1 to N, where N is the number of assets. We applied a pairwise concatenation of node and link features as input variables for the link prediction, while edges on the network at time $t + h$ are used as the target variable, where h is the number of trading weeks

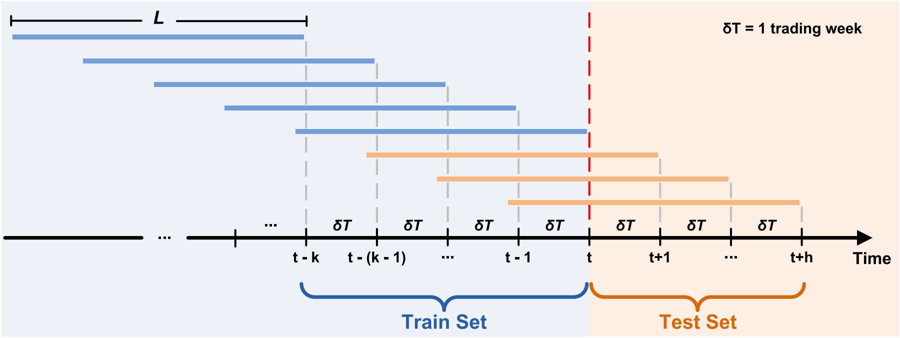


Fig. 3 Train and test sets used to induce the machine learning model. Machine learning models were trained and tested using a rolling window approach. Considering L as the size of the log-return time series and t as current time, we create the train set using data from $t - k$ to $t - 1$ and the test set using data from t. The target of the supervised learning is the network $G(t + h)$, where h is the number of trading weeks. After training and testing the machine learning model, the time-step δT is used to move the rolling window forward, in order to restart the process and re-train the machine learning model. The train set includes data from 1 March 2005 to 30 May 2007 and the test set has data from 30 May 2007 to 18 December 2019

interpretable and widely used machine learning model. Further information regarding the experimental setup is described in the Supplementary Material, Section S.2.

3.2.1 Network features

As previously mentioned, we proposed an approach for market structure forecasting based on supervised machine learning. In order to provide information to train this supervised method, we extracted a set of network features at node and link level. These features are used as input to the machine learning model. We summarized the network features as follows:

- **Node-Level Features** assess the position of a node within the overall structure of a given graph $G(V, E)$ [61]. Table 1 presents a set of node-level features related to node/stock $i \in V$ used as input to the machine learning model.
- **Link-Level Features** examine both the contents and patterns of relationships in a given graph $G(V, E)$ and measure the implications of these relationships [61]. Table 2 presents a set link-level features related to link $(i, j) \in E$ used as input to the machine learning model.

Researchers in finance, particularly in portfolio management, commonly use asset correlation in important use cases, such as risk management. Given the importance of this information in financial analyses, we also explore them as input feature for market structure forecasting. However, we are interested in analyzing how topological information helps to forecast the market structure itself. For this reason, we separated the feature set into two

Table 1 Node-level features: Features were calculated to node $i, \forall\, i \in V$ for a given graph $G(V, E)$

Name	Definition
Node Degree	$deg(i) = \|i\|$
Weighted Node Degree	$deg_w(i) = \sum_{j \in N_i} w_{<i,j>}$, where $w_{<i,j>}$ is the weight of the edge $e(i, j)$
Average Neighbor Degree	$avg(i) = \frac{\sum_{j \in N_i} \|j\|}{\|i\|}$
Propensity of i to Increase its Degree	$\gamma(i) = \frac{\|i\|}{deg_w(i)}$
Node Betweenness	$b(v) = \sum_{i,j \in V \setminus v} \frac{\sigma_{ij}(v)}{\sigma_{ij}}$, where $\sigma_{ij}(v)$ is the number of shortest paths between i and j passing through node v and σ_{ij} the total number of shortest paths from i to j
Node Closeness	$nc(i) = \frac{n-1}{\sum_{j \in V \setminus i} d(i,j)}$, where $d(i, j)$ represents the distance between i and j and n is the number of nodes in the graph
Node Eigenvector	$ne(i) = x_i \frac{1}{\lambda} \sum_{j=1}^{n} d_{ij} x_j$, where d_{ij} represents an entry of the adjacency matrix D (0 or 1), λ denotes the largest eigenvalue, x_i and x_j denotes the centrality of node i and j, respectively
Node Clustering Coefficient	$cc(i) = \frac{2\|e_{jk}\|}{\|i\| * (\|i\|-1)} : j, k \in N_i, e_{jk} \in E$

Consider N_i as the set of adjacent vertices (neighborhood) of node i. This set contains only non-pair-wise correlation features

Table 2 Link-level features: Features were calculated between nodes i and j, $\forall\ (i, j) \in E$ for a given graph $G(V, E)$

Name	Definition
Link Existence in $G(t)$ (*)	$E(i, j) = \begin{cases} 1 & \text{exists link} \\ 0 & \text{not exists link.} \end{cases}$
Correlation Value (*)	$C(i, j) = \rho_{ij}$, where $\rho_{i,j}$ is the Pearson's correlation coefficient between time series of log-returns of assets i and j
Common neighbors	$CN(i, j) = \lvert N_i \cap N_j \rvert$
Jaccard Coefficient	$JC(i, j) = \frac{\lvert N_i \cap N_j \rvert}{\lvert N_i \cup N_j \rvert}$
Adamic-Adar Coefficient	$AA(i, j) = \sum_{k \in N_i \cap N_j} \frac{1}{\log \lvert N_k \rvert}$, where N_k is the set of adjacent vertices of node k
Sorenson-Dice Coefficient	$SDC(i, j) = \frac{2 * \lvert N_i \cap N_j \rvert}{\lvert i \rvert + \lvert j \rvert}$
Edge Betweenness	$B(i, j) = \sum_{i, j \in V} \frac{\sigma_{ij}(e)}{\sigma_{ij}}$, where $\sigma_{ij}(e)$ is the number of shortest paths between i and j crossing the edge e and $\sigma_{i,j}$ is the total number of shortest paths from i to j
Same Community [62]	$\text{SC}(i, j) = \begin{cases} 1 & \text{if } i \text{ and } j \in \text{same community} \\ 0 & \text{if } i \text{ and } j \notin \text{same community.} \end{cases}$
Preferential Attachment	$PA(i, j) = \lvert i \rvert * \lvert j \rvert$, where $\lvert i \rvert$ and $\lvert j \rvert$ represent the node degree of vertex i and j

Pair-wise correlation features are marked with (*), while the remaining are features based on non pair-wise correlation. Consider N_i and N_j as the set of adjacent vertices of node i and j, respectively

distinct subsets. We labeled the two subsets according to their source of information: *(i)* pair-wise correlation features, which are attributes based on asset correlation and not derived from any other network information, and *(ii)* non-pair-wise correlation features, which are attributes derived from the network topology. While pair-wise correlation features are traditionally used in financial analysis, the importance of non-pair-wise correlation features to forecast market structure is a research question investigated in this work. Thus, we can compare their information gain in market structure forecasting. In Table 1, all features are non-pair-wise correlation attributes. In Table 2, the pair-wise correlation features are marked with (*).

3.2.2 Model evaluation

We calculate the *Area Under the ROC curve* (AUC) to evaluate the predictive performance of the link prediction methods. This metric is largely applied in binary classification and unbalanced problems and ranges from 0.5 to 1, where 0.5 represents a random naive algorithm and 1 represents the highest result. The AUC measure gives a summary metric for the algorithm's overall performance with different prediction set sizes, while a detailed look into the shape of the ROC curve reveals the predictive performance of the algorithm at each prediction set size [63].

To verify the performance of the proposed method, we compared it against seven baseline methods, organized into two distinct groups: (i) Naive Method, which represents the common approach used in financial market analysis, and (ii) Similarity-Based Method, which represents how several works in the literature solve the link prediction problem [59]. The baseline methods are described below:

1. *Naive Method*—assumes that the snapshot used for decision-making is static, through the use of a non-forward looking of the correlation matrix. The method in this group is described in the following:
   - Time Invariant (TI): This algorithm uses the link occurrence in graph $G(t)$ as the prediction of link occurrence in graph $G(t+h)$, assuming that market structure is time invariant. This assumption is traditionally used in risk management algorithms, which commonly rely on a backward-looking covariance matrix to estimate portfolio risk [17, 64].
2. *Similarity-Based Methods*—methods commonly used in literature for link prediction, as the problem addressed in this work [65]. The methods in this group are described in the following:
   - Common Neighbors [59] (CN): This is a simple and effective link prediction method based on common neighbors shared by two nodes. Pairs of nodes with high number of common neighbors tend to establish a link;
   - Preferential Attachment [66] (PA): This method defines that new links are formed between nodes with higher degrees rather than nodes with lower degrees;
   - Jaccard Coefficient [65] (JC): This method is based on similarity Jaccard's coefficient, taking into account the number of common neighbors shared by two nodes, but normalized by the total number of neighbors of both nodes;
   - Adamic-Adar [67] (AA): This method is also based on common neighbors shared by two nodes. Instead of using the raw number of common neighbors as CN, it is defined using the sum of the inverse of the logarithmic degree of each shared neighbor.
   - Local Path Index [68] (LP): Similar to CN, this method uses information from the next 2 and 3 nearest neighbors instead of using only information of the neighbors shared by two nodes.
   - Random Walk with Restart [69] (RW): Based on Random Walk, it is a special case of following the Markov chain, starting from a given node and randomly reaching a selected neighbor. The restart looks for the probability of a random walker starting from node x visits node y and comes back to the initial state node x [65].

