Network Representations for Multivariate Time-series with Applications in Portfolio Optimization and Deep Learning

Yuanrong Wang

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Computer Science

University College London

February 26, 2024

I, Yuanrong Wang, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

This dissertation probes the role of complex network theory in modelling multivariate time-series systems, a vital aspect in a wide spectrum of contemporary science. Our methodology counters many practical limitations by exploring the sparse topology of time-series data.

Financial time series are marked by persistent discontinuities and low signal-to-noise ratios. In this work, we propose two methodologies predicated on information filtering networks—a noise filtering technique—to address these complexities. These methodologies are subsequently extended to portfolio optimization problems. Inverse Covariance Clustering, a multivariate temporal clustering method, is integrated with contemporary portfolio optimization strategies with the aim of mitigating the impact of time-series discontinuities, colloquially termed as regime shifts in finance. Statistically Robust Information Filtering Network represents a novel framework designed to augment noise filtering in information filtering networks and enhance the signal-to-noise ratio in processed financial time-series data, thereby bolstering the diversification of portfolio construction.

Moreover, we explore the utilization of information filtering networks within the domain of deep learning for modelling multivariate time series. We exhibit the benefits of deploying a filtered, sparse graph predicated on the input time-series network topology, as opposed to a fully connected graph in GNN. Further inspired by this concept, we propose an innovative MLP-like sparse architecture that also leverages network topology, and explicitly considers higher-order interactions. The incorporation of this network topology into both proposed architectures has demonstrated notable efficacy and efficiency in managing multivariate time-series data.

Impact Statement

This thesis adapts numerous theories from network science, commonly associated with theoretical physics, to the realm of time-series modelling—a key focus in many contemporary fields. It leverages a novel approach to viewing time-series correlation through the lens of graphs, facilitating subsequent processing and modelling. This new perspective pushes the boundaries of some existing research by introducing a new research methodology.

Moreover, most applications in this thesis draw from real-world data and possess the potential to address various prevailing challenges, including but not limited to overly large and consuming computational models, and financial noises preventing automation processes. In fact, the author has introduced several core frameworks and concepts, which are now employed by globally recognized companies.

Acknowledgements

The master teaches the trade, but the apprentice's skill is self-made. The expedition towards a PhD is seldom, if ever, without its hurdles and challenges. I am extremely fortunate to have been guided along this journey by my supervisor, Prof. Tomaso Aste, whose steadfast support has been instrumental in my accomplishments, both within and beyond this thesis.

I am deeply grateful to all my friends and colleagues in the lab and at work, particularly Antonio Briola and Danial Saef, who were brave enough to become my coauthors. Your patience and consideration save lives (of many papers and me). Special mention must be made to Dr. Christian Michler, my line manager at work, a mentor, a friend, and a source of constant inspiration.

A journey of this length and intensity would not have been possible without the ceaseless encouragement and company provided by my family and friends. My parents deserve heartfelt appreciation for their unconditional love. I would also like to express my gratitude to my closest friends, DH, JW, LL, CW, and others, who have shown immense tolerance for my eccentricities and maverick.

The concluding words of gratitude are reserved for the mother of three cats and a lizard, the recipient of two prominent acknowledgements in both my undergraduate and PhD theses, the love of my life and my fiancée, beloved Gloria Xu. Our journey together has demonstrated that we are, undeniably, good as two but better as one.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Multivariate temporal sequences inherently form the core of numerous contemporary scientific concepts, and the refinement of such intricate systems' modelling proves challenging due to its elevated dimensionality and interdependencies. However, its successful implementation carries rewards applicable to a multitude of domains. This dissertation utilizes complex network theory as the foundational approach to scrutinize multivariate temporal patterns and explore the interactions between variables.

The first half of the thesis is concentrated on financial time-series data, characterized by a conspicuous low signal-to-noise ratio and the prevalence of abrupt jumps and shifts. Herein, we demonstrate the application of information filtering networks - a network-based noise filtering mechanism - to substantially mitigate noise within time-series modelling. Subsequently, we provide evidence that these methods can be extrapolated to portfolio optimization problems, resulting in enhanced portfolio diversification and increased resilience to market regime fluctuations.

Additionally, stimulated by the insights gained in the financial sector, we adopt a broader perspective to examine the network topology of more general time-series data. The sparse topology generated from the information filtering network based on feature correlation provides the blueprint for innovative designs of neural network architecture, which are concurrently sparse and efficient. The second half of the thesis centres around the design analysis of the novel neural network architec-

ture for time-series data processing.

The topic-related literature is reviewed in Chapter 2, including multivariate market regimes and portfolio optimization in Sections 2.3 and 2.4, and deep learning for multivariate time-series modelling in Section 2.5.

1.1 Financial Time-series Data for Portfolio Optimization

misrepresents genuine underlying trends in the financial markets. It can be caused by various factors and can lead to irrational or misleading investment decisions Finance experiences high noise due to the vast amount of data available, much of which is irrelevant and misrepresents genuine underlying trends or fundamentals of financial markets. Therefore, financial time-series data is characterized by a low signal-to-noise ratio and a high frequency of discontinuities and shifts, which inherently complicates the extraction of reliable statistical measures or the identification of repetitive and significant patterns. Multivariate time-series problems often rely on historical signals and interdependencies, expressed as correlation or covariance, to make predictions for the future. Consequently, forecasting and modelling multivariate time series, particularly within the scope of financial data science, is exceptionally demanding. Portfolio optimization is an increasingly studied complex application within this sphere, predominantly relying on historical statistical analysis. As highlighted in Markowitz's groundbreaking research [1], the optimal allocation of assets in a portfolio is determined by the empirical covariance and mean returns of assets. Therefore, the enhanced estimation and modelling of co-movements of underlying assets inherently refine portfolio design. Inspired by Pozzi et al. [2], we utilize a network to depict the asset universe, wherein each node symbolizes an asset, and the edge connecting two nodes denotes a pairwise similarity measures, such as correlation distance, between them. Further, in Chapter 3, we present a multivariate time-series clustering methodology to be amalgamated with contemporary portfolio optimization strategies, aiming to diminish the impact of discontinuities in time-series, colloquially referred to as regime shifts in finance. In Chapter 4,

we propose a novel framework to augment noise filtering in information filtering networks and bolster the signal-to-noise ratio in the processed financial time-series data, thereby enhancing the diversification of portfolio construction.

Financial multivariate time series often exhibit non-stationarity, characterized by constant jumps and shifts. Consequently, market conditions are perpetually dynamic, and accommodating this inherent non-stationarity within portfolio investment strategies presents a significant challenge. In Chapter 3, we introduce Inverse Covariance Clustering-Portfolio Optimization (ICC-PO). At its core, ICC identifies and clusters market states extrapolated from historical data analytics and predicts the impending market state, which can subsequently be amalgamated with a variety of optimization strategies. Our comprehensive experiments conducted across three distinct markets, namely NASDAQ, FTSE, and HS300, over a decade-long period, underscore the benefits of our proposed algorithm. By applying an identical portfolio optimization technique to the data subset corresponding to a superior cluster, rather than the entire training period, we demonstrate that portfolios can be constructed with markedly elevated Sharpe Ratios, showcasing enhanced statistical robustness and resilience, with considerable diminutions in the maximum loss in extreme scenarios. This effect persists across varying markets, periods, optimization techniques, and portfolio asset selections, corroborating the ability of our methodology to adequately account for the time-series shifts in historical empirical data.

Financial data is rife with various noise sources, e.g., market sentiments, news and media, hype and speculation, etc., which culminate in a remarkably low signal-to-noise ratio. Consequently, time-series filtering and signal-processing techniques are extensively employed in finance to extract a smooth and significant trendline prior to further processing. Existing information filtering networks have proven highly effective in de-noising the empirical correlation and covariance of multivariate time series. However, the construction process of such networks imposes certain topological structures, which may inadvertently introduce forms of noise. In Chapter 4, we propose the Statistically Robust Information Filtering Network

(SR-IFN). This novel method is premised on bootstrapping and seeks to discard redundant edges during the network formation process. We leverage SR-IFN to further mitigate noise within financial data and subsequently construct portfolios based on the filtered correlation topology. The portfolios generated in this manner have been demonstrated to offer greater diversification with elevated returns and reduced volatility across three countries, thereby substantiating the efficacy of SR-IFN in processing multivariate time-series data.

1.2 Neural Network Designs Based on Time-series Data Topology

In the first half of this thesis, we delve into two distinct methodologies for managing discontinuities and low signal-to-noise ratios in multivariate financial time-series data. Both approaches are predicated on the information filtering network, constructed from the correlation of the input data. The sparse topology extracted from the resulting network effectively sieves out insignificant components while preserving critical information. Inspired by these outcomes, we strive to incorporate the topology of the input data as a prior when designing and constructing neural network architectures, specifically tailored for processing time-series data. In Chapter 6, we leverage the information filtering network as a sparse graph to supplant the commonly used Laplacian graph in Graph Neural Networks (GNNs), marking our initial successful venture in combining information filtering networks with neural networks. Furthermore, encouraged by the success of this initial integration, in Chapter 5, we utilize the simplicial levels of the input time-series' correlation topology to guide the design of a sparse Multi-Layer Perceptron (MLP)-like neural network.

The scope of multivariate time-series prediction applications extends from everyday business tasks, such as sales volume forecasting and traffic prediction, to more specialized domains like biostatistics and action recognition. Multivariate time-series forecasting methodologies presuppose interdependencies among dynamically evolving variables, which help capture systematic trends. Specifically,

the forecast for each variable depends not only on its historical temporal information but also on other variables. Deep learning has leveraged spatio-temporal neural network architecture to model and predict multivariate time series. The temporal component, exemplified by Long Short-Term Memory (LSTM) and Recurrent Neural Networks (RNN), captures local patterns of individual time series. Simultaneously, the spatial element, represented by Graph Neural Networks (GNN) and Convolutional Neural Networks (CNN), aggregates the interdependencies between them. In Chapter 5, we integrate a filtering module into the spatio-temporal architecture to generate a sparse topological graph based on the information filtering network of the input data topology. This sparse graph is then fed into and supplants the traditionally used Laplacian graph in the spatial component of a GNN. A series of experiments demonstrate that this proposed sparse substitution offers superior performance. Furthermore, when compared with the state-of-the-art Diffusion Convolutional Recurrent Neural Network (DCRNN), the results indicate that a combination of a less complex GNN with graph sparsification and filtering can achieve equal or superior efficiency than complicated state-of-the-art models in multivariate time-series regression tasks.

Building upon the initial successful integration of the information filtering network and the graph neural network, using the topology of input data, we proceed to directly leverage this topology in the design phase of a sparse Multi-Layer Perceptron (MLP)-like neural network in Chapter 6. Utilizing sophisticated network-based information filtering techniques, we succeed in identifying the simplicial structures inherent in the underlying input time-series data. Subsequently, we employ each neural network layer to represent each order of simplex, with the connections between each layer representing the formation of a higher-order simplex by a lower-order simplex. This results in an innovative neural network that constitutes a sparse higher-order graphical architecture, independent of the message-passing framework. The effectiveness of this novel approach is demonstrated in time-series and tabular regression problems traditionally viewed as challenging for deep learning. The findings underscore the benefits of this innovative design, which can equal

or exceed the performance of state-of-the-art machine learning and deep learning models while utilizing a significantly reduced number of parameters.

Finally, we summarize the contributions in Chapter 7 and consider extensions and potential future work.

The main chapters of this thesis are based on published and publishing in-progress works:

- Chapter 3: Wang, Y., & Aste, T. (2021). Dynamic portfolio optimization with inverse covariance clustering. Expert Syst. Appl., 213, 118739.
- Chapter 4: Wang, Y., Briola, A., & Aste, T. (2023). Topological Portfolio Selection and Optimization. Submitted to 12th International Conference on Complex Networks & Their Applications.
- Chapter 5: Wang, Y., & Aste, T. (2022). Network Filtering of Spatial-temporal GNN for Multivariate Time-series Prediction. Proceedings of the Third ACM International Conference on AI in Finance.
- Chapter 6: Wang, Y., Briola, A., & Aste, T. (2023). Accepted in Proceedings of the 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML) at the 40th International Conference on Machine Learning.

Chapter 2

Background Literature

2.1 Multivariate Time-series Modelling

Forecasting time-series data has been a cornerstone problem in the fields of statistics, data science, and machine learning for an extended period. Techniques for this have evolved from traditional pattern recognition to modern machine learning. Univariate time-series forecasting concentrates on analyzing independent time series by identifying temporal patterns rooted in historical behaviours. Techniques such as the moving average (MA), the auto-regressive (AR), the auto-regressive moving average (ARMA), and the autoregressive integrated moving average (ARIMA) [3] exemplify these. The suitability of modern machine learning models, such as Long Short-Term Memory (LSTM) units, for handling this problem has been demonstrated in numerous studies, e.g., FC-LSTM [4] and SMF [5].

Multivariate forecasting, on the other hand, engages with a correlated set of time series. The Vector Auto-Regressive model (VAR) and the Vector Auto-Regressive Moving Average model (VARMA) [6, 7] extend the aforementioned linear models into a multivariate space by accounting for the interdependency among time series. However, VAR solely relies on lagged data and doesn't incorporate exogenous variables or external factors that may affect the time series data. While VARMA does incorporate additional moving average features, they both depend on the assumption of stationarity and linearity and suffer from the curse of dimensionality. Initial attempts at integrating Convolutional Neural Networks (CNN) and

Recurrent Neural Networks (RNN) were directed at learning local spatial dependencies and temporal patterns [8, 9]. The introduction of deep learning methods add additional non-linearity power to model fitting and is more suitable for non-stationary use cases. Further work involves the state space model in Deep-State [10] and the matrix factorization approach in DeepGLO [11].

2.2 Sparse Correlation and Covariance Structure

Two computational method families employ sparse approximation techniques to estimate the inverse covariance matrix. The sparsification is effective because the least significant components in a covariance matrix are often largely prone to small changes and can lead to instability. Sparsified models filter out these insignificant components, and thus improve the model's resilience to noise. As correlation is a scaled form of covariance, filtering and sparsification methods are equivalently applicable in both cases.

2.2.1 Graphical Models

A widely used approach for inverse covariance estimation is based on graph models. Meinshausen and Buhlmann in 2006 [12] regard the zero entries in the inverse covariance matrix of a multi-variable normal distribution as conditional independence between variables. These structural zeros can thus be obtained through neighbourhood selection with LASSO regression by fitting a LASSO to each variable and using the others as predictors. Similar methods that maximize L_1 penalized log-likelihood have been studied by Yuan and Lin [13] and Banerjee et al. [14]. In 2008, Friedman et al. [15] developed an efficient Graphical LASSO that uses L_1 norm regularization to control the sparsity in the precision matrix. The sparse inverse covariance matrix can be obtained by minimizing the regularized negative log-likelihood [16]:

$$\Sigma^{-1}_{\text{glasso}} = \min_{\Sigma^{-1}}(-\log\det\Sigma^{-1} + \text{Tr}(\hat{\Sigma}^{-1}\Sigma^{-1}) + \lambda||\Sigma^{-1}||_1) \tag{2.1}$$

where $\hat{\Sigma}^{-1}$ is the empirical inverse covariance, $||\Sigma^{-1}||_1$ denotes the sum of the absolute values of Σ^{-1}, and λ is the regularization constant, optimised by cross-validation.

2.2.2 Information Filtering Network

An alternative approach that uses information filtering networks has been shown to deliver better results with lower computational burden and larger interpretability [17]. In the past few years, information filtering network analysis of complex system data has advanced significantly. It models interactions in a complex system as a network structure of elements (vertices) and interactions (edges). The best-known approach, the Minimum Spanning Tree (MST) was first introduced by Boruvka in 1926 [18] and it can be solved exactly (see [19] and [20] for two common approaches). The MST reduces the structure to a connected tree which retains the larger correlations. To better extract useful information, Tumminello et al. [21] and Aste and Di Matteo [22] introduced the use of planar graphs in the Planar Maximally Filtered Graph (PMFG) algorithm. Recent studies have extended the approach to chordal graphs of variable sparsity [23, 24]. Research fields ranging from finance [17] to neural systems [25] have applied this approach as a powerful tool to understand high dimensional dependency and construct a sparse representation. It was shown that for chordal information filtering networks, such as the Triangulated Maximally Filtered Graph (TMFG) [23], one can obtain a sparse precision matrix that is positively definite and has the structure of the network paving the way for a proper L_0-norm topological regularization [26], detailed algorithm see Algorithm 1 Further study in Maximally Filtered Clique Forest (MFCF) [27] extends the generality of the method by applying it to different sizes of cliques. This approach has proven to be computationally more efficient and stable than Graphical LASSO [15] and covariance shrinkage methods [28, 29, 30], especially when few data points are available [17, 22]. In addition to shrinkage-like sparsification methods, Random Matrix Theory (RMT) offers an alternative approach to reduce the impact of noise and irrelevant information in covariance matrices [31, 32, 33]. MT-based sparsification methods use insights from the theory to identify and retain

significant eigenvalues and corresponding eigenvectors while discarding the less informative ones. However, RMT-based methods are usually computationally heavier and more sensitive to parameter selection (the impact of regime shifts is higher) than shrinkage-based methods.

Algorithm 1 TMFG built on the similarity matrix $\hat{\mathbf{C}}$ to maximise the likelihood of features' relevance.

Input Similarity matrix $\hat{\mathbf{C}} \in \mathbf{R}^{n,n}$ from a set of observations $\{x_{1,1},\ldots,x_{s,1}\},\{x_{1,2},\ldots,x_{s,2}\}\ldots\{x_{1,n},\ldots,x_{s,n}\}$.
Output Sparse adjacency matrix $\boldsymbol{A}$ describing the TMFG.

Initialize four empty sets: $\mathscr{C}$ (cliques), $\mathscr{T}$ (triangles), $\mathscr{S}$ (separators) and $\mathscr{V}$ (vertices);
Initialize an adjacency matrix $\boldsymbol{A} \in \mathbf{R}^{n,n}$ with all zeros;
$\mathscr{C}_1 \leftarrow$ tetrahedron, $\{v_1,v_2,v_3,v_4\}$, obtained choosing the 4 entries of $\hat{\mathbf{C}}$ maximising the similarity among features;
$\mathscr{T} \leftarrow$ the four triangular faces in $\mathscr{C}_1$: $\{v_1,v_2,v_3\},\{v_1,v_2,v_4\}$, $\{v_1,v_3,v_4\},\{v_2,v_3,v_4\}$;
$\mathscr{V} \leftarrow$ Assign to $\mathscr{V}$ the remaining $n-4$ vertices not in $\mathscr{C}_1$;
while $\mathscr{V}$ is not empty **do**
    Find the combination of $\{v_a,v_b,v_c\} \in \mathscr{T}$ (i.e. t) and $v_d \in \mathscr{V}$ which maximises MAXIMUMGAIN($\hat{\mathbf{C}}$, $\mathscr{V}$, t);
    $\{v_a,v_b,v_c,v_d\}$ is a new 4-clique $\mathscr{C}$, $\{v_a,v_b,v_c\}$ becomes a separator $\mathscr{S}$, three new triangular faces, $\{v_a,v_b,v_d\}$, $\{v_a,v_c,v_d\}$ and $\{v_b,v_c,v_d\}$ are created .
    Remove v_d from $\mathscr{V}$;
    Remove $\{v_a,v_b,v_c\}$ from $\mathscr{T}$;
    Add $\{v_a,v_b,v_d\}$, $\{v_a,v_c,v_d\}$ and $\{v_b,v_c,v_d\}$ to $\mathscr{T}$;
end while
For each pair of nodes i,j in $\mathscr{C}$, set $\boldsymbol{A}_{i,j}=1$;
return $\boldsymbol{A}$.

function MAXIMUMGAIN($\hat{\mathbf{C}}$, $\mathscr{V}$, t)
    Initialize a vector of zeros $g \in \mathbf{R}^{1\times n}$;
    for $j \in t$ **do**
        for $v \notin \mathscr{V}$ **do**
            $\hat{\mathbf{C}}_{v,j}=0$
        end for
        $g = g \oplus \hat{\mathbf{C}}_{v,j}$
    end for
    return $\max\{g\}$.
end function

2.3 Market Regimes

2.3.1 Fat-tails and asymmetric return distributions

Non-Gaussian probability distributions in financial returns are primarily caused by the complex and dynamic nature of financial markets. Typical events like trading mechanisms and liquidity dynamics under market microstructure, time-varying volatility (Heteroskedasticity), and volatility clustering all cause the market to exhibit non-Gaussian behaviors. Hence, normal distributions do not represent well the observed probability distribution of the financial market's asset price returns. Indeed, they instead have a larger number of small returns following Gaussian statistics, but also a larger number of very large positive and negative returns of sizes that would be impossible with Gaussian statistics [34, 35, 36]. They also have often asymmetric distributions with larger negative returns (losses) more likely than large positive ones (gains). Several alternative probability distributions have been used in the literature, namely, Student-t [37, 38, 39], Laplace [40, 41, 42] and Pareto-Levy [43, 44] distributions. In addition, alternative approaches to accounting for asymmetry have been taken into account, with early works by Markowitz himself which in 1959 [45] employed semi-covariance (the covariance from negative returns only) as an alternative risk measure to better describe the downside market moves. Furthermore, limited sample size is a critical contributing factor to estimation errors. Yet, simply extending the sample size introduces data from events that happened far in the past which are likely to be less representative of present market conditions. Hence, methods ranging from shrinkage [28], to LASSO regularization [46, 47], and Monte Carlo based re-sampling [48, 49] have been used to reduce this issue.

2.3.2 Non-stationarity

Assumptions regarding market stationarity and portfolio re-allocation are often considered together since multi-period investment is proven to be an effective solution to mitigate the effect of Market turmoil. Several contributions have shown that dynamic reallocation brings improvements in the resilience to market volatility with respect to the original single-period portfolio diversification methods [50, 51, 52].

Nonetheless, such methods still fail to address structured market movements. Indeed, accounting for such changes requires forecasting the future market state. Further studies on market states have been proposed to model and predict the intrinsic properties of these dynamics, and two main streams are discussed below. The first one uses the Markov decision process to model the transition probability between different market regimes. An approach based on Hidden Markov Model (HMM) has demonstrated great efficiency and validity [53]. However, it often encounters problems mainly associated with the curse of dimensionality, as the dimensionality of hidden states is linear to the number of assets considered [54, 55]. Bayesian methods help to mitigate HMM's dimensionality issue, typical methods include Markov Chain Monte Carlo (MCMC) methods [56, 57], such as Gibbs sampling or Metropolis-Hastings used for parameter estimation in HMMs, Sequential Monte Carlo (SMC) for state estimation in HMMs [58], and Nonparametric Bayesian models, like the Dirichlet Process, that can adaptively model the complexity of an HMM without specifying the number of states beforehand [59]. On the other stream, researchers believe that the market comprises mixed multivariate distributions, and each state effectively corresponds to a distribution. Hence, temporal clustering methods such as Gaussian Mixture Model (GMM) [60, 61, 62], K-Means Clustering [63, 64, 65] have been applied for this purpose. Then, portfolios can be re-adjusted according to the predicted state with a selected re-allocation period. Yet, these methods are often based on strong assumptions and they are not originally designed for time series, which results in issues e.g., GMMs assume that each cluster follows a Gaussian distribution, and K-Means may produce unequal cluster size due to noise and extreme data. This is also to some extent the approach of the decision-theoretic Bayesian methods [66, 67, 68], such as the Black-Litterman model [69, 70], which includes in the optimization a Bayesian prior on the future state.

2.3.3 Market Regimes Clustering

After the initial pitfall of Markov-Model-based methods [53, 55, 54], mainly due to the curse of dimensionality, literature has started to look for alternative methods to cluster similar temporal data points into the same group based on certain com-

parison criteria. Such temporal clustering methods can mostly be divided into two approaches: subsequent clustering and point clustering. Subsequent clustering uses a sliding window to capture a period of data points and analyze for recurrent patterns [71, 72]. The four main methods of subsequent clustering are: (i) hierarchical [73, 74, 75]; (ii) partitioning [76, 77]; (iii) density-based [78, 79, 80] and; (iv) pattern discovery [81, 82, 83]. These methods have all shown applicability to financial data analysis and portfolio construction. An alternative approach is point clustering that, instead of measuring spatial similarity between two slices of time series, looks at each temporal point individually, and assigns this multivariate observation to an appropriate cluster based on distance metrics [84, 85, 86]. Hence, in point clustering, the choice of distance is core. In macroeconomics, the market states are not the representation of solely upward or downward trends of the market, but also the relative dynamics of equity prices, which naturally makes correlations a convenient choice of collective dynamics. A stationary correlation structure was assumed as the common approach in the industry in the 90s [70, 87], which was, however, later shown to be overly presumptive [88, 89, 90]. Consequently, research has been devoted to studying time-varying correlations. Models, such as Generalized Autoregressive Conditional Heteroskedasticity (GARCH) [91] and the Dynamic Conditional Correlation (DCC) [92] have been proposed for simulating and predicting this dynamical correlation. However, most of these models suffer from the curse of dimensionality and can only be applied to a limited number of assets, as the number of parameters increases super-linearly with the number of variables.