3.3 Market data

In this study, we used data from six different stock market indices spread across the American, European and Asian markets. The stock indices were chosen to measure the performance of the proposed approach in different scenarios, given the diversity of the stock markets. Moreover, it is important to mention that they represent the stock market of the region or country where they are listed. We considered the following indices and associated countries/regions:

- **DAX30** (Germany): This is a stock market index that consists of the 30 largest and most liquid German companies trading on the Frankfurt Stock Exchange.

- **EUROSTOXX50** (Eurozone): This is a list of the 50 companies that are leaders in their respective sectors from eleven Eurozone countries, including Austria, Belgium, Finland, France, Germany, Ireland, Italy, Luxembourg, the Netherlands, Portugal and Spain.
- **FTSE100** (UK): This is an index listed in the London Stock Exchange. The Financial Times Stock Exchange Index (FTSE) is Britain's main asset indicator, managed by the independent organization and calculated based on the 100 largest companies in the UK.
- **HANGSENG50** (Hong Kong): This is an index listed in the Stock Exchange of Hong Kong. This stock market index has the 50 constituent companies with the highest market capitalization. It is the main indicator of the market performance in Hong Kong.
- **NASDAQ100** (USA): This is an index composed of the 100 non-financial largest companies listed in NASDAQ.
- **NIFTY50** (India): This is a stock market index listed in the National Stock Exchange of India based on the 50 largest Indian companies.

Each financial index has a daily price time series for each one of its constituent stocks. Price time series are constructed using daily closing prices collected from *Thomson Reuters*. The list of company constituents of each stock market index is not static and may change over time. In this article, we only consider companies that were part of the underlying indices across the entire period analyzed, as commonly used in other studies, when node prediction is out of scope [17, 70]. We consider prices ranging from 1 March 2005 to 18 December 2019.

4 Results and discussion

In this section, we present the experimental results for financial market structure forecasting. Initially, we present a set of descriptive analyses on evolution of financial networks and a brief discussion about the impact of different network filtering methods in the financial market structure. Afterward, we present a set of predictive analyses related to the machine learning approach and the benchmark methods. Finally, we present a discussion about the interpretability of the machine learning models.

4.1 Descriptive analysis

We present a set of descriptive analyses of temporal financial networks created across different market indices. The following sections present a set of analyzes that allow us to understand the characteristics of databases and temporal financial networks.

4.1.1 Financial network persistence

The first descriptive analysis describes financial network persistence, considering $L = 252$ trading days to create each graph (results regarding $L \in \{126, 504\}$ trading days can be found in Supplementary Material, Section S.3). This analysis allows us to measure how the financial networks change their structure over time. We estimate the network persistence by calculating pair-wise network similarity between $G(t)$ and $G(t')$ using the Jaccard Distance, defined as follows:

$$sim(G(t), G(t')) = \frac{|G(t) \cap G(t')|}{|G(t) \cup G(t)|}, \tag{4}$$

where t and t' range from 12 May 2006 to 18 December 2019.

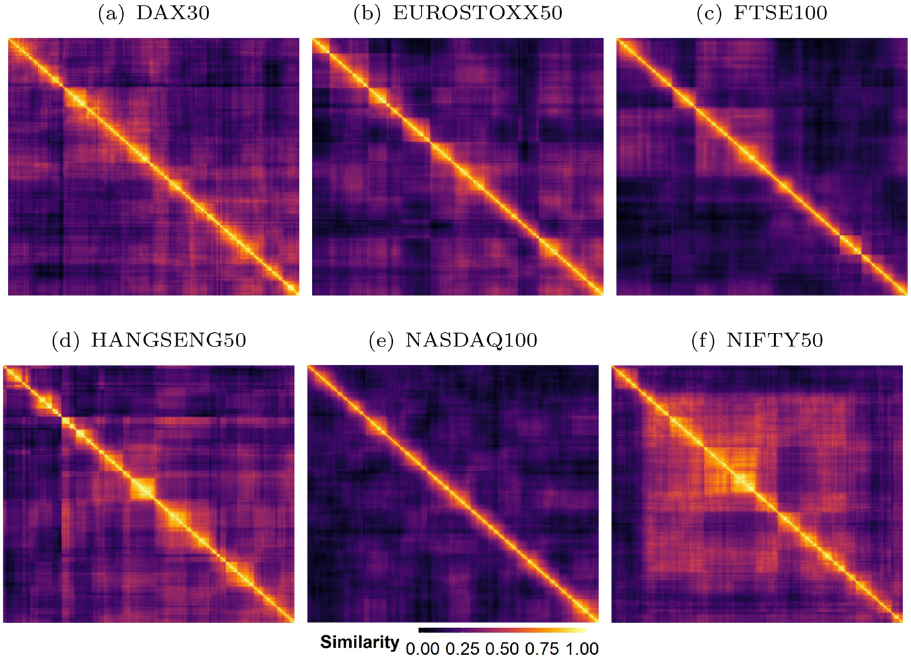


Fig. 4 DAG—Cross-similarity matrix for each market index. We calculate the pair-wise Jaccard Distance across all financial networks $G(t)$ and $G(t')$ ranging from 12 May 2006 to 18 December 2019, related to a given market index. For each market index figure, the first network on 12 May 2006 is represented in the top-left and the last network on 18 December 2019 in the bottom-right corner of each individual figure

Figures 4, 5 and 6 present the cross-similarity analysis for DAG, DTN and DMST of each stock market index, respectively. In the individual figure of each stock market index, the first network is represented in the top-left and the last network is represented in the bottom-right, where the first network is 12 May 2006 and the last network is 18 December 2019. In general, we can observe that the structure consistently changes over time, which emphasizes the importance of tools to forecast market structure.

DAG results in Fig. 4 show network structure changes considerably throughout the time in all stock market indices. Figure 5 presents results from the DTN network filtering method. We can observe the similarity among networks tends to be noisier than the previous DAG method. In some periods, the similarity among the networks is maximum, while at other times it reaches zero, as can be seen in NASDAQ100 and NIFTY50. The DTN network filtering method can produce disconnected or even empty graphs, which may cause these similarity oscillations. DMST results are shown in Fig. 6. This figure shows that there is low similarity for long-range comparisons among trees created by the DMST filtering method for all market indices, suggesting low stability as reported by other authors [71, 72].

After analyzing the persistence of financial networks, we present an analysis of the distance among all matrices to measure how similar is the evolution of the persistence between markets. Given the cross-similarity matrices of each market, we calculate the distance among all matrices to measure the market similarity in terms of network evolution. This analysis allows us to identify which markets have similar behavior considering the persistence of networks.

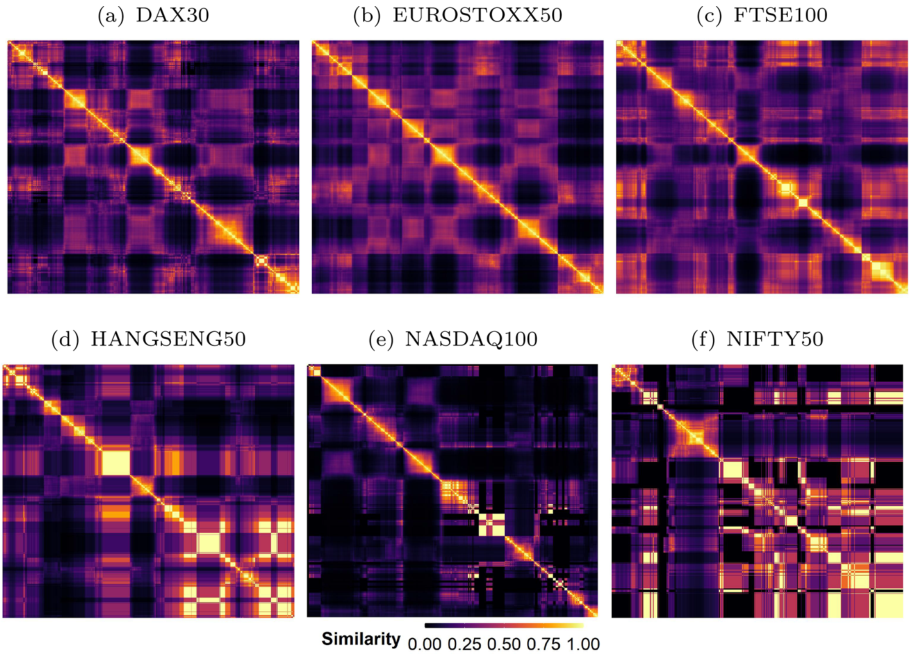


Fig. 5 DTN—Cross-similarity matrix for each market index. We calculate the pair-wise Jaccard Distance across all financial networks $G(t)$ and $G(t')$ ranging from 12 May 2006 to 18 December 2019, related to a given market index. For each market index figure, the first network on 12 May 2006 is represented in the top-left and the last network on 18 December 2019 in the bottom right of each individual figure