In 2017, Hallac et al. proposed the Toeplitz Inverse Covariance Clustering (TICC) [93] algorithm, originally devised for electric vehicle action sensors. It classifies states based on the likelihood measures of short subsequences of observations and corresponding sparse precision matrix. After clustering, the precision matrix of each state is estimated under a Toeplitz constraint where each descending diagonal of the covariance matrix from left to right is constant, i.e., and the covariance structure between variables (or features) is assumed to be constant along the diagonal. Inspired by TICC, Procacci and Aste in 2020 [94] proposed a closed-

related methodology named Inverse Covariance Clustering (ICC). This approach provides a point clustering of observations also enforcing temporal consistency by penalizing switching between states. The ICC method also uses sparse precision matrices but sparsification is attained via information filtering networks (see next Subsection). One main advantage of ICC, compared to TICC, is its flexibility in the selection of similarity measures. It was also stated in their original paper that different clustering distances separate market states differently. For example, likelihood distance distinguishes better with pre- and post-crisis periods, Euclidean distance discriminates well between bull and bear states, and Mahalanobis distance detects both crisis periods and bull/bear states.

2.4 Portfolio Optimization

2.4.1 Mean-variance Optimization

Despite the unquestionable merits and pioneering status of Markowitz's mean-variance optimization (MVO) approach, there are some major assumptions, and several bad applications, that reduce its efficacy for practical implementations. Firstly, MVO assumes that asset returns follow a finite-variance distribution and higher moments are monotonic with variance. Many financial theories simplify this assumption by adopting a normal distribution, and consequently, the models that utilize such theories do not account for extreme market situations. Moreover, the variance of a normal distribution as a risk measure does not distinguish between upside and downside moves in the market. Secondly, the MVO rely on the inversion of a covariance matrix and this operation makes the method highly sensitive to estimation error especially when the covariance is estimated on a relatively short time period and when such a past period is not representative of the future. Indeed, historical financial market data is never a good representation of the true underlying distribution as the observations are often partial. Furthermore, most MVO implementations are assuming market stationarity, which is that the mean and variance are assumed constant in each asset, while the correlation is static between assets. MVO is designed to avoid unsystematic risks by optimizing diversification. However, the sys-

tematic risks from market movements are not addressed by the MVO methodology and this is usually the most significant factor for investment decisions. Lastly, a single-period investment will almost never work in reality. A constant reallocation is vital to respond to the rapidly changing environment.

2.4.2 Dynamic Portfolio Allocation

The traditional Markowitz model optimizes on a single period only, and it relies heavily on the assumption of constant asset mean vectors and covariance matrix. Therefore, this static and long investment horizon is inadequate in a dynamic marketplace. Yet, the mean-variance criteria inspire the development of multi-period dynamic portfolio construction. The dynamic portfolio optimization field currently follows two main streams. A discrete-time model was proposed by Samuelson in 1969 [95] and developed since by Hakansson, Grauer and others [96, 97, 98]. It separates an investment horizon into discrete periods, and the portfolio can be reallocated at the end of each period. In contrast, a continuous-time model was introduced by Merton [99] in the same year, and together with further studies described the continuous rebalancing of securities for a fixed planning horizon [100, 101, 102].

The two alternative assumptions that are often made in dynamic portfolio optimization problems are market completeness and investment horizon. A complete market is an approximation to the real market where friction, transaction costs and asset liquidity exist, and a dynamic portfolio has to consider those real-world factors [103, 104, 105, 106]. A more ideal scenario is instead the incomplete market where some conditions are waived so that research can only focus on the dynamic asset selection process and ignore some practical issues [107, 108, 109, 110]. Similarly, an infinite horizon is a naive assumption to a finite horizon where investors will withdraw investment with an exit time. The earlier pioneers [96, 100, 104] in this field, such as Samuelson [95], Merton [99, 101], began with the infinite horizon assumption, while later researchers in the 90s and the beginning of the millennium [111, 112, 106, 110] led by He & Pearson [113] and Karatzas et.al [114] started to introduce the finite horizon into the problem.

2.4.3 Network-based Portfolio Optimization

Traditional methods of portfolio optimization largely rely on empirical covariance and correlation, which predominantly capture linear dependencies among assets. Nevertheless, financial market networks synthesized from historical data, tend to encapsulate the entire system's complexity, including non-linearities, and often yield superior outcomes in terms of portfolio construction. Pozzi et al. [2] in 2013 found that risk is not uniformly distributed across the market, with peripheral assets of a financial network demonstrating greater success in diversification and leading to superior performance. This finding has subsequently led research to focus on quantifying peripherality and constructing highly diversified, low-risk portfolios. To quantify peripherality, a graph must be initially treated using network filtering methodologies, such as information filtering networks, which transform the complete graph constructed from the correlation matrix or other linear [115, 116, 117] and non-linear [118, 119, 120] similarity measures to a sparse network retaining only strongest relationships. Subsequently, different centrality measures, including degree centrality, betweenness centrality, eccentricity, and closeness centrality, are computed for each node. Nodes are then ranked in ascending order to be incorporated into the portfolio with equal or Markowitz weights [121, 2], or weights that are calculated based on the centrality measures [122, 123]. Additional research includes network-based allocation with machine learning [124], cross-sectional equity sector portfolio construction [125, 126], and graph clustering-based portfolio construction [127, 128, 129]. Other network-based analyses on practical portfolio execution have been explored, e.g., liquidity [130] and transaction costs [**?**], however, such methods have yet to be combined in the network-based portfolio construction process.

2.5 Deep Learning for Multivariate Time-series Modelling

2.5.1 Neural networks

2.5.1.1 Spatio-temporal Neural Networks

Existing research in multivariate time series forecasting can be broadly divided into two primary categories: statistical methods and deep learning-based methods. Statistical approaches usually assume linear correlations among variables (i.e., time series) and use their lagged dependency to forecast through a regression, as exemplified by the vector auto-regressive model (VAR) [131] and Gaussian process model (GP) [132]. In contrast, deep learning-based methods, such as LSTNet [133] and TPA-LSTM [134], utilize Convolutional Neural Networks (CNN) to identify spatial dependencies among variables and combine them with Long Short-Term Memory (LSTM) networks to process the temporal information. Although they have been widely applied across various application domains, including finance [135] and weather data [136], these architectures do not explicitly model dependency structures among variables, being strongly limited on the interpretability side. Recently, spatio-temporal graph neural networks (STGNNs) [137, 138] have attracted interest reaching state-of-the-art performances, as exemplified by MTGNN [139]. STGNNs integrate graph convolutional networks and sequential recurrent models, with the former addressing non-Euclidean dependencies among variables and the latter capturing temporal patterns. The inclusion of advanced convolutional or aggregational layers accounting for sparsity and higher-order interactions has resulted in further improvements of STGNNs [140, 141, 142, 143, 144, 145, 146, 147].

2.5.1.2 Sparse Neural Networks

Recent advancements in artificial intelligence have exacerbated the challenges related to models' computational and energy efficiency. To mitigate these issues, researchers have proposed new architectures characterized by fewer parameters and sparse structures. Some of them have successfully reduced the complexity of very large models to drastically improve efficiency with negligible performance degrada-

tion [148, 149, 150, 151, 152, 153, 154, 155]. Others have not only simplified the architectures but also enhanced models' efficacy, further demonstrating that fewer parameters yield better model generalization [139, 156, 157, 158, 159, 160, 161, 162].

Nonetheless, in the majority of literature, sparse topological connectivity is pursued either after the training phase, which bears benefits only during the inference phase, or during the backpropagation phase which usually adds complexity and run-time to the training. In contrast, network-inspired pruning methods incorporate pruning at the earliest stage of the process, allowing for the establishment of a foundational topological architecture that can then be elaborated upon [163, 164, 165].

2.5.2 Graph Neural Networks

2.5.2.1 Spatio-temporal GNNs

Spatio-temporal graph neural network has been proposed recently for multivariate time-series problems. To capture the correlation between time-series in the spatial component, each time-series is modelled as a node in a graph whereas the edge between every two nodes represents their correlation. Early work applies spatio-temporal GNN for traffic forecasting [166, 167, 168, 169, 145]. Further studies have been extended to other fields, e.g., action recognition [170, 171] and bio-statistics with many interesting works for COVID-19 [172, 173, 174]. For financial applications, Matsunaga et al. [175] is one of the first studies exploring the idea of incorporating company knowledge graphs directly into the predictive model by GNN. Later, Hou et al. [176] proposed to use a variational autoencoder (VAE) to process stock fundamental information and cluster it into graph structure. This learned adjacency matrix is then fed into a GCN-LSTM for further forecasting. Similar work has been done by Pillay & Moodley [177] with a different model architecture called Graph WaveNet. The most recent advancement is a spatio-temporal GNN for portfolio/asset management proposed by Amudi [178]. They combine a stock sector graph, a correlation graph and a supply-chain graph into one super graph and use the multi-head attention in GAT as a sparsification method to select the meaningful subgraph for prediction. In line with this work, we focus on filtered/sparsified (inverse) correlation graphs generated from matrix filtering/sparsification techniques.

2.5.2.2 Sparse GNNs

Much literature has discussed graph sparsification in GNN. Some, by including regularization, reduce unnecessary edges, which can largely improve the efficiency and efficacy of large-scale graph problems [141, 142]. Some leverage stochastic edge pruning in graphs as a dropout-equivalent regularization to enhance the training process [143, 144]. Others train the GNN to learn sparsification as an integrated part before applying it to downstream tasks. NeuralSparse learns to sample k-neighbour subgraph as input for GNN [145]. Luo proposes to prune task-irrelevant edges [146]. Kim uses the disconnected edges of sparse graphs to guide attention in GAT [147].

Chapter 3

Multivariate Time-series Clustering for Dynamic Portfolio Optimization

3.1 Introduction

Multivariate time series are high-dimensional and usually require carefully designed mathematical models to describe accurately. Especially, constant jumps and shifts in financial time-series data require additional consideration and extensive parameters to describe [179]. To identify the structural non-stationarity, many multivariate time-series clustering methods have been proposed [76, 77, 78, 79, 80, 81, 82, 83]. In this chapter, we introduce a clustering method that is based on the sparse network of inverse covariance of the underlying time series. Moreover, as an application, we apply it to portfolio optimization problems.

In the field of asset management, the problem of portfolio allocation has gained unprecedented popularity over the past few years. Constructing a good portfolio combines the art and science of balancing between trade-offs and the aim of meeting long-term financial goals. The simple core of any portfolio optimization is to assign optimal weights to each portfolio's component in order to minimize investment risk and maximize the return. In 1952, Markowitz [1] demonstrated that, by assuming risk to be quantifiable by the variance of the portfolio's returns, the optimal weights that minimize the portfolio's variance at a given average portfolio's return can be computed with a simple and exact formula. However, Markowitz's

theoretical maximum is attained only in-sample, on the training dataset, whereas off-sample, on the test set where investment is made, performances of Markowitz's portfolio can be largely sub-optimal, as the estimated means and variances of assets from historical data are subject to estimation error. In off-sample periods, these errors can lead to suboptimal allocations as the true parameters may differ. Namely, it is essentially a product of in-sample data and may not capture the complexities of future market dynamics, such as regime shifts, making historical optimization less relevant in different market environments.

Markowitz's modern portfolio theory is the foundation of modern quantitative asset management. There are however two main limitations in Markowitz's assumptions. The first limitation concerns the use of the portfolio's variance as a measure of risk. The variance (when defined) is indeed a measure of the width of the distribution but there are other properties that are better measures of risk (e.g. the value at risk) and might not be reducible to the variance from shifting and scaling transformations. The second limitation concerns the ability to estimate the (future) means and covariance of the asset's returns in the portfolio.

After Markowitz's seminal work, many portfolio selection methodologies have been introduced to cure the first limitation concerning the reliance on variance for risk quantification and nowadays there are several well-established approaches that go well beyond the use of variance as sole risk measure [180]. Furthermore, with the enormous development of machine learning optimization techniques there are presently virtually no limitations in constructing optimal portfolios based on any kind of risk measure [181, 182, 183]. However, balancing the objective function and loss between different risk measures is difficult to design and results in overfitting and low interpretability.

Addressing the second limitation is harder. Indeed, normally, one does not have information from the future that would allow one to set the future properties of the asset's multivariate distribution. Therefore, the reliance on past observations and the assumption that they will significantly represent the future is hard to avoid. Nonetheless, markets are not stationary, it is common knowledge that they

cyclically pass through bull and bear states and occasionally deepen into crisis periods. For each of these periods, the market prices' returns have different statistical properties and they are not describable by means of a unique multivariate probability distribution, and the longer the period, the less resembles it is to a Gaussian distribution. This is especially relevant for factors that matter most to the management of portfolio risk. A long investment horizon consists of many crisis periods, and extreme crisis periods have a distribution with fatter tails and they tend to be more asymmetrical with the left tail having a larger probability for large losses than the right tail for equivalent gains. Portfolio constructions must take into account these differences and devise different investment strategies for each market condition. This is indeed the ground basis for any dynamic asset allocation. However, such a wise allocation would imply the knowledge of the future market state and forecasting it from past observations is not an easy task.

Covariance is a Gaussian measurement of assets' historical properties, which suffers the aforementioned downside. However, it is mostly intuitive and simple to work with and widely used in modern portfolio construction. Therefore, in this chapter, we provide an algorithm termed Inverse Covariance Clustering-Portfolio Optimization (ICC-PO) to address the non-stationarity problem, by identifying the inherent market states and forecast the most likely future state. The Inverse Covariance Clustering (ICC) [94] is a novel temporal clustering method for market states clustering. In this chapter we propose to make use of this temporal clustering classification, constructing different optimal portfolios associated with two ICC market state clusters. The clusters are constructed in the in-sample training set (the past) and then are used separately to train the portfolio optimizer of choice which is then tested on an off-sample period following the training set (the future). For the optimization, we used two approaches based on the classical Markowitz's approach but devised to have only positive weights (no short-sellings). They are the Sequential Least Square Quadratic Programming (SLS) and the Critical Line Algorithm (CLA). Let us note, that the ICC-PO approach allows the use of any optimization method of choice.

We tested the approach with three extensive experiments with daily data, from 2010 to 2020, from three different markets: NASDAQ, FTSE and HS300. For each market, we selected 100 largest market capitalization constituent stocks and quantified the off-sample performances of portfolios constructed from in-sample training data using separately the two ICC-market states. We demonstrate that the difference in returns and risks (computed on the testing set) between the two optimal portfolios, constructed from the two ICC-market states (on the training set), is very large with Sharpe Ratios that more than double and with very large differences in the likelihoods of large negative returns that can have up to three times smaller quantiles (i.e. the value at risk). We provide a simple criteria to forecast the best performing out-of-sample market state which we named 'State 0'. Our results also show that sparsification of the inverse covariance matrix through information filtering networks [23, 17] improves the results, this is a confirmation of a previous result [184] extended however in this chapter to a different dataset, different portfolio optimizers and different markets. The robustness of the method is tested by gathering statistics over 100 re-sampling of consecutive train-test sets randomly selected across the 10 years period 2010 to 2020. Furthermore, reliance on portfolio basket choices is tested by doing the same experiments with a random selection of 100 stocks instead of the 100 most capitalized. For simplicity and demonstration, we use linear correlation/covariance in our experiments, further extension includes rank correlation/covariance and copula which models the joint distribution of asset returns and includes covariance.

The main contribution of this chapter consists of the demonstration that market observations at different times can be classified into different states. Such states have distinct statistical properties, and they continue to be separable in log-likelihood after the in-sample training period showing temporal persistence. Such persistence enables us to predict the best-performing state with a higher log-likelihood in the off-sample investment period. Moreover, in this chapter we confirm the intuitive argument by Procacci & Aste [184] that a model with a larger log-likelihood must perform better for portfolio optimization with respect to one

with a lower log-likelihood.

3.2 Methodologies

In the present chapter, we combine ICC clustering with market state forecasting to be used for portfolio optimization. Let us list in this Section the main methods we use in our approach.

3.2.1 Inverse covariance temporal clustering for portfolio optimization (ICC-PO)

Let's consider a set of n assets with $\mathbf{r}_t \in \mathbb{R}^{1\times n}$ the vector of returns at time t. The corresponding vector of their expected values is $\mu = \mathbb{E}(\mathbf{r}_t) \in \mathbb{R}^{1\times n}$ and their covariance matrix is $\Sigma = \mathbb{E}((\mathbf{r}_t - \mu)^\top(\mathbf{r}_t - \mu)) \in \mathbb{R}^{n\times n}$. The ICC clustering method depends on the choice of a gain function, $G_{t,k}$, which is a measure that qualifies the gain when the time t returns, $\mathbf{r}_t$, are associated with cluster k. Indeed, the ICC approach gathers together in cluster k observations that have the largest gain in such a cluster with respect to any other cluster: $G_{t,k} > G_{t,h}$ for all $h \neq k$. For instance, in [94] it was used

$$G^{Eu}_{t,k} = -(\mathbf{r}_t - \mu_k)(\mathbf{r}_t - \mu_k)^\top \tag{3.1}$$

where μ_k is the sample mean return computed from the observations in cluster k. This gain is minus the square of the Euclidean distance between the observation and the centroid of cluster k. A distance associated with the likelihood for multivariate normal distributions is instead

$$G^{No}_{t,k} = \frac{1}{2}\ln|\hat{\Sigma}_k^{-1}| - n\frac{d^2_{t,k}}{2}, \tag{3.2}$$

with

$$d^2_{t,k} = (\mathbf{r}_t - \hat{\mu}_k)\hat{\Sigma}_k^{-1}(\mathbf{r}_t - \hat{\mu}_k)^\top \tag{3.3}$$

the Mahalanobis distance where $\hat{\Sigma}_k$ is the sample covariance computed form the observations in cluster k. While for the multivariate Student-t one has

$$G^{St}_{t,k} = \frac{1}{2}\ln|\hat{\Sigma}_k^{-1}| - \frac{\nu+n}{2}\ln\left(1+\frac{d^2_{t,k}}{\nu}\right) \tag{3.4}$$

where, in this case, ν is the degree of freedom.

We extensively tested all these gain functions observing that $G^{Eu}_{t,k}$ is particularly efficient in selecting clusters with a prevalence of positive or negative returns but it is performing poorly in the portfolio optimization problem. The normal and Student-t likelihood-related gains have similar performances, but $G^{St}_{t,k}$ turns out to be on average larger and we adopted it for the experiments we present in this chapter. We also tested an hybrid distance $G_{t,k} = c_1 \ln|\hat{\Sigma}_k^{-1}| - c_2 d^2_{t,k}$ with the two arbitrary constants, c_1 and c_2, that allow to gauge between the effects of the natural log of determinant of the covariance (as a part of entropy term up to a constant that depends on the dimension of the Gaussian distribution) and the Mahalanobis distance term. The measure of the determinant of covariance is an equivalent estimation of the differential entropy of the multivariate system, while Mahalanobis distance measures between points and distributions.

The ICC approach uses sparse inverse covariance that was shown to improve considerably the results over the full covariance in terms of the increase of gains. As a sparsification technique we used the sparse inverse constructed with TMFG information filtering graphs [23] using the local-global (LoGo) inversion procedure described in [17], where the elements of the inverse are computed by inverting local sample covariance matrices from only four variables at the time and adding them up. The result is a sparse inverse covariance with $3n-6$ non-zero entries in the upper diagonal (instead of $n(n-1)/2$ in the full matrix). Such a matrix is positively defined, if the number of observations is larger than four, independently of the size of the whole matrix ($n \times n$). Sparse portfolios are simply obtained by applying a portfolio optimization method (see next subsection) with a sparse inverse covariance instead of a full covariance as input.

A final key element of the ICC methodology is the temporal consistency of the

cluster that is imposed by penalizing frequent switches between clusters. In this chapter, the penalizer parameter γ is estimated in the train set through a grid search so that the average cluster persistence is of a given length (30 days in this chapter).

The assignment of the temporal instance t to a cluster number, k_t, is performed iteratively starting from an initial random cluster assignment. Specifically, we evaluate the penalized gain

$$\tilde{G}_{t,k_t} = G_{t,k_t} - \gamma \delta_{k_{t-1},k_t}, \tag{3.5}$$

and assign observation t to the cluster with the largest penalized gain. In the previous expression, δ_{k_{t-1},k_t} is the Kronecker delta returning one if $k_{t-1} = k_t$ and zero otherwise. After the assignment of the time-t observation to a given cluster k_t, all cluster parameters (means and covariances) are recomputed with the new cluster assignments.

We then performed a mean-variance portfolio optimization method independently for each ICC state. Obtaining optimal weights associated with each temporal cluster. To apply effectively such optimized weights to the portfolio problem we have to forecast the state that is most likely to be predominant in the future test set where the investment is performed. For this purpose, we made use of the short-term persistence of such states and we assigned as most likely future state the one that is predominant in the last part of the train set. In this chapter, we consider two clusters only.

3.2.2 Portfolio optimization methods

Our proposed methodology is made of three main stages. First, we use ICC for the temporal clustering of the training dataset into two market states. Second, we forecast which of the two states will be predominant in the future, test dataset, where the investment is made. Third, we perform portfolio optimization using training data from the forecasted predominant ICC state. Our approach is, to a large extent, agnostic to the kind of optimization adopted. In this chapter, for the experiments, we used two, mean-variance optimization methods: 1. the Sequential Least Squares Quadratic Programming approach and; 2. the Critical Line Algorithm method. Let

us briefly recall the basic elements of these two portfolio optimization methods.

For the experiments in this chapter Markowitz's optimal weights can be computed with the Python package 'Numpy' for direct matrix multiplication. The exact solution is shown in Appendix A.1.5. In the literature, this solution is referred to as 'unconstrained' because, besides the normalization and average conditions, the weights have no other constraints. On the other hand, in some practical cases, one might want to add further conditions to the weights. For instance, many real-world situations do not allow short selling, which hence makes it necessary to impose only positive weights in the range $w_i \in [0, 1]$. This constrained optimization problem is expensive to be solved analytically and thus numerical optimization methods must be adopted.

Two numerical optimization methods have been adopted in the experiment. The **sequential least square quadratic programming (SLS)** [185, 186, 187] is considered to be one of the most efficient computational methods to solve general nonlinear constrained optimization problems. Jackson et al. and Cesarone et al. demonstrate its effectiveness in finance [188, 189]. There is an easy-to-use package implemented in Python's **SciPy.optimize** library which we applied in our experiments. The **Critical Line Algorithm (CLA)** is an efficient alternative to the quadratic optimizer for the mean-variance model, as it is specifically designed for inequality portfolio optimization. It was already originally introduced in the Markowitz Portfolio Selection paper [1], and its computational implementation has become increasingly popular [190, 191]. CLA also solves constrained problems with conditions in inequalities, but unlike SLS, it divides a constrained problem into a series of unconstrained sub-problems. In our experiment, to compute CLA optimization for portfolio selection, we leveraged the implementation from the open-source **portfoliolab** Python library from Hudson and Thames [192]. A key drawback of CLA is called the Curse of Markowtitz, which is that a small change can lead to a very unstable inverse covariance matrix calculation. Our employment of sparse inverse covariance matrices via an information filtering network produces more robust results that are more resilient to the noise produced by small

changes, and the overall model delivers better performances, in terms of Sharpe ratio, annualized returns and volatility, with respect to the model with full inverse covariance. Mathematical and algorithmic details of SLS and CLA are included in Appendix A.1.5 for reference.

In summary, these two portfolio optimization methodologies output optimal portfolio weights $\mathbf{W}$ from an input constituted of: (i) a set of observations $\mathbf{r}_t$; (ii) a vector of mean returns μ; (iii) a covariance Σ. As we shall see shortly, in our implementation these inputs are provided in various combinations including selecting from ICC states and sparsifying.

3.3 Implementation

3.3.1 Data

We carried out several experiments using historical financial time-series data from three major capital markets: NASDAQ, FTSE and HS300. We selected 100 stocks from each of these three markets during the trading period between 01/01/2010 and 01/01/2020. For each stock, we calculated the daily log-return, $r_i(t) = \log(P_i(t)) - \log(P_i(t-1))$, using closing prices P_i of stock i at day t. For the 100 stocks, in the main chapter, we selected the largest market capitalization constituents but in the appendix, we repeat the experiments with random selection obtaining comparable results.