To do this, we use the cosine similarity, calculated using the following formula:

$$cosine\_sim(a, b) = \frac{\sqrt{\sum (a-b)^2}}{\sqrt{\sum a^2} * \sqrt{\sum b^2}}, \tag{5}$$

where a and b are two nonzero numeric vectors and represents the upper triangle of two distinct cross-similarity matrices. This metric ranges from 0 to 1 and it is defined as the angular distance from two vectors.

Table 3 presents the pairwise cosine similarity for DAG, DTN and DMST. As we have the commutativity property in cosine similarity, where $cosine\_sim(a, b)$ is equal to $cosine\_sim(b, a)$, we show the possible combinations among all market indices. It is possible to notice that all similarity analyses among all market indices are presented in Table 3(DAX30 *vs.* EUROSTOXX50, DAX30 *vs.* FTSE100 and so on). DAX30 and EUROSTOXX50 have the highest cosine similarity for DAG and DTN. For DMST, the highest value is between FTSE100 and EUROSTOXX50. This analysis demonstrates that the network persistence among markets from Europe are higher than markets from other regions of the world, given the three network filtering methods.

4.1.2 Financial network evolution

The second descriptive analysis is the similarity between the current financial network $G(t)$ and the future network $G(t+h)$, where h is the time lag, $\forall\ h \in \{1, 5, 10, 15, 20\}$ trading

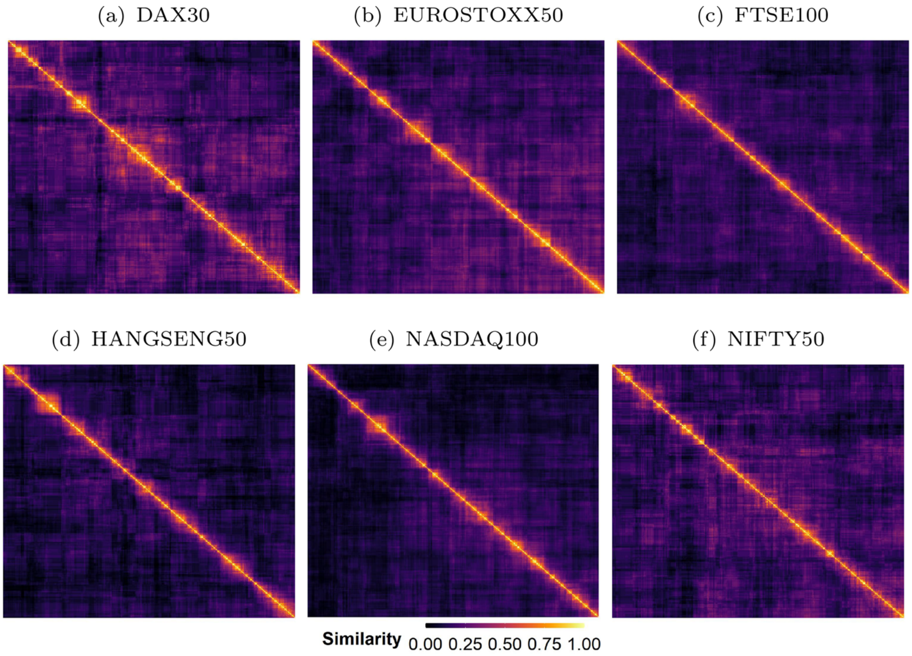


Fig. 6 DMST—Cross-similarity matrix for each market index. We calculate the pair-wise Jaccard Distance across all financial networks $G(t)$ and $G(t')$ ranging from 12 May 2006 to 18 December 2019, related to a given market index. For each market index figure, the first network on 12 May 2006 is represented in the top-left corner and the last network on 18 December 2019 in the bottom right of each individual figure

weeks. This analysis provides an accurate point of view concerning how the current network changes in the near future—if they do not change, we do not need to forecast them. We quantify the changes in the network structure using the Jaccard Distance between $G(t)$ and $G(t+h)$, considering $L = 252$ trading days to create each graph. Figure 7 presents the distribution of networks similarity related to the three network filtering methods DAG, DTN and DMST of each stock market index. Experimental results suggest a high similarity distribution among networks considering $h = 1$ step ahead to all network filtering methods. However, the similarity distribution decreases with h, mainly in the DMST method. Considering $h = 20$, DMST presents a mean similarity lower than 25% in all markets. In general, financial networks tend to have a certain margin of similarity for low h, but as h increases, they become more and more dissimilar, hence justifying the importance of forecasting future market structures, particularly in high-horizon forecasting scenarios. Analyzing the DTN method, NIFTY50 and HANGSENG50 present a different behavior for larger h, where the distribution of the similarity behaves differently from other markets, oscillating between the maximum value and almost zero for larger h, as shown in $h = 5$, $h = 10$ and $h = 15$. This amplitude can be explained by the analysis presented in Fig. 5, which shows that for some periods the similarity among networks is high, but it is also very low for other periods. The smallest similarity values are presented for the DMST method considering $L = 20$.

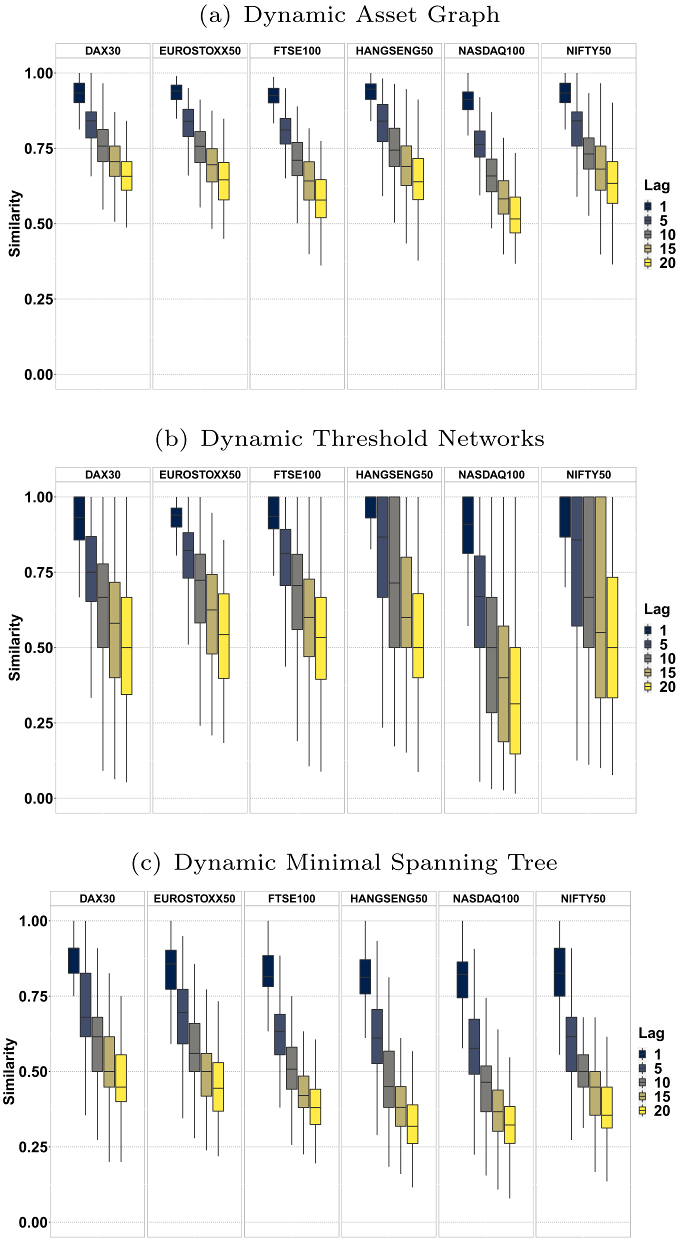
Table 3 Cosine distance from cross-similarity results