3.3.2 Experiments

The optimal portfolio weights are obtained from the data in the train set and performances are measured over the test set where portfolio weights are left unchanged. As performance indicators, we compute portfolio return, portfolio standard deviation (i.e. volatility) and Sharpe ratio over the investment horizon (test period). We report the annualized value of these quantities, estimated as the daily values multiplied by $\sqrt{252}$. For statistical robustness, for each market, we compute the above portfolio performance indicators over 100 randomly chosen consecutive train-test periods within the ten years dataset. Results are reported for the mean performances and the 5%-95% quantile ranges over such re-sampling.

The test set length (investment horizon) was established at 30 days which is a reasonable value for practical applications, we however also report in the appendix results for horizons of 10, 20 and 100 days found consistent results. The train set length was established by performing experiments with train sets of $L = 0.5, 1, 2, 3, 4$ years. Figure 3.1 reports the annualized average Sharpe Ratio computed on the test set as a function of the train set length. One can observe from the top figure that the lengths between one and two years yield consistently good performances. We adopted the period of 2 years as the optimal compromise between statistical robustness and best performances.

In the experiments, we first compute, on the training set, the ICC time clusters assuming two states and Student-t log-likelihood, Eq.3.4 as gain function. The choice of two states is for the sake of simplicity, we tested also 3 states obtaining inferior but comparable results. We verified that Student-t likelihood is best performing among the tested gain functions, in Appendix we report results also for Normal log-likelihoods (Eq.3.2). The switching penalty parameter γ in ICC was set so that the average cluster size is around 30 days, i.e. consistent with the 30-day investment horizon. This selection of average cluster size and investment horizon is a somehow arbitrary choice based on the effective threshold of the portfolio performance measured by the Sharpe Ratio. We then labelled 'Sparse 0' the state that is most abundant among the last 20 days of observations at the end of the train period. Conversely, we labelled 'Sparse 1' the other state. The term Sparse is used to indicate that this portfolio uses sparse inverse covariance. To set such a 'prevalence period' of 20 days we first performed a grid search over the combination of training duration $L = 0.5, 1, 2, 3, 4$ years, and using prevalence periods of $10, 20, 30, 100, L/2, L$ days. This search confirmed that small values of prevalence periods, of $10, 20, 30$ days, provide better results than larger prevalence periods. We therefore set a prevalence period of 20 days as it provides the most consistent results across the grid search and it is also consistent with the length of the test set.

The bottom plot in Figure 3.1 reveals that Sparse 0 has consistently better performances over Full, best results for training periods of one year. Let us notice

that, having an ICC average cluster size of 30 days, it makes it hard to cluster well in a small training period of six months, and often unbalanced clusters where one cluster dominates the period are obtained. On the contrary, a large training duration (4 years) makes the model prone to unnecessary patterns and noise, and in turn, reduces performance. Thus we chose 1 year as the best compromise for the length of the training set. In Appendix A.1.1 and A.1.3 we see that similar results are obtained for the other two markets (FTSE, HS300).

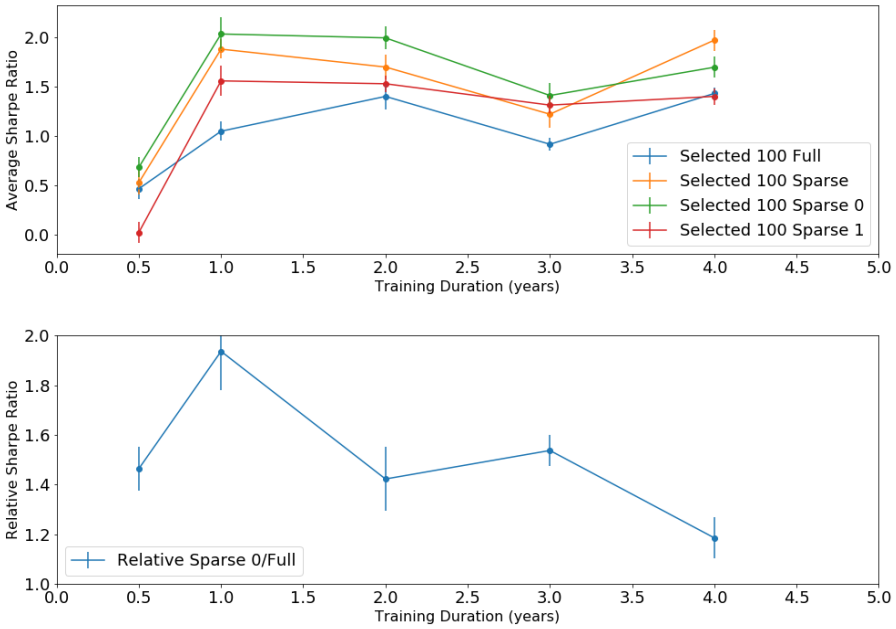


Figure 3.1: Sharpe Ratio for portfolios with 100 largest market capitalization constituent stocks of NASDAQ Composite optimized using different training set durations. The top subplot reports the average Sharpe Ratios (SR) with error bars reporting 1 standard deviation, for Full, Sparse, Sparse 0 and 1, statistics on 100 training-testing periods chosen at random within the 10 years dataset. The bottom subplot report instead the relative Sharpe Ratios between Sparse 0 and Full, $SR_{Sparse0}/SR_{Full}$.

We compute optimal portfolios using the two (SLS and CLA) optimization methods. We trained each optimization method both on the whole train dataset and, separately, on the two Sparse 0 and Sparse 1 states. We used the sample means for each of the respective sets and either the 'full' sample covariances (Pearson's estimate) or the 'sparse' sample covariances (TMFG-LoGo estimate [23, 17]). There-

fore, for each optimization method we have four optimized portfolios: two computed on the whole training set and with full or sparse inverse covariance (named 'Full' and 'Sparse' respectively); two computed on the two ICC market states and with sparse inverse covariance (named 'Sparse 0' and 'Sparse 1'). For benchmarking, these portfolios are also compared to a portfolio with equal weights, $w_i = 1/n$ named 'Naive'. Overall, we have therefore 4×2 plus 1 differently optimized portfolios that are recomputed 100 times over randomly sampled time periods. Such optimized portfolio weights are applied, for each of the three markets, to the 100 most capitalized stocks. In the appendix, we repeat the experiments for randomly selected stocks.

3.4 Results

3.4.1 Log-likelihood

We computed the daily Student-t log-likelihood, using Eq.(3.4), for each of the 30-day investment horizons. Figure 3.2, reports the averages of the differences for each day between the log-likelihood of Sparse 0 and full and also between Sparse 1 and full. The average is taken over the 100 random re-samplings.

Figure 3.2, shows mostly positive gains for Sparse 0 indicating that, for most days across the investment horizon, it has larger log-likelihoods than Full. Sparse 1 gain instead reveals mostly negative results against full. This therefore indicates that while Sparse 0 is, on average, a better model to describe the multivariate nature of the log returns in the test set with respect to Full; instead, Sparse 1 is in average worst. Since both Sparse 0 and 1 were sparsified using TMFG, the difference between them must therefore be a consequence of clustering. One might note that, even though some Sparse 1 log-likelihood gains are in the positive domain, they anyway have smaller magnitudes than their Sparse 0 counterparts. This result clearly shows the effectiveness and importance of considering market states.

Let us note that the two ICC clusters gather together observations that maximize in-sample log-likelihood in the respective clustered periods. The fact that these models (i.e. in-sample means and covariance) from these clusters still corre-

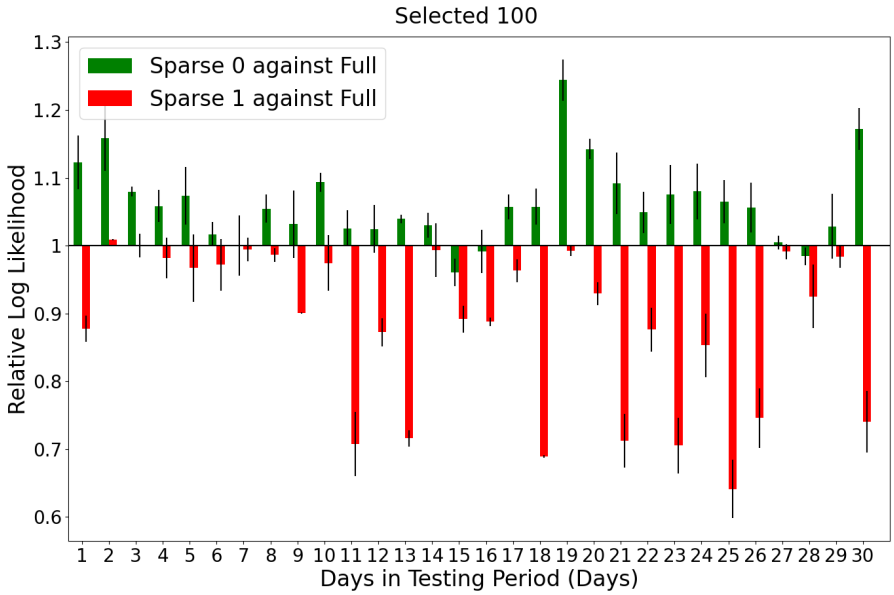


Figure 3.2: Student-t log-likelihood for 100 largest market capitalization constituent stocks of NASDAQ Composite v.s. number of days in the test period after training. Each bar represents the average gain of the Sparse 0 (green) or 1 (red) with respect to the Full in each day. Averages are over 100 re-samplings.

spond to different log-likelihood performances in the off-sample test set indicates that the states are still relevant off-sample. Further, the better off-sample performances of the Sparse 0 state indicate predictability; i.e. if one state outperforms another during the training period, it will remain better performing throughout the test period. Furthermore, to illustrate the universality of the log-likelihood results, we have included similar Student-t log-likelihood (Appendix A.1.1) as well as Normal log-likelihood (Appendix A.1.3) plots for random 100 stocks selections for all three major indices (NASDAQ, FTSE, HS300), where similar patterns are observed.

3.4.2 Portfolio Performance

We tested portfolio performances over a 30-day investment horizon for the 100 largest market capitalization constituent stocks of NASDAQ Composite computed with the four portfolio optimization methods, SLS and CLA and using as inputs Full, Sparse, Sparse 0 and Sparse 1. We also report the $1/n$ Naive construction as the benchmark. Tables 3.1 and 3.2 report portfolio performances for the combination of portfolio constructions (column 'Solver') and inputs (column 'State'). Per-

Solver	State	Return (%)	(5,95)th percentile	Volatility (%)	(5,95)th percentile	Sharpe	(5,95)th percentile
	$1/n$ Naive	14.46	(-36,55)	17.4	(14,28)	1.536	(-1.4,4.3)
SLS	Full	22.71	(-28,96)	19.5	(14,28)	1.627	(-1.4,5.4)
SLS	Sparse	21.81	(-23,74)	17.5	(14,26)	1.764	(-1.0,6.2)
SLS	Sparse 0	**29.04**	(**-6**,66)	**16.0**	(12,**25**)	**2.478*****	(-0.3,6.9)
SLS	Sparse 1	5.35	(-49,57)	19.8	(14,34)	0.978	(-2.3,4.6)
CLA	Full	21.97	(-69,97)	19.5	(14,31)	1.541	(-2.1,6.5)
CLA	Sparse	22.27	(-32,85)	17.0	(12,27)	1.758	(-1.9,6.5)
CLA	Sparse 0	**28.73**	(**-27**,76)	**15.8**	(11,**26**)	**2.372*****	(-1.5,7.6)
CLA	Sparse 1	12.48	(-57,86)	18.7	(12,32)	0.964	(-2.9,6.6)

Table 3.1: Portfolio performances were obtained by using Student-t log-likelihood for ICC clustering. We report annualized return, annualized volatility and annualized Sharpe Ratio computed on 30 30-day investment period after the 1 year training set. The values are averages and 5th and 95th percentiles computed over a 30-day investment horizon obtained from 100 re-sampling of consecutive training-investment periods chosen at random within the 10-year dataset. The underlying assets are 100 largest market capitalization constituent stocks of the NASDAQ Composite. Highlight in bold are return, volatility and Sharpe Ratio indicating the optimal state in each market solver combination, while highlights in the 5th return and 95th volatility showcase the extreme behaviours (excluding the state Market). The state $1/n$ Naive is the equally weighted un-optimised portfolio and it is reported as a benchmark. A pairwise T-test has been performed, and the p-values for the best-performing results against the second best-performing results in Sharpe Ratio are highlighted next to the best-performing Sharpe Ratio, where * denotes 5% significance, ** for 1% and *** for 0.1% respectively.

formances are quantified in terms of annualized portfolio return, annualized portfolio standard deviation (volatility) and the annualized Sharpe ratio over a 30-day investment horizon. We report the 5% and 95% quantiles and the means computed from the 100 random resampling of consecutive training-investment periods chosen at random within the 10-year dataset. The maximum returns and Sharpe Ratios, as well as the minimum volatility, are highlighted in bold. Thus showing the best performer in each market-solver combination. In addition, we highlight the minimum 5th percentile returns to depict the state suffering the least loss, and the maximum 95th percentile volatility to showcase the most stable state in extreme market situations.