	EUROSTX	FTSE	HSENG	NASDAQ	NIFTY
DAG					
DAX	**0.9532**	0.9435	0.9472	0.9341	0.9257
EUROSTX		0.9228	0.9403	0.9420	0.9070
FTSE			0.9150	0.9358	0.8978
HSENG				0.9297	0.9302
NASDAQ					0.9137
DTN					
DAX	**0.9338**	0.8367	0.7573	0.6209	0.5795
EUROSTX		0.8755	0.7873	0.6143	0.6000
FTSE			0.8331	0.5479	0.5503
HSENG				0.5892	0.5531
NASDAQ					0.4269
DMST					
DAX	0.9486	0.9354	0.8967	0.9011	0.9200
EUROSTX		**0.9500**	0.9058	0.9294	0.9312
FTSE			0.9253	0.9400	0.9338
HSENG				0.9169	0.9080
NASDAQ					0.9160

The bold values are the highest cosine similarity measures for each financial network DAG, DTN and DMST We calculate the cosine similarity from cross-similarity matrices. We use the upper triangle of each matrix as the input vector. European markets have the highest similarity

4.1.3 Financial network structure

The third descriptive analysis represents the financial network structure and is presented in Fig. 8. We present the Cumulative Distribution Function (CDF) of the node degree across networks of each index using the DAG, DTN and DMST network filtering methods. This analysis provides information concerning the node degree according to three main aspects: *(i)* the impact of time series size L; *(ii)* network filtering method and *(iii)* size of the market index, considering the number of constituents. We calculated the node degree distribution across all financial networks ranging from 3 March 2007 to 18 December 2019. Results using $L \in \{126, 252, 504\}$ trading days as rolling window size are presented. We observe in Fig. 8 that market indices with the smallest number of constituents present a similar behavior in terms of node degree when we use the DAG network filtering method. Besides, DAG nodes are prone to have a higher occurrence of node with no connections. The DTN method also presents high probability of nodes without edges, mainly on NIFTY50, NASDAQ100 and HANGSENG50. EUROSTOXX50 presents a distinct shape compared with the other market indices in DTN with the smallest number of nodes without a connection—more than 75% of nodes has a degree greater than 1 edge. On the other hand, for all market indices, at least 50% of the nodes have 4 or more connections in DAG. Considering the number of stocks in each market index, we can also conclude that there are no nodes connecting to all other vertices in any network filtering method because the largest degree distribution of each market index. Results also suggest the degree distribution of the market indices is similar for


(a) Dynamic Asset Graph

(b) Dynamic Threshold Networks

(c) Dynamic Minimal Spanning Tree

Fig. 7 Networks Similarity versus Time Lag. Figure shows the distribution of networks persistence considering $h = \{1, 5, 10, 15, 20\}$ trading weeks ahead related to the three network filtering methods: DAG, DTN and DMST. Network similarity is quantified using the Jaccard Distance between graphs $G(t)$ and $G(t+h)$

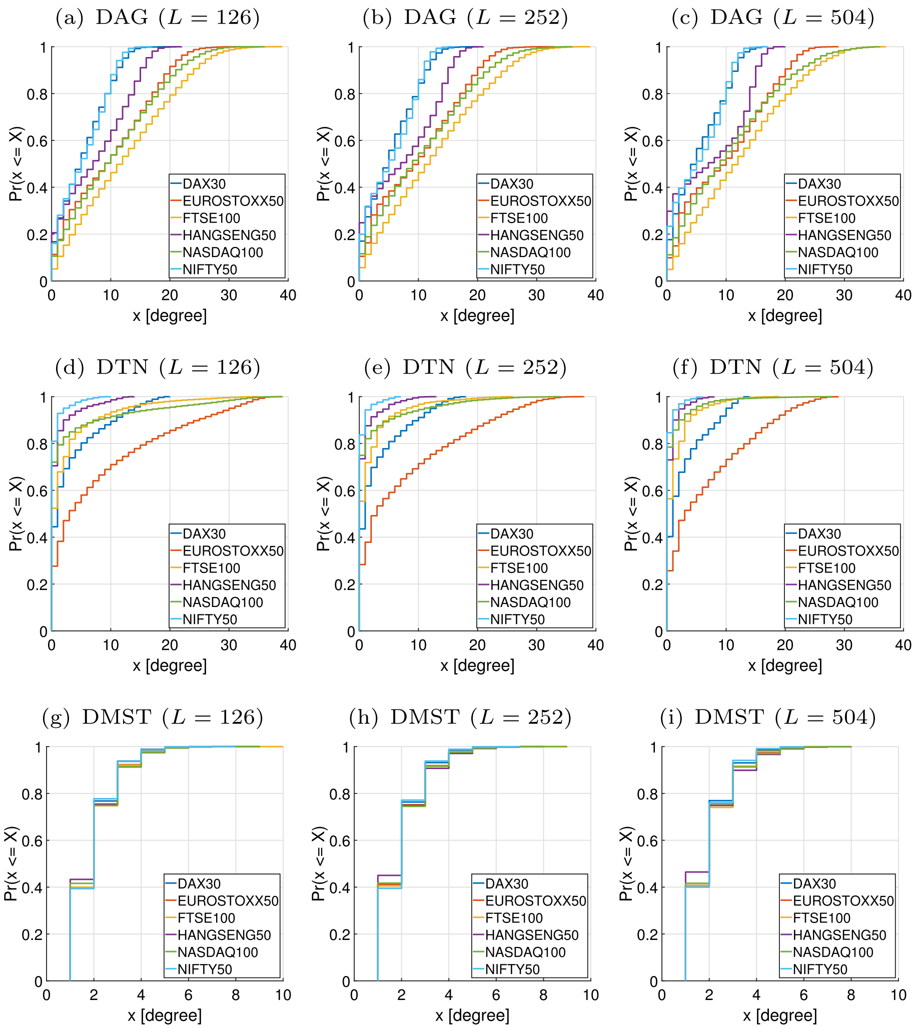


Fig. 8 CDF of node degree across networks using DAG, DTN and DMST network filtering methods. We calculate the cumulative distribution function of node degree across all stock networks using the size of rolling window $L = 126$, 252 and 504 trading days. The period of the experiments ranges from 3 March 2007 to 18 December 2019

$L = 126$, 252 and 504 trading days in all network filtering methods, indicating that the size of L does not affect the degree distribution of stock networks of each market index.

4.2 Predictive analysis

In this section, we present a set of experimental results related to market structure forecasting using machine learning. First, we investigate the predictive performance of the proposed method in different scenarios, comparing it against the benchmark methods. Then, we present a qualitative analysis concerning the model interpretability and its implications.

4.2.1 Performance results

We used a machine learning approach to forecast the financial network $G(t+h)$, where h is the number of weeks ahead, $h = 1, 2, \ldots, 20$ trading weeks. We discuss and report results using the size of rolling windows $L = 252$ trading days to construct the financial networks. Results regarding $L \in \{126, 504\}$ trading days can be found in the Supplementary Material, Section S.4. Figures 9, 10 and 11 show the AUC measure of the proposed machine learning method compared to baseline algorithms for DAG, DTN and DMST network filtering methods. For each time step ahead h, we calculated the average AUC of each method and its respective standard error over the test period, ranging from 5 May 2007 to 18 December 2019.

Denoted as "ML", the machine learning method outperforms the baseline methods in all market indices and all network filtering methods. In general, predictive performance decreases as the time lag h increases. Despite its simplicity, TI is quite effective and presents good performance across market indices and network filtering methods, similar to RW algorithm. Figure 9 presents results for the DAG network filtering method, suggesting that market indices with a small number of constituents have a higher AUC than markets with a large number of constituents. Results also suggest that the RW algorithm produces a edge ranking quite similar to TI. The JC method presents the worst predictive performance in all market indices, except for FTSE100 in which PA presents lower AUC values for the DAG network filtering method.