From the mean values reported in Tables 3.1 (Student-t log-likelihood), we observe that Sparse 0 outperforms Full, and this supremacy dominates for the two solvers. More specifically, Sparse 0 is on average 29.3%, 19.5% and 53.1% better

Solver	State	Return (%)	(5,95)th percentile	Volatility (%)	(5,95)th percentile	Sharpe	(5,95)th percentile
	$1/n$ Naive	14.46	(-36,55)	17.4	(14,28)	1.536	(-1.4,4.3)
SLS	Full	22.98	(-28,96)	19.3	(14,28)	1.667	(-1.4,5.4)
SLS	Sparse	21.96	(-23,74)	17.3	(14,26)	1.787	(-1.0,6.2)
SLS	Sparse 0	**29.00**	(**-14**,66)	**15.9**	(12,**23**)	**2.260*****	(-0.8,4.6)
SLS	Sparse 1	6.84	(-43,63)	19.5	(14,30)	0.845	(-1.8,4.6)
CLA	Full	20.63	(-76,97)	19.5	(14,31)	1.456	(-3.0,6.5)
CLA	Sparse	21.15	(-53,85)	17.0	(12,**27**)	1.678	(-2.1,6.5)
CLA	Sparse 0	**27.08**	(**-14**,79)	**15.6**	(10,30)	**2.175*****	(-0.7,6.6)
CLA	Sparse 1	11.54	(-69,77)	18.6	(14,36)	1.028	(-2.6,5.6)

Table 3.2: Portfolio performances obtained by using Normal log-likelihood for ICC clustering. We report annualized return, annualized volatility and annualized Sharpe Ratio computed on 30 days investment period after the 1 year training set. The values are averages and 5th and 95th percentiles computed over 30-day investment horizon from obtained from 100 re-sampling of consecutive training-investment periods chosen at random within the 10 years dataset. The underlying assets are 100 largest market capitalization constituent stocks of NASDAQ Composite. Highlight in bold are return, volatility and Sharpe Ratio indicating the optimal state in each market solver combination, while highlights in 5th return and 95th volatility showcase the extreme behaviours (excluding the state Market). The state $1/n$ Naive is the equally weighted un-optimised portfolio and it is reported as benchmark. A pairwise T-test has been performed, and the p-values for the best-performing results against the second best-performing results in Sharpe Ratio are highlighted next to the best-performing Sharpe Ratio, where * denotes 5% significance, ** for 1% and *** for 0.1% respectively.

in return, volatility and Sharpe Ratio than Full across all two solvers. We observe instead that Sparse 1 is considerably worst than Full indicating therefore that the significant gain of State 0 comes from filtering out the 'disadvantageous' Sparse 1 state rather than sparsification. This is indeed confirmed by the small observed gains of Sparse over Full. these results are confirmed by the analysis of the 5th and 95th quantiles where we notice that Sparse 0 consistently achieves the least minimum extreme loss and the least maximum extreme risk. Specifically, Sparse 0 on average loses 66.0% less and is 13.6% less volatile than Full on 5th percentile return and 95th percentile volatility respectively. In other words, the integrated clustering portfolio optimization algorithm, ICC-PO, that we proposed can boost returns with less risk than the traditional benchmark, as well as provide extra resilience in extreme market situations.

To test the sensitivity of this method to the specific ICC clustering gain func-

tion, we performed the same analysis using the normal log-likelihood gain function for ICC clustering. Results are reported in Table 3.2. Consistently with the previous results, we observe 28.6%, 18.8% and 42.0% improvements in return, volatility and Sharpe Ratio, with 73.1% and 10.2% gains in 5th percentile return and 95th percentile volatility. The comparison illustrates a 11.1% Sharpe Ratio improvement in Student-t log-likelihood and a 7.1% 5th percentile return advance in Normal log-likelihood. In other words, Student-t is a better model for the market and boosts portfolio performance. However, Normal log-likelihood generates a higher resilience to extreme loss. Since the average gain in return and volatility are similar in the two cases, the performance difference should mainly come from general upward-shifted ranges in the Student-t Sharpe Ratio.

Similar tables of optimization results using 10, 20, 30 and 100-day investment horizons, can be found in Appendix A.1.2 for Student-t log-likelihood and A.1.4 for Normal log-likelihood. These experiments were carried out over randomly selected 100 stocks baskets (instead of the 100 most capitalized ones); the set of 100 random stocks was re-chosen for each of the 100 re-sampling. Most of the general patterns found earlier still hold regardless of the length of the testing period and underlying assets. The relative difference, namely, the gain between Sparse 0 and Full remains roughly the same. This consistency further confirms the generality of our ICC-PO model. In this case, we report only the percentiles of the performance measures because being re-sampled on different constituents, mean values might be misleading.

3.5 Discussion

The results presented in Section 3.4.1 quantitatively demonstrate an effective gain in log-likelihood after applying temporal ICC clustering and computing the optimized sparse portfolio associated with the most persistent ICC cluster in the last 20 days of training (the Sparse 0 portfolio). We highlighted that the additional gain in the ICC-PO construction is mainly a consequence of the market states clustering and only partially a consequence of sparsification. These results are extremely

robust showing comparable patterns across experiments conducted for three major capital markets, using two different solving methodologies, adopting four investment horizons and using both Student-t and normal log-likelihoods gain functions (Appendices A.1.2 and A.1.4). The results in Appendices A.1.2 and A.1.4 obtained for 100 random stocks in the US, the UK and the Chinese markets show a broader variability but overall well-aligned results. Our results also confirm the observation, by Procacci and Aste [184], that models with larger likelihood better solve the portfolio optimization problem.

As for the analysis of the 100 NASDAQ's most capitalized stocks, also for the random selection and the three markets, we observe that the Normal log-likelihood, Sparse 0 is 33.8% less than Full in the 5th percentile Return, whereas the Student-t log-likelihood is only 20.4%, which illustrates that the Normal statistically loses less money in extreme situations. Namely, it results that there are general advantages in using Student-t over Normal log-likelihood, yet, the latter performs better at limiting risks. The edge in three main performance matrices depicts the Student-t's better market modelling property as suggested in the literature, especially for limited sample daily log-return. In contrast, the mere pitfall in risk measures may probably come from the fat-tail nature of the Student-t distribution.

It is difficult to assess the efficiency of ICC-PO by direct comparison to the literature since our focused result is the relative difference between Sparse 0 from Full. The most informative measurements of general performance used widely in the field of portfolio management are Sharpe Ratio (the risk-adjusted return), Jensen's Alpha (the abnormal return over the theoretical expectation), Treynor Ratio (the risk-adjusted excess return from a risk-free asset) and Roy Ratio (the risk-adjusted excess return from the market index) [193]. Literature identifies that the Sharpe Ratio at values around 1 is commonly considered as the boundary between a good and bad investment strategy, while the Sharpe Ratio at values around 2 represents an excellent standard, and 3 and above are more likely to be achieved in a High-Frequency Trading (HFT) strategy [194, 195]. During the 10-year we investigated the annualized Sharpe Ratio for NASDAQ-100, FTSE-250 and HS300 have been

respectively equal to 1.77, 0.42 and 1.07 [196, 197, 198]. While our results for the various portfolio construction combinations generally lie in a reasonable range around these values, we note that the average Sharpe Ratio of the Sparse 0 based on the 100 largest market capitalization stocks from NASDAQ is 2.425, as well as 100 random stocks from NASDAQ is 2.132, from FTSE is 1.682 and from HS300 is 1.814 greatly exceeding the index's performances.

Apart from the Sharpe Ratio for general performance assessment, the risk is often a critical consideration in portfolio investment due to the risk aversion nature of investors and the quadratic utility function assumption. Two widely used risk measures value at risk (VaR) [199] and probable maximum loss (PML) [200], are interpreted as the minimum and the maximum loss expected in a portfolio over a time period. As a proxy combination of VaR and PML, we reported the 5th percentile Return in the random re-sampling. The observed general 66.0%, 49.8%, 21.6% and 32.4% reductions in loss respectively for largest-market-capitalization NASDAQ, NASDAQ, FTSE and HS300 are highly significant results indicating likely large improvements of both VaR and PML.

Lastly, as ICC-PO is computationally very efficient, it can be easily re-run for every allocation window making dynamic portfolio allocation easy.

3.6 Summary

In this chapter, we have successfully demonstrated the use of sparse networks in multivariate time-series clustering and its application in portfolio optimization. This clustering method is efficient and effective as the underlying sparse modelling of a multivariate time-series network sufficiently improves the signal-to-noise ratio.

Portfolio optimization lies at the core of quantitative investment. Automation in the dynamic allocation process is a challenging goal with a large community of academics and practitioners dedicated to this task which requires a precise and accurate modelling of the past market performance and a predictive inference of the future market state. However, it is never an easy task to predict the future, not to mention doing so constantly. Explanatory as they are, only certain signals pos-

sess the forecasting ability and normally only for a limited period of time. Hence, the results of our proposed algorithm ICC-PO are worth to be mentioned. Indeed, we improve the equal weight benchmark by over 50% in Sharpe Ratio, obtaining a statistically more robust and resilient investment performance, especially in the extreme market situations with large reductions in losses.

In this chapter, we demonstrated that markets can be classified in different states with distinct statistical properties. By using two states, classified and clustered using log-likelihood as gain function and sparse inverse covariance estimation, we have shown that the two clustered states continue to be distinguishable in log-likelihood after the train (in-sample) period, with one having systematically larger log-likelihood than the one computed from the whole, unclustered, training sample. We have shown that the state with a larger log-likelihood tends to be the one with the largest likelihood in the last period of training, indicating temporal persistence and providing a way for predictability of the best-performing state in the off-sample investment period. Portfolios optimized with data from the best-performing state's cluster give significantly better results than portfolios constructed from the full dataset or the other state. This also confirms the intuitive argument (see [184]) that a model with a larger likelihood must perform better for portfolio optimization purposes than a model with a lower likelihood. These results were tested extensively across a period of ten years, across three different markets, with portfolios from two different optimizers, with clustering from two different log-likelihoods, and both by using a selected group of most capitalized stocks as well as by randomly picking a stock basket.

The choice of using two market states has been dictated by simplicity. Future work will investigate the effect of the number of ICC clusters on the results. Our results are based on a naive selection of stocks from major indices. Hence, with a carefully designed portfolio basket, as commonly done in industrial practices, we expect further improvement of the results. Also, a wider application in asset classes is a straightforward extension of the method.

Chapter 4

Sparse Multivariate Time-series Network for Portfolio Optimization

4.1 Introduction

In the preceding chapter, we substantiated the utilization of a sparse network for modelling multivariate time series, resulting in considerable noise reduction. Nonetheless, the filtration procedure involved in sparse network formation frequently introduces noise due to the imposed graphical constraints, e.g., no-cycle condition must be fulfilled for MST, despite including a cycle might further reduce the sum of the weight of edges. In the present chapter, we introduce an innovative bootstrapping framework specifically designed to amplify the signal-to-noise ratio and provide an improved representation of interactions within multivariate time series. Once more, we apply this proposed method in the realm of portfolio optimization to showcase its potency.

The optimization of financial portfolios has long been a focal point of investigation within the domains of finance and quantitative trading. The first mathematical formulation of the problem follows the seminal works of Markowitz in the 1950s [201, 1]. An optimal portfolio that minimizes a risk matrix (e.g., variance) for a given expected return is to be found by solving a quadratic optimization problem under linear constraint, and the closed-form solutions form the efficient frontier. The minimum variance portfolio (MVP) lies on the efficient frontier line minimiz-

ing the variance, and it is widely regarded and practised by academia and industry as the most classic solution to the optimization problem. The MVP solution is simple and elegant that is contingent solely on the covariance of assets' historical return, independent of the mean. The covariance captures the volatility of a single asset and the interdependencies (correlation) between them. However, empirical covariance is notably unstable, particularly within multivariate financial time series where the signal-to-noise ratio is exceptionally low. Consequently, minor perturbations can trigger significant deviations. Additionally, financial markets are frequently subject to shifts and jumps, rendering the historical empirical covariance an unsatisfactory predictor of future trends, and the prediction of future covariance a challenging task. These factors unfortunately result in the mathematical optimality of the MVP failing to persist in off-sample periods.