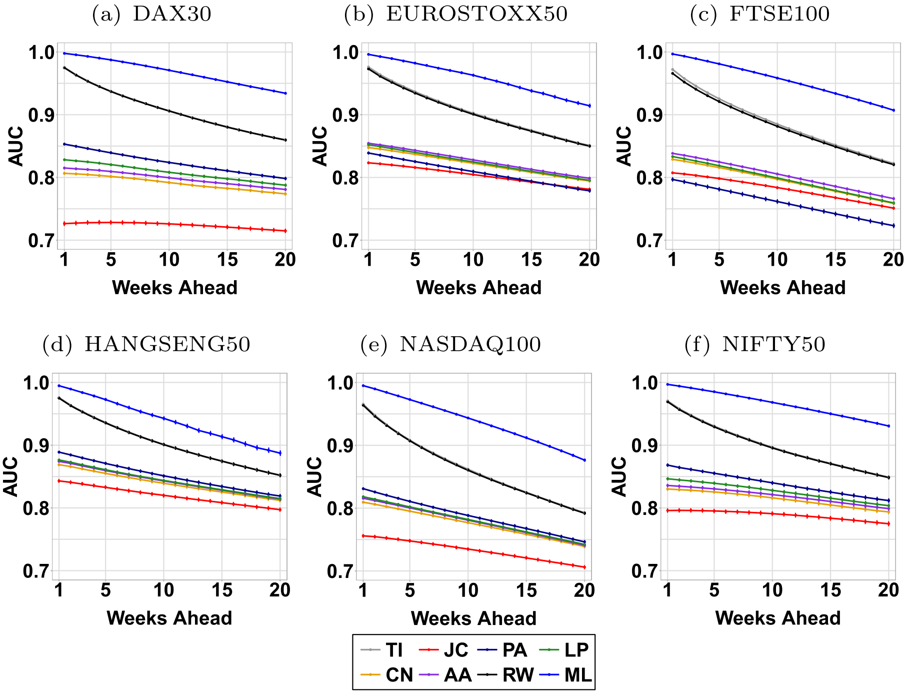


Fig. 9 DAG—Predictive performance comparison of all methods. This figure shows the AUC measure of the machine learning method compared to the baseline methods. For each time step, we calculate the AUC average of each method and its respective standard error over the entire test period. The machine learning method outperforms the baseline methods in all market indices

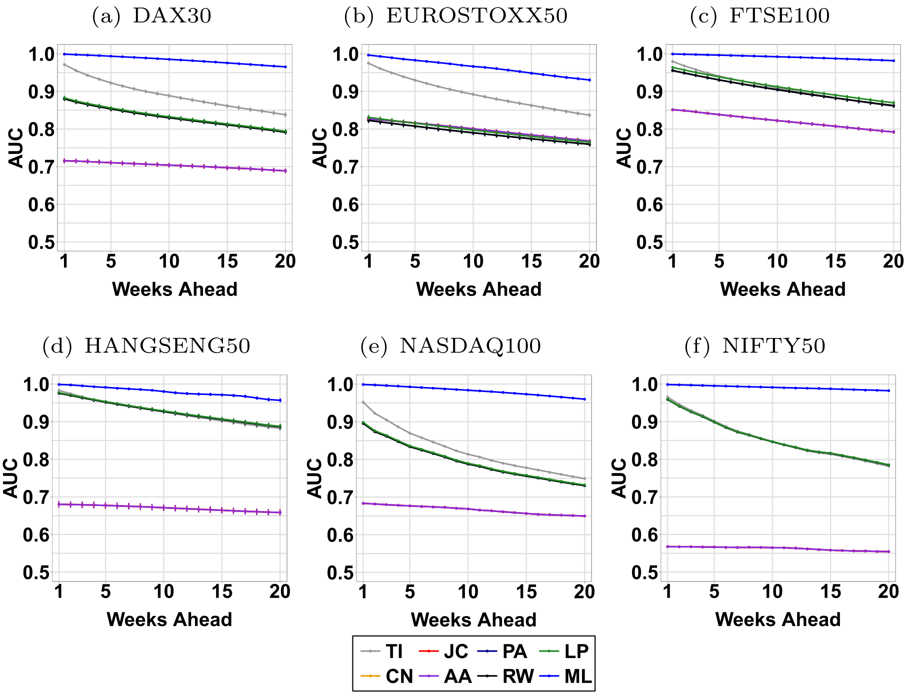


Fig. 10 DTN—Predictive performance comparison of all methods. This figure shows the AUC measure of the machine learning method compared against the baseline methods. For each time step, we calculate the AUC average of each method and its respective standard error over the entire test period. The machine learning method outperforms the baseline methods in all market indices

Figure 10 presents results for the DTN network filtering method. ML results are superior in all markets and suggest the proposed method can accurately identify links with high correlation due the main purpose of DTN method. We can observe that baseline algorithms have worst results for HANGSENG50, NASDAQ100 and NIFTY50 indices. As presented in Fig. 8, these market indices have expressive number of nodes without connections. TI algorithm outperforms baseline algorithms in DAX30, EUROSTOXX50 and NASDAQ100. Figure 11 presents results related to the DMST network filtering method. Baseline methods have the worst results among the three filtering methods, except for the TI and RW algorithms. ML outperforms the benchmark methods in all markets.

Figure 12 presents the proposed method AUC performance for h trading weeks ahead ($1 \leq h \leq 20$) using the DAG, DTN and DMST network filtering methods. The AUC measure decreases as the time lag h increases. We also compared our results against the benchmark time invariant method TI, where the network $G(t)$ is used as the forecast $G(t+h)$. We choose TI to compare our method due to its superior performance over all benchmark methods presented in the previous analysis. Moreover, we selected the TI method because it is derived from information from the pair-wise correlation, as described in Table 2. The AUC* improvement is calculated as follows:

$$\text{AUC}^* = (\text{AUC}_m - 0.5)/(\text{AUC}_b - 0.5) - 1, \tag{6}$$

where AUC_m is the machine learning AUC and AUC_b is the benchmark's AUC.

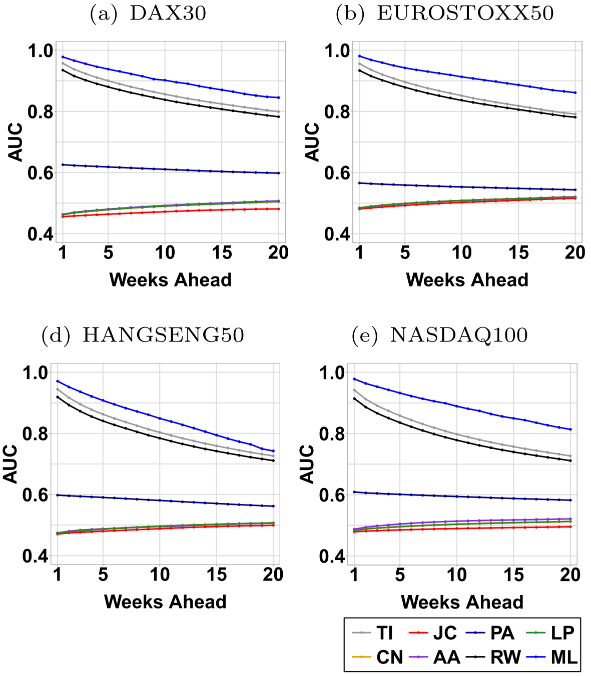


Fig. 11 DMST—Predictive performance comparison of all methods. This figure shows the AUC measure of the machine learning method compared against the baseline methods. For each time step, we calculate the AUC average of each method and its respective standard error over the entire test period. The machine learning method outperforms the baseline methods in all market indices

Figures 12b, d and f present AUC* improvement results and their standard errors for DAG, DTN and DMST network filtering methods.

The proposed method presents similar AUC results for all network filtering methods. Results using DAG shown in Fig. 12a suggest that networks with fewer constituents have better AUC results. Figure 12b shows that the highest AUC* improvement is from NASDAQ100, reaching almost 30% for $h = 20$ weeks ahead. On the other hand, for the DTN method shown in Fig. 12c, the best results are FTSE100 and NIFTY50, in which EUROSTOXX50 is the most distinct result. The biggest AUC* improvement related to DTN shown in Fig. 12d is over NASDAQ100 and NIFTY50, reaching almost 40%. Results shown in Fig. 12e are related to the DMST network filtering method and have a similar decay of AUC for all markets, where DAX30 is the best result. Interestingly, the AUC* improvement shown in Fig. 12e presents similar curves to NIFTY50 and HANGSENG50 markets. Results show that AUC* improvement for NIFTY50 and HANGSENG50 increases until approximately $h = 9$, achieving almost 12% on NIFTY50. After this max value, the AUC* improvement decreases as h increases. NASDAQ100 presents the best AUC* improvement, reaching almost 19% for $h = 15$ trading weeks ahead.

4.2.2 Model interpretability

In finance, particularly in portfolio management, the investment risk is calculated using the correlation among portfolio assets. This is the main information used to estimate risk and,

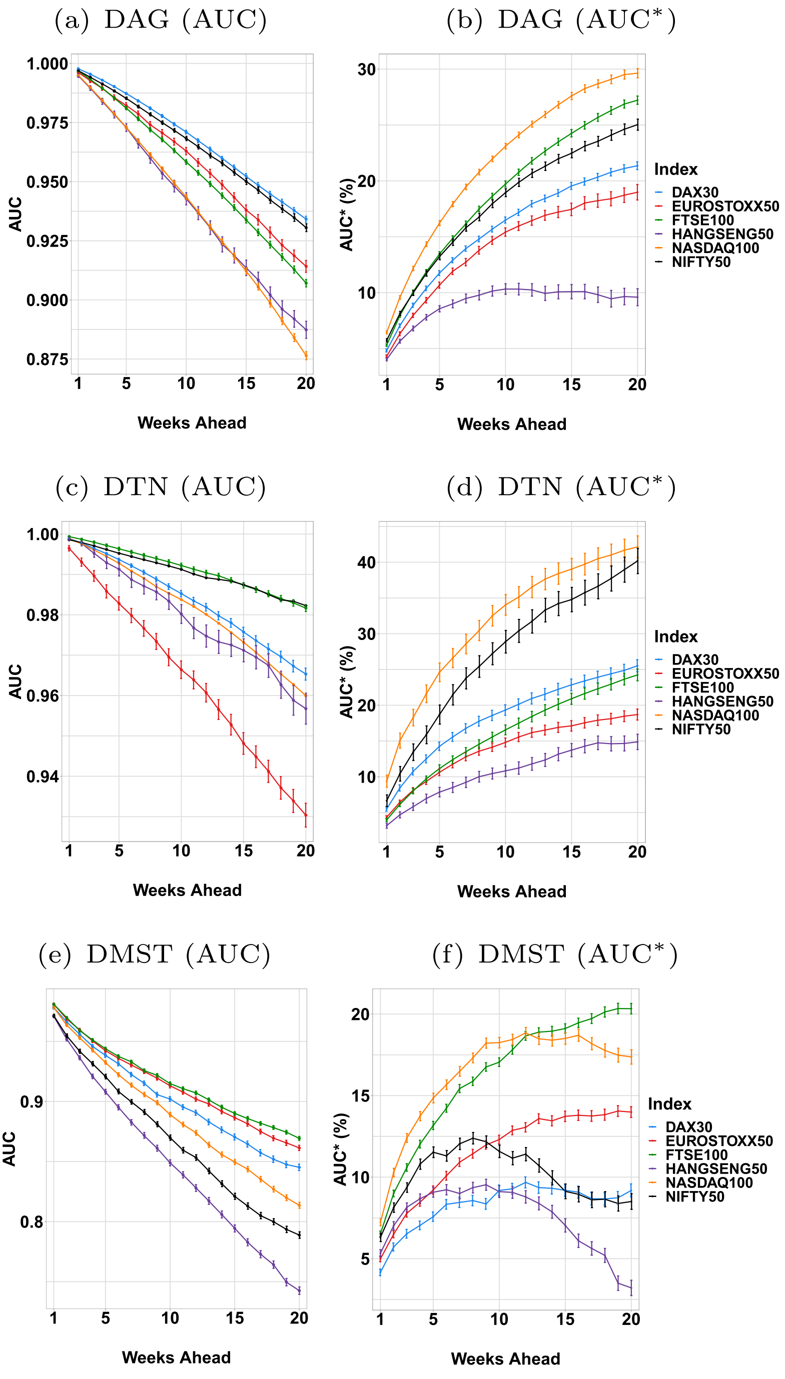


Fig. 12 Machine learning AUC and AUC* for DAG, DTN and DMST network filtering methods. Panels (**a**), (**c**) and (**e**) present the machine learning AUC measure and its standard error for h trading weeks ahead ($1 \leq h \leq 20$). Panels (**b**), (**d**) and (**f**) present the AUC improvement over the benchmark time-invariant method and its standard error. Results for $L = 252$

given its importance in financial analyses, we also explore them as an input feature for market structure forecasting. However, we want to measure how the topology of the network helps forecast the future network itself. In other words, we are interested in evaluating the importance of non-pair-wise correlation features for the forecasting market structure. As described in Sect. 3.2.1, we separated the feature set into two subsets: pair-wise correlation features and non-pair-wise correlation features. After constructing the boosted trees in the XGBoost model, we can estimate the importance of each individual attribute. The importance of an attribute is related to the number of times that it is used to create relevant split decisions, i.e., split points that improve the performance metrics [73]. For each market index, we calculate the average and standard error of aggregate importance of pair-wise correlation

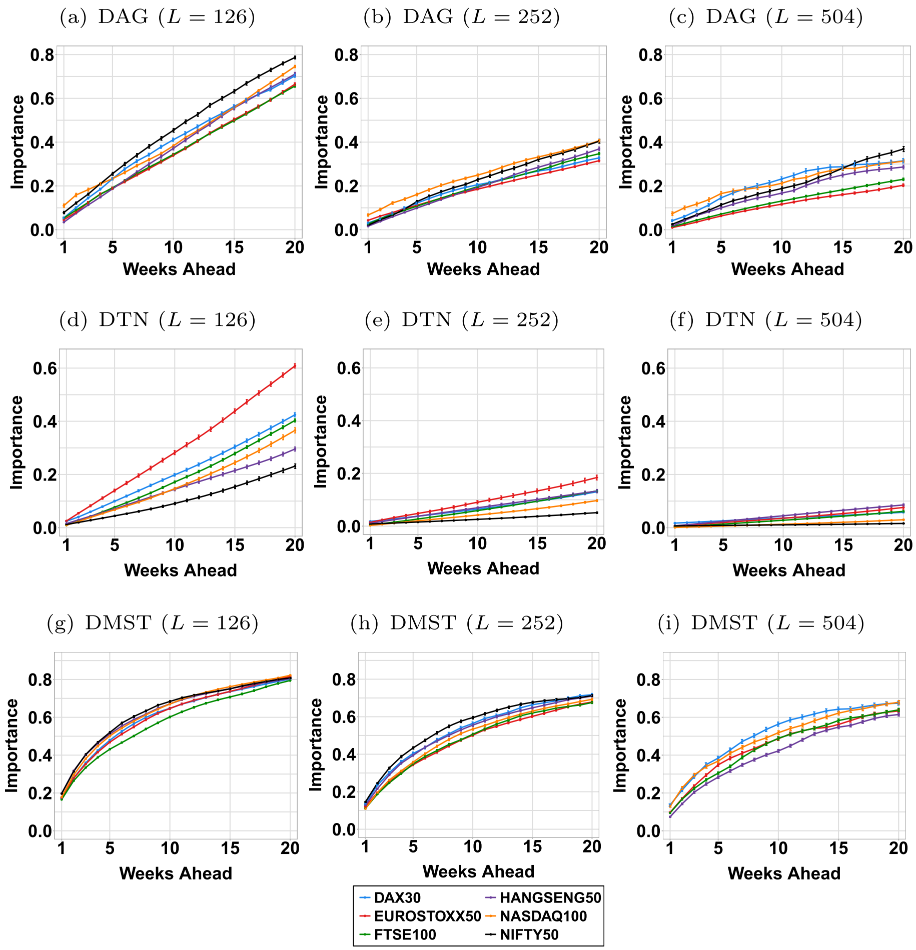


Fig. 13 Importance of non-pair-wise correlation features for DAG, DTN and DMST. Figure shows the aggregate importance for non-pair-wise correlation features using the size of rolling window $L = \{126, 252, 504\}$ trading days and DAG, DTN and DMST network filtering methods. Results show the importance of these features increases with the time step h. The importance of non-pair-wise correlation features for $L = 126$ trading days is higher than $L = 252$ and $L = 504$ for all network filtering methods. The growth of the importance of this subset is consistent across all markets. An interesting result is that the importance of non-pair-wise correlation features changes according to the network filtering method

and non-pair-wise correlation features. Figure 13 presents results related to the importance of non-pair-wise correlation features, considering the network filtering methods DAG, DTN and DMST and $L \in \{126, 252, 504\}$ trading days as the rolling window size. It is important to note that the importance of the two feature subsets adds up to 1.

Results presented in Fig. 13 show that non-pair-wise correlation features help forecast the future market using different network filtering methods. We observe that the importance of non-pair-wise correlation features increases with h. Moreover, the importance of this subset of features changes according to the network filtering method. Their importance can be observed mainly for smaller L, such as $L = 126$, shown in Fig. 13a, d and g, where their importance for $h = 20$ reaches almost 80% for NIFTY50 using the DAG method, 60% for EUROSTOXX50 using DTN and almost 90% for all markets using DMST. For the DMST method, shown in Fig. 13g, h and i, the importance of non-pair-wise correlation features has a similar shape to $L = 126$, 252 and 504 rolling window size. DAG results are shown in Fig. 13a–c. For short h values, non-pair-wise correlation attributes do not add much information when compared to pair-wise correlation features. However, the importance of these features rapidly increases with the time step h, suggesting that these attributes can be more useful than pair-wise correlation attributes for long-horizon forecasting exercises, particularly for short rolling window sizes. For $L = 252$ and $L = 504$, non-pair-wise correlation features have less importance in forecasting networks modeled using DAG and DTN network filtering methods. Considering DMST results, the importance of non-pair-wise features rapidly increases, even for short h values. This behavior is different from DAG and DTN. A possible explanation for this is the low persistence of trees, as shown in Fig. 6. Thus, network features are able to add more information to the ML model when compared to pair-wise correlation features.