Recent progress in the realm of network science, especially in network filtering, has provided alternatives to the traditional covariance-based methodologies. The covariance and correlation matrices can be interpreted as a graph/network, and can be condensed to essential information under certain graphical constraints, such as the minimum spanning tree (MST). The resulting filtered matrix typically exhibits sparsity with many structural zeroes, which correspond to statistically insignificant network components, thereby enhancing the matrix's robustness and generality. The resulting sparse network has been demonstrated to be useful for visualization and to aptly reflect market dynamics [202, 203]. Consequently, investment decisions, including portfolio selection and optimization, can be based on this network. For instance, the positive-defined sparse inverse covariance matrix from Triangulated Maximally Filtered Graph (TMFG) can directly substitute the original empirical inverse covariance matrix in the Markowitz model, yielding substantial improvements [204, 205]. Other topological information, such as centrality and peripherality, and community clusters, can be used as criteria for stock selection and portfolio weight optimization.

Information Filtering Networks (IFN) represent a robust and computationally efficient network filtering technique. However, due to certain topological con-

straints necessary for network construction, superfluous edges need to be introduced, which leads to slightly increasing the amount of noise. This chapter presents a novel method, the Statistically Robust Information Filtering Network (SR-IFN), which employs a statistically robust bootstrapping approach to mitigate the noise introduced during the IFN's building pipeline. In this method, the underlying multivariate time series is bootstrapped multiple times and transformed into sparse sub-networks. These sub-networks are then ensembled, and only the key structures that occur more frequently than a predefined threshold are retained. This strategy prunes unnecessary components, increasing the informativeness of the remaining edges and maximising the likelihood of the modelled system. This enhanced sparse network is then used for portfolio selection based on connectivity, as a peripheral portfolio is more diverse and entails lower risk. Further optimization can be carried out using the centrality of assets as a measure for weight calculation. We conduct experiments utilizing the component stocks of NASDAQ, FTSE, and HS300, representative of the equity markets in the US, UK, and China, respectively, and we include both scenarios with and without 20 basis point transaction costs.

4.2 Methodologies

4.2.1 Statistically Robust IFN (SR-IFN)

The application of Information Filtering Networks (IFNs) has been extensively explored within the field of finance, particularly for the purpose of correlation/covariance sparsification and filtering. Nevertheless, given that IFNs specify a complete network/graph under certain topological constraints, such as planarity for PMFG and chordality for TMFG, the resultant network structure incorporates elements that, while necessary to uphold these constraints, are irrelevant to the original information. This chapter introduces a Statistically Robust (SR) method aimed at enhancing the stability and performance of IFNs by endeavouring to eliminate these constraint-related structures. Triangulated Maximally Filtered Graph (TMFG) is employed as the core IFN for the purposes of the ensuing experiments.

The construction process for TMFG is outlined in Algorithm 1. It relies on

Algorithm 2 Statistically Robust IFN (SR-IFN)

Input A set of observations $\hat{\mathbf{x}}_{s,n} \in \mathbf{R}^{s,n}$, the confidence level p_{cl}, and the number of repetitions t_r

Output Sparse similarity matrix $\mathbf{S}$.

1: Initialize an empty ensemble adjacency matrix $\mathbf{A} \in \mathbf{R}^{n,n}$ with all zeros;
2: Initialize an empty final sparse similarity matrix $\mathbf{S} \in \mathbf{R}^{n,n}$ with all zeros;
3: Calculate the original correlation matrix $\hat{\mathbf{C}} \in \mathbf{R}^{n,n}$ from $\hat{\mathbf{x}}_{s,n}$;
4: **for** $t \leftarrow 1$ to t_r **do**
5: Randomly bootstrap $\hat{\mathbf{x}}_{s,n}$ in the first dimension and obtain bootstrapped $\hat{\mathbf{x}}^t_{s,n}$;
6: Calculate the bootstrapped correlation $\hat{\mathbf{C}}^t \in \mathbf{R}^{n,n}$ from $\hat{\mathbf{x}}^t_{s,n}$;
7: Obtain the bootstrapped sparse adjacency matrix $\mathbf{A}^t$ from $\hat{\mathbf{C}}^t$ by TMFG in Algorithm 1;
8: $\mathbf{A} += \mathbf{A}^t$
9: **end for**
10: **for** each pair of nodes i, j in $\mathbf{A}$ **do**
11: **if** $\frac{A_{i,j}}{t_r} > p_{cl}$ **then**
12: $S_{i,j} = \hat{C}_{i,j}$;
13: **end if**
14: **end for**
15: **return** $\mathbf{S}$.

a simple topological move that maintains both planarity and chordality. TMFG has been demonstrated to be a computationally efficient model capable of generating sparse probabilistic modelling via topological regularization. However, it is not without limitations: unnecessary edges may be added to satisfy the graph's chordality, thereby introducing undesirable noise, a particular issue in fields characterized by a low signal-to-noise ratio, such as finance. To address this limitation, we propose the Statistically Robust (SR) method, detailed in Algorithm 2.

Temporal sequential dependence is reduced by randomly bootstrapping the observations in each repetition, which also results in each bootstrapped sample possessing a distinct network structure. Therefore, superfluous edges will be added differently in each case. By retaining structures that appear more frequently than a certain threshold, we can discard unnecessary edges and noise as they lack statistical robustness and tend to occur randomly, hence infrequently, see Figure 4.1. Algorithm 2 illustrates this process, introducing an ensemble adjacency matrix $\mathbf{A}$, which amalgamates all adjacency matrices from bootstrapped sub-TMFGs. After

all repetitions, the occurrence probability of an edge between any pair is calculated, and only if this probability exceeds the defined confidence level (ConfLv) threshold, do we retain the edge. The final output **S** is a similarity matrix representing the original correlation, but with many structural zeros from the discarded edges to ensure sparsity.

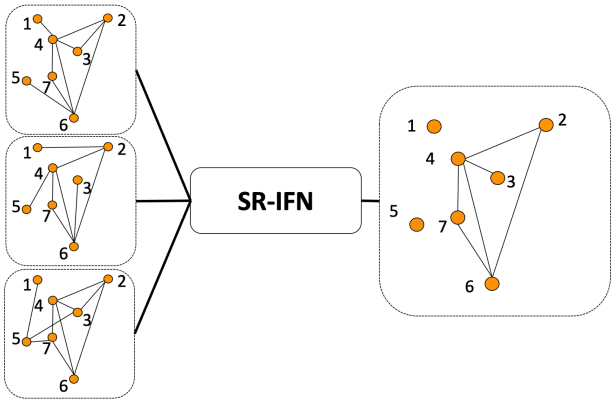


Figure 4.1: Statistically robust bootstrapping process. Three sub-networks were generated from one observation set with bootstrapping. Only edges that present more than a two-thirds majority will be preserved in the resulting statistically robust network.

4.2.2 Bootstrapped Centrality Measures

Subsequent experiments will utilize three centrality measures for portfolio weight calculation, including Degree Centrality, Communicability Betweenness Centrality, and Absolute Correlation.

Degree Centrality is one of the simplest and most common centrality measures used to quantify the prominence of a node in a network. It is based on the idea that nodes with more connections (edges) to other nodes are more central and influential within the network. For an undirected network, the degree centrality, c_i^d of a node i is calculated as the number of edges (connections) it has denoted by k_i. The normalized degree centrality is obtained by dividing k_i by the maximum possible number of connections, which is $(n-1)$, where n is the total number of nodes in the network,

$$c_i^d = k_i/(n-1). \tag{4.1}$$

This normalization allows for the comparison of centrality scores across different

networks.

Communicability Betweenness Centrality (CBC) is an extension of the traditional betweenness centrality measure, which is based solely on the shortest paths between nodes. While betweenness centrality focuses on the number of shortest paths that pass through a given node, CBC takes into account the weighted sum of all paths between nodes, where the weight of each path is inversely proportional to its length. Mathematically, communicability between nodes i and j is calculated using the exponential of the adjacency matrix elements, $\mathbf{A}$, of the network. The adjacency matrix is a square matrix whose element $\mathbf{A}_{j,k}$ represents the connection between nodes i and j. The communicability between nodes i and j is given by the (j,k)-th element of the exponential of matrix element, denoted as $\exp(\mathbf{A}_{j,k})$. Communicability Betweenness Centrality is then calculated by summing the relative changes in communicability for all pairs of nodes when a node is removed from the network. For node i, the CBC is computed as:

$$c_i^{CBC} = \frac{\sum_{i \neq j \neq k} \exp(\mathbf{A}_{j,k}) - \exp(\mathbf{A}_{j,k} - E_i)}{\exp(\mathbf{A}_{j,k})} \tag{4.2}$$

where E_i is a matrix with the same dimensions as $\mathbf{A}_{j,k}$, representing the connections of node i (i.e., with 1s in the positions corresponding to the edges of node k and 0s elsewhere), and $(\mathbf{A}_{j,k} - E_i)$ represents a new adjacency matrix by removing node k from the network. In this formula, the summation is over all pairs of nodes i and j, excluding node k. The CBC quantifies the importance of node k by considering its role in facilitating communication between all pairs of nodes in the network, taking into account both direct and indirect paths.

The portfolio selection methods in the above section select assets with statistically significant decorrelation among the portfolio. Therefore an intuitive way for weights optimization is directly using the sum of absolute pairwise correlation, as the portfolio weight for each node/asset. Therefore, it is expressed as

$$c_i^{corr} = \sum_{j, i \neq j} |\rho_{i,j}|, \tag{4.3}$$

where $\rho_{i,j}$ represents the pairwise correlation between node i and node j.

A bootstrapping approach akin to Algorithm 2 is utilized for calculating statistically robust centrality. In each repetition, a sub-centrality, c^t, is determined within the sub-network obtained from bootstrapped observations, and the overall centrality is obtained by averaging all sub-centralities, as shown:

$$c_i = \frac{1}{t_r} \sum_{t=1}^{t_r} c_i^t \tag{4.4}$$

where c_i is the ensembled centrality, c_i^t is the sub-centrality for node i, and t_r is the number of repetitions.

4.2.3 Portfolio Selection and Optimization

In our application of the Statistically Robust Information Filtering Network (SR-IFN), we consider a total of N assets with T time-stamped historical observations. The resultant sparse similarity matrix, $\mathbf{S}$, represents the pairwise correlations between assets. Given the sparse nature of $\mathbf{S}$, it allows for the division of assets into two subsets: connected and disconnected. Disconnected assets lack any link to other assets, while connected ones possess at least one such link. By adjusting the confidence level (ConfLv) threshold within the SR-IFN, we can manipulate the quantities of disconnected and connected assets. For the purpose of establishing a portfolio with minimal correlation, we include all disconnected assets, while excluding the connected ones. More specifically, we select assets for which the sum of pairwise correlations is zero, as expressed in the following equation:

$$\sum_{j,i\neq j} \mathbf{S}_{i,j} = 0. \tag{4.5}$$

This results in the selection of assets that exhibit a very low statistical correlation with others within the portfolio.

To further enhance the portfolio, we optimize the weights such that they are

inversely proportional to the ensembled centrality measures,

$$w_i = \frac{1/(c_i + \varepsilon)}{\sum_j 1/(c_j + \varepsilon)} \tag{4.6}$$

where w_i is the weight, and c_i represents the centrality for asset i in the portfolio, ε is a regularization constant and $\varepsilon << 1$.

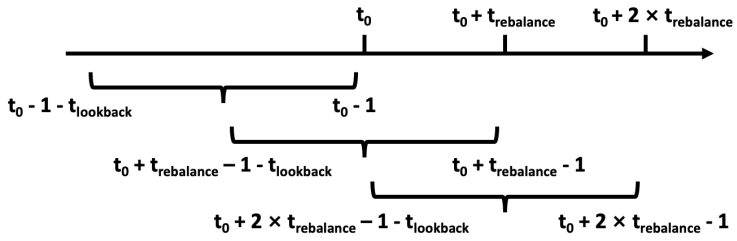


Figure 4.2: Portfolio weight is rolling rebalanced over the period. It is rebalanced every $t_{rebalance}$-day with a $t_{lookback}$ days look-back window of history. Therefore, the weights of the portfolio only change $t_{rebalance}$-day to avoid excessive trading and transaction costs. The calculation of new weights depends on the historical look-back window of $t_{lookback}$ days.