5 Conclusion

In this article, we investigated stock market structure forecasting of multiple financial markets using financial networks modeled using stock returns of major market indices constituents. The stock market structure was modeled as networks, where nodes represent assets and edges represent the relationship among them. Three correlation-based filtering methods were used to create stock networks: Dynamic Asset Graphs (DAG), Dynamic Threshold Networks (DTN) and Dynamic Minimal Spanning Tree (DMST). We formulated market structure forecasting as a network link prediction problem, where we aim to accurately predict the edges that will be present in future networks. We proposed and experimentally assessed a machine learning model based on node- and link-based financial network features to forecast future market structure.

We used data from company constituents of six different stock market indices from the USA, the UK, India, Europe, Germany and Hong Kong markets, ranging from 1 March 2005 to 18 December 2019. To assess the predictive performance of the model, we compared it to seven link prediction benchmark algorithms. Experimental results showed the proposed model was able to forecast the market structure with a performance superior to all benchmark methods and for all market indices, regardless the network filter method. We also measured the improvement against the Time-Invariant (TI) algorithm, which assumes that the network does not change over time. Experimental results showed a greater improvement over the TI in networks created using the DTN filtering method, reaching almost 40% improvement for NASDAQ100. Our experimental results also suggested that topological network information is useful in forecasting stock market structure compared to pair-wise correlation measures, particularly for long-horizon predictions.

As work limitations, we should emphasize that we only used assets that stayed in the market index throughout the whole period, which limits the insertion and removal of nodes in the networks. In addition, for networks with large number of nodes, the execution time increased significantly, both for generating derived features and for training ML models.

Our results can be useful in the study of stock market dynamics and to improve portfolio selection and risk management on a forward-looking basis and market structure estimation. As future work, we plan to use the predicted stock market structure as input in portfolio and risk management tools to evaluate its usefulness in risk management scenarios. Future work also includes market structure forecasting using order book data for high-frequency trading analysis and the study of different asset classes beyond equities.

Acknowledgements D.C. and A.C.P.L.F.C would like thank to CAPES, IFSULDEMINAS, Intel and CNPq (Grant 202006/2018-2) for their support.

Author Contributions D.C. and T.T.P.S. developed the proposed model. D.C. and T.T.P.S. conceived and designed the experiments. D.C. and T.T.P.S. prepared figures and tables, implemented and carried out the experiments. All authors analyzed the results and wrote the manuscript. All authors reviewed the article.

Declarations

Conflict of interest The authors declare no competing financial interests.

References

1. Livan G, Inoue J-I, Scalas E (2012) On the non-stationarity of financial time series: impact on optimal portfolio selection. J Stat Mech Theory Exp 2012(07):07025
2. Morales R, Matteo TD, Aste T (2013) Non-stationary multifractality in stock returns. Phys. A 392(24):6470–6483. https://doi.org/10.1016/j.physa.2013.08.037
3. Cont R (2005) Long range dependence in financial markets. In: Lévy-Véhel J, Lutton E (eds) Long range dependence in financial markets. Springer, London, pp 159–179
4. Mantegna RN (1999) Hierarchical structure in financial markets. Eur Phys J B Condens Matter Complex Syst 11(1):193–197. https://doi.org/10.1007/s100510050929
5. Tumminello M, Aste T, Di Matteo T, Mantegna RN (2005) A tool for filtering information in complex systems. Proc Natl Acad Sci 102(30):10421–10426. https://doi.org/10.1073/pnas.0500298102
6. Iori G, Mantegna RN (2018) Chapter 11 empirical analyses of networks in finance. In: Hommes C, LeBaron B (eds) Handbook of computational economics, vol 4. Elsevier, Amsterdam, pp 637–685
7. Marti G, Nielsen F, Bińkowski M, Donnat P (2021) A review of two decades of correlations, hierarchies, networks and clustering in financial markets. In: Nielsen F (ed) Progress in information geometry. Signals and Communication Technology. Springer, Cham, pp 245–274
8. Morales R, Di Matteo T, Gramatica R, Aste T (2012) Dynamical generalized hurst exponent as a tool to monitor unstable periods in financial time series. Phys A 391(11):3180–3189
9. Song W-M, Aste T, Di Matteo T (2008) Analysis on filtered correlation graph for information extraction. Stat Mech Mol Biophys 88
10. Pozzi F, Di Matteo T, Aste T (2013) Spread of risk across financial markets: better to invest in the peripheries. Sci Rep 3:1665
11. Hüttner A, Mai J-F, Mineo S (2018) Portfolio selection based on graphs: Does it align with Markowitz-optimal portfolios? Depend Model 6(1):63–87
12. Tumminello M, Lillo F, Mantegna RN (2010) Correlation, hierarchies, and networks in financial markets. J Econ Behav Organ 75(1):40–58. https://doi.org/10.1016/j.jebo.2010.01.004
13. Musmeci N, Aste T, di Matteo T (2014) Clustering and hierarchy of financial markets data: advantages of the dbht. CoRR arxiv: 1406.0496v2

14. Song W-M, Di Matteo T, Aste T (2012) Hierarchical information clustering by means of topologically embedded graphs. PLoS ONE 7(3):31929
15. Musmeci N, Nicosia V, Aste T, Di Matteo T, Latora V (2017) The multiplex dependency structure of financial markets. Complexity. https://doi.org/10.1155/2017/9586064
16. Barfuss W, Massara GP, Di Matteo T, Aste T (2016) Parsimonious modeling with information filtering networks. Phys Rev E 94:062306. https://doi.org/10.1103/PhysRevE.94.062306
17. Souza TTP, Aste T (2019) Predicting future stock market structure by combining social and financial network information. Phys A 535:122343. https://doi.org/10.1016/j.physa.2019.122343
18. Spelta A (2017) Financial market predictability with tensor decomposition and links forecast. Appl Netw Sci 2(1):7
19. Musmeci N, Aste T, Di Matteo T (2016) Interplay between past market correlation structure changes and future volatility outbursts. Sci Rep 6:36320
20. Park JH, Chang W, Song JW (2020) Link prediction in the granger causality network of the global currency market. Phys A 553:124668
21. Castilho D, Gama J, Mundim L, Carvalho A (2019) Improving portfolio optimization using weighted link prediction in dynamic stock networks. In: International conference on computational science (ICCS)
22. Trippi RR, Turban E (1992) Neural networks in finance and investing: using artificial intelligence to improve real world performance. McGraw-Hill, Inc., Columbus
23. Long W, Lu Z, Cui L (2019) Deep learning-based feature engineering for stock price movement prediction. Knowl-Based Syst 164:163–173
24. Liu Y (2019) Novel volatility forecasting using deep learning-long short term memory recurrent neural networks. Expert Syst Appl 132:99–109
25. Pagolu VS, Reddy KN, Panda G, Majhi B (2016) Sentiment analysis of twitter data for predicting stock market movements. In: 2016 International conference on signal processing, communication, power and embedded system (SCOPES). IEEE, pp 1345–1350
26. Potvin J-Y, Soriano P, Vallée M (2004) Generating trading rules on the stock markets with genetic programming. Comput Oper Res 31(7):1033–1047
27. Martínez V, Berzal F, Cubero J-C (2017) A survey of link prediction in complex networks. ACM Comput Surv (CSUR) 49(4):69
28. Grover A, Leskovec J (2016) node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 855–864
29. Gopikrishnan P, Plerou V, Liu Y, Amaral LN, Gabaix X, Stanley HE (2000) Scaling and correlation in financial time series. Phys A 287(3–4):362–373
30. Onnela J-P, Chakraborti A, Kaski K, Kertesz J, Kanto A (2003) Dynamics of market correlations: taxonomy and portfolio analysis. Phys Rev E 68(5):056110
31. Lee GS, Djauhari MA (2012) An overall centrality measure: the case of us stock market. Int J Electr Comput Sci 12(6):99–103
32. Bonanno G, Caldarelli G, Lillo F, Mantegna RN (2003) Topology of correlation-based minimal spanning trees in real and model markets. Phys Rev E 68(4):046130
33. Bonanno G, Caldarelli G, Lillo F, Micciche S, Vandewalle N, Mantegna RN (2004) Networks of equities in financial markets. Eur Phys J B-Condens Matter Complex Syst 38(2):363–371
34. Eom C, Oh G, Jung W-S, Jeong H, Kim S (2009) Topological properties of stock networks based on minimal spanning tree and random matrix theory in financial time series. Phys A Stat Mech Appl 388(6):900–906
35. Wang P, Xu B, Wu Y, Zhou X (2015) Link prediction in social networks: the state-of-the-art. SCIENCE CHINA Inf Sci 58(1):1–38
36. Al Hasan M, Chaoji V, Salem S, Zaki M (2006) Link prediction using supervised learning. In: SDM06: Workshop on Link Analysis, Counter-terrorism and Security
37. Lichtenwalter RN, Lussier JT, Chawla NV (2010) New perspectives and methods in link prediction. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining, pp 243–252
38. Aouay S, Jamoussi S, Gargouri F (2014) Feature based link prediction. In: 2014 IEEE/ACS 11th international conference on computer systems and applications (AICCSA). IEEE, pp 523–527
39. Fire M, Tenenboim L, Lesser O, Puzis R, Rokach L, Elovici Y (2011) Link prediction in social networks using computationally efficient topological features. In: 2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing. IEEE, pp 73–80
40. Fire M, Tenenboim-Chekina L, Puzis R, Lesser O, Rokach L, Elovici Y (2014) Computationally efficient link prediction in a variety of social networks. ACM Trans Intell Syst Technol (TIST) 5(1):1–25