Dynamic allocation of the portfolio is achieved through rebalancing every $t_{rebalance}$-day. The selection criteria in Equation 4.5 and weights in Equation 4.6 are re-calculated based on a $t_{lookback}$ days look-back window of history, see Figure 4.2.3.

4.3 Implementation

4.3.1 Data

A series of experiments were conducted utilizing historical financial time-series data obtained from three principal capital markets: NSDAQ, FTSE, and HS300 (data obtained from Bloomberg), spanning the period from January 1, 2010, to January 1, 2020. For each constituent stock, the daily log-return, denoted as $r_i(t) = \log(P_i(t)) - \log(P_i(t-1))$, was computed using closing prices. Detailed statistics of the daily log-return distribution are furnished in Table 4.1 for subsequent comparison and discussion.

	NASDAQ	FTSE	HS300
Ann. Return	16.0%	5.8%	9.9%
Ann. Std.Dev.	17.5%	15.8%	23.2%
D. Skewness	-0.44	-0.95	-0.89
Max. Drawdown	-24.0%	-43.5%	-52.3%

Table 4.1: Statistics table for the log return distribution in NASDAQ, FTSE and HS300 between 01/01/2010 and 01/01/2020, including annualized return mean, annualized return standard deviation, daily return skewness, and maximum drawdown.

The chosen indexes are emblematic of distinctly divergent market dynamics during the designated period. NASDAQ was in a phase colloquially referred to as its 'golden ten years', characterized by a substantial annualized mean return and moderate volatility. The skewness of the return distribution is less negative compared to the other indexes, indicating fewer extreme loss events and consequently, a lower maximum drawdown throughout this period. In contrast, both FTSE and HS300 exhibited a high negative skewness and substantial drawdown over the same period. Additionally, the FTSE was more conservative, with a lower average return and volatility, whereas the HS300 displayed considerably higher volatility.

4.3.2 Experiment Setup

This section is devoted to the selection of portfolios exclusively from an index component stock pool. Consequently, the weights of the portfolio are maintained at $1/N$, where N represents the total number of assets in the chosen portfolio. The portfolio undergoes rebalancing every $t_{\text{rebalance}}$ days, with a historical look-back period of t_{lookback} days for the measurement of empirical correlation and other historical statistical properties. Experiments are included both with and without transaction costs of 20 basis points (bps) to simulate commission and bid-ask spread costs. The complete period is partitioned into in-sample and out-of-sample periods before and after 01/01/2017. A grid search over $t_{\text{rebalance}}$ and t_{lookback} is conducted in-sample for analysis and optimization, and the optimal parameters are retained for the out-of-sample period to showcase the persistence and significance of the method. In addition, to demonstrate robustness and present statistics, all experiments across the three markets are repeated and ensembled with varying starting dates.

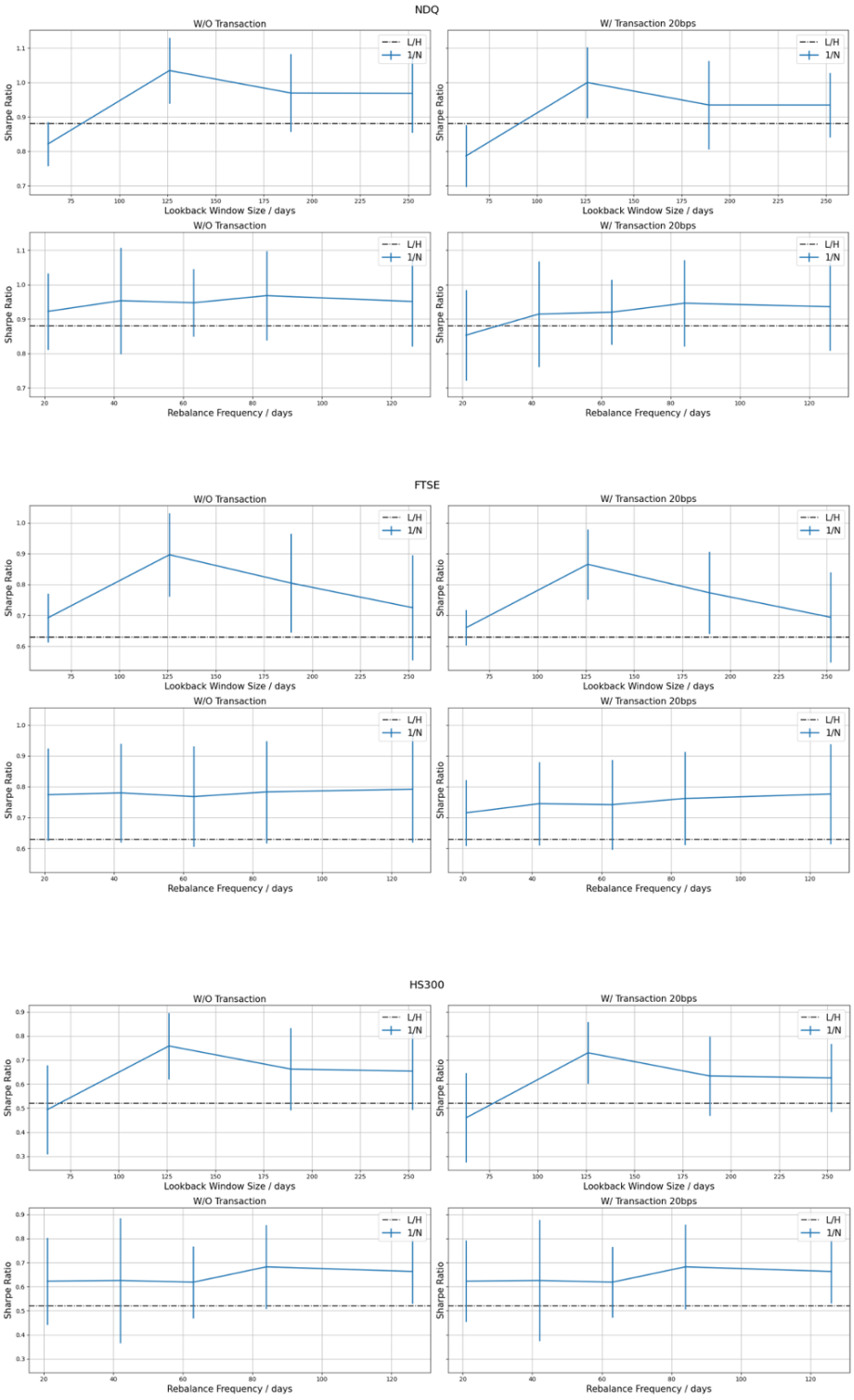


Figure 4.3: Grid search results in Sharpe Ratio across NASDAQ (top left), FTSE (top right) and HS300 (bottom) over various rebalance frequencies and lookback window sizes. The Left and right columns show results with and without 20bps transaction costs. L/H represents the long-hold portfolio over the entire stock pool, and 1/N represents the simple equally weighted selected portfolio averaged over other parameters. The grid search results of Confidence Levels of SR-IFN are not shown as it is not optimized for the remaining experiments.

Figure 4.3.2 portrays the grid search results over a range of rebalance frequencies, $t_{\text{rebalance}}$, and lookback window sizes, t_{lookback}. The results are averaged across varying $t_{\text{rebalance}}$, t_{lookback} and Confidence Levels of Statistically Robust Information Filtering Network (SR-IFN, denoted as ConfLv), but the outcomes in relation to ConfLv are not displayed as it is not optimized for the remaining experiments. This grid search is executed in-sample from 01/01/2010 to 01/01/2017. While it is safe to assume that t_{lookback} is optimally at 126 days for all three markets, there is a minor discrepancy among $t_{\text{rebalance}}$. Nonetheless, for simplicity and consistency, we employ an 84-day $t_{\text{rebalance}}$ for the remaining experiments.

4.4 Results

4.4.1 Topological Portfolio Selection

The Statistically Robust Information Filtering Network (SR-IFN), introduced in Section 4.2.1, provides a statistically robust selection predicated on historical correlation. The remaining correlated features, derived from the historical period, due to their robustness, are anticipated to maintain their correlation for a brief future period. This intrinsic ability to predict future correlation serves as a pivotal criterion for numerous portfolio selection and optimization techniques, as their primary objective is to identify the least correlated portfolio. In this section, we scrutinize the influence of the peripheral portfolio over the central portfolio, as defined by the correlation graph, and exhibit the supplemental gain from the SR-IFN peripheral portfolio in comparison to a classic correlation-based portfolio.

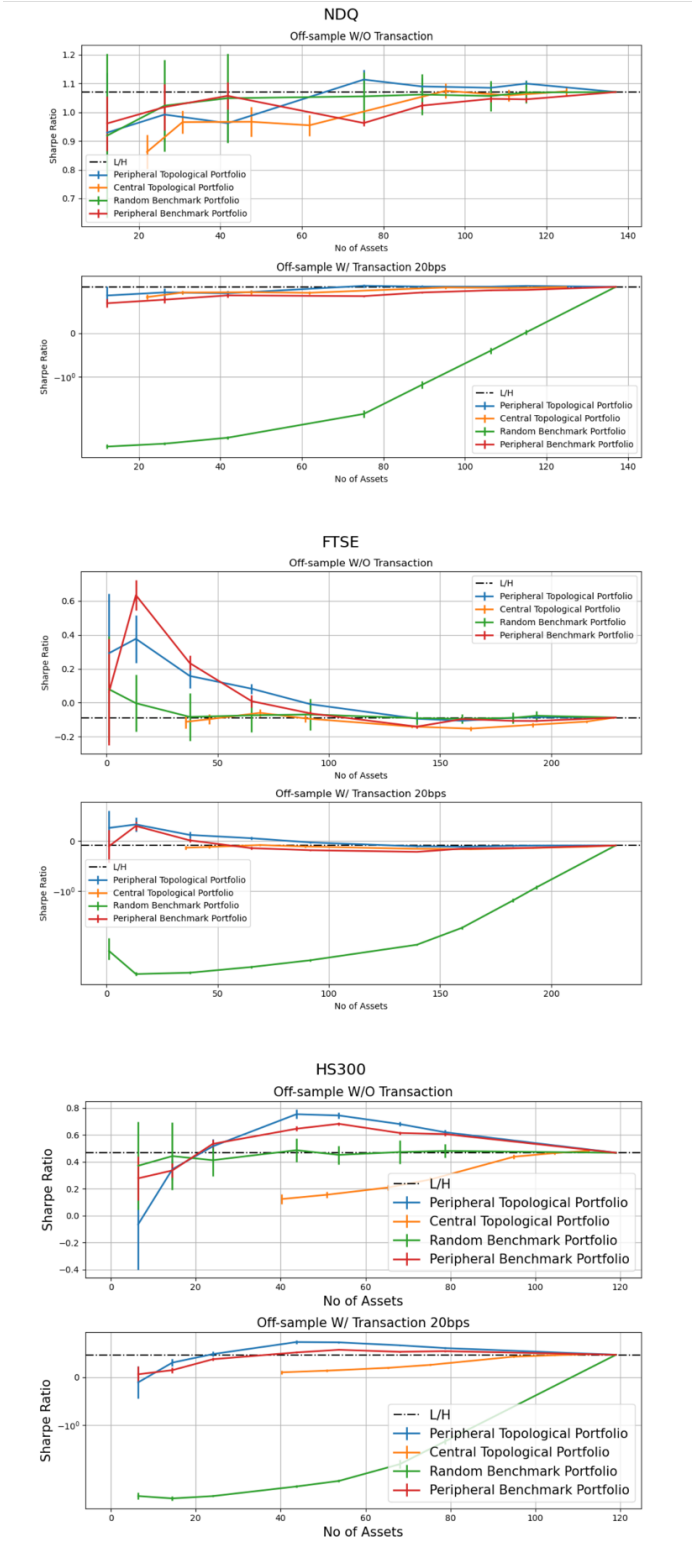


Figure 4.4: Portfolio selection results in Sharpe Ratio across NASDAQ (top left), FTSE (top right) and HS300 (bottom). The selection is performed based on the SR-IFN, and we report the central (orange) and peripheral (blue) portfolios as well as a random subsampled (green) portfolio to showcase the efficacy of the peripheral one. For each subplot, the left, middle and right columns represent the full, in-sample (before 2017) and off-sample (2017-2020) periods, and the top and bottom (with symmetrical log scale) rows represent without and with 20bps transaction fees. The rebalance window is fixed at 84 days and the lookback window is fixed at 126 days, optimised in the in-sample period. The peripheral and central portfolios do not have the exact same number of assets in comparison, as SR-IFN selects based on confidence level instead of an exact