41. Zhu Y, Huang D, Xu W, Zhang B (2020) Link prediction combining network structure and topic distribution in large-scale directed network. J Organ Comput Electron Commer 30(2):169–185
42. Tan F, Xia Y, Zhu B (2014) Link prediction in complex networks: a mutual information perspective. PLoS ONE 9(9):107056
43. Malhotra D, Goyal R (2020) Link prediction in complex networks using information-theoretic measures. J Complex Netw 8(4):035
44. Bu Z, Wang Y, Li H-J, Jiang J, Wu Z, Cao J (2019) Link prediction in temporal networks: integrating survival analysis and game theory. Inf Sci 498:41–61
45. Ma C, Bao Z-K, Zhang H-F (2017) Improving link prediction in complex networks by adaptively exploiting multiple structural features of networks. Phys Lett A 381(39):3369–3376
46. Yao H, Lu Y (2017) Analyzing the potential influence of shanghai stock market based on link prediction method. J Syst Sci Inf 5(5):446–461
47. Lu Y, Guo Y, Korhonen A (2017) Link prediction in drug-target interactions network using similarity indices. BMC Bioinform 18(1):1–9
48. Wang W, Lv H, Zhao Y, Liu D, Wang Y, Zhang Y (2020) DLS: a link prediction method based on network local structure for predicting drug-protein interactions. Front Bioeng Biotechnol 8:330
49. Lim M, Abdullah A, Jhanjhi N, Supramaniam M (2019) Hidden link prediction in criminal networks using the deep reinforcement learning technique. Computers 8(1):8
50. Tumminello M, Aste T, Di Matteo T, Mantegna RN (2005) A tool for filtering information in complex systems. Proc Natl Acad Sci United States Am 102(30):10421–10426. https://doi.org/10.1073/pnas.0500298102
51. Onnela JP, Chakraborti A, Kaski K, Kertesz J, Kanto A (2003) Asset trees and asset graphs in financial markets. Phys Scr T106:48–54
52. Onnela J-P, Kaski K, Kertész J (2004) Clustering and information in correlation based financial networks. Eur Phys J B-Condens Matter Complex Syst 38(2):353–362
53. Onnela J-P, Chakraborti A, Kaski K, Kertesz J (2003) Dynamic asset trees and black monday. Phys A 324(1):247–252
54. Meng H, Xie W-J, Jiang Z-Q, Podobnik B, Zhou W-X, Stanley HE (2014) Systemic risk and spatiotemporal dynamics of the us housing market. Sci Rep 4(1):1–7
55. Mantegna RN, Stanley HE (1999) Introduction to econophysics: correlations and complexity in finance. Cambridge University Press, Cambridge
56. Yang Y, Yang H (2008) Complex network-based time series analysis. Phys A 387(5–6):1381–1386
57. Tumminello M, Di Matteo T, Aste T, Mantegna RN (2007) Correlation based networks of equity returns sampled at different time horizons. Eur Phys J B 55(2):209–217
58. Martínez V, Berzal F, Cubero J-C (2016) A survey of link prediction in complex networks. ACM Comput Surv (CSUR) 49(4):1–33
59. Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. J Am Soc Inform Sci Technol 58(7):1019–1031
60. Chen T, Guestrin C (2016) XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 785–794
61. Oliveira M, Gama J (2012) An overview of social network analysis. Wiley Interdiscip Rev Data Min Knowl Dis 2(2):99–115
62. Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):10008
63. Huang Z, Lin DK (2009) The time-series link prediction problem with applications in communication surveillance. INFORMS J Comput 21(2):286–303
64. Markowitz H (1952) Portfolio selection. J Financ 7(1):77–91
65. Mutlu EC, Oghaz TA (2019) Review on graph feature learning and feature extraction techniques for link prediction. Proceedings of ACM
66. Barabási A-L, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512
67. Adamic LA, Adar E (2003) Friends and neighbors on the web. Soc Netw 25(3):211–230
68. Zhou T, Lü L, Zhang Y-C (2009) Predicting missing links via local information. Eur Phys J B 71(4):623–630
69. Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. Comput Netw ISDN Syst 30(1–7):107–117
70. Castilho D, Gama J, Mundim LR, de Carvalho ACPLF (2019) Improving portfolio optimization using weighted link prediction in dynamic stock networks. In: International conference on computational science—ICCS 2019. Springer, pp. 340–353
71. Carlsson GE, Mémoli F et al (2010) Characterization, stability and convergence of hierarchical clustering methods. J Mach Learn Res 11(Apr):1425–1470

72. Marti G, Very P, Donnat P, Nielsen F (2015) A proposal of a methodological framework with experimental guidelines to investigate clustering stability on financial time series. In: 2015 IEEE 14th international conference on machine learning and applications (ICMLA). pp 32–37, IEEE
73. Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning: data mining, inference, and prediction. Springer, Berlin

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Douglas Castilho is a professor at Federal Institute of South of Minas Gerais (Brazil) and head-researcher at Laboratory of Technology and Innovation. He obtained his PhD in Computer Science and Computational Mathematics at the University of São Paulo (Brazil) in 2022, with a period as visiting research at INESCTEC/University of Porto (Portugal). He obtained his MSc degree in Computer Science at the Federal University of Minas Gerais (Brazil) in 2014. During his career, he was awarded with the Outstanding Student prize in 2012, granted by the Brazilian Society of Computing. He has been working with machine learning and financial market since 2010. He participated as finalist in the Data Science Game 2017, an international competition of data science held in Paris (France). He is interested in the following areas: Computational Intelligence, Financial Computing, Complex Networks, with an emphasis on Financial Networks, Quantitative Finance, Algotrading and Improvement Techniques, High Frequency Trading and Online Social Networks.

Tharsis Souza is a computer scientist passionate about data-driven products. He is Senior Vice President of Product Management, Modeling Engineering at Two Sigma Investments and Lecturer at Columbia University, Faculty member of the MSc. in Applied Analytics program. Prior to Two Sigma, he spent 10+ years delivering new technology products in a variety of companies from start-ups to Fortune 500's in the USA, Brazil, and the UK. He is an author of scholarly publications and a regular speaker in academic and business conferences. He also enjoys mentoring under-represented students & working professionals. Tharsis holds a Ph.D. in Computer Science from UCL, University of London following an M.Phil. and M.Sc. in Computer Science and a B.Sc. in Computer Engineering.

Soong Moon Kang is an Associate Professor at University College London School of Management. His research interests include social network analysis, complex systems, computational social science, social psychology, innovation and creativity, organization theory, business strategy and entrepreneurship. He holds a Ph.D. in Management Science and Engineering, a M.A. in Sociology and a M.S. in Engineering-Economic Systems from Stanford University, and a degree of Diplom-Ingenieur in Mechanical Engineering from Technische Universität Berlin.

João Gama is a Full Professor at the School of Economics, University of Porto, Portugal. He received his Ph.D. in Computer Science from the University of Porto in 2000. He is EurAI Fellow, IEEE Fellow, and Fellow of the Asia-Pacific AI Association. He is member of the board of directors of the LIAAD, a group belonging to INESC Tec. His main contributions are in the area of learning from data streams, where he has an extensive list of publications. He is the Editor-in-Chief of the International Journal of Data Science and Analytics, published by Springer.

André C. Ponce de Leon Ferreira de Carvalho has a PhD in electronic engineering from University of Kent, UK. He is Full Professor and Director of the Institute of Mathematics and Computer Science of the University of São Paulo, Brazil, Director of the Brazilian Applied AI Research Centre IARA (Research Centre for Smart Environments), a member the Governance Committee of the Brazilian Strategy for Artificial Intelligence (EBIA) and of the Artificial Intelligence Technical Committee of the International Federation for Information Processing (IFIP). His main research interests are artificial intelligence, machine learning and data science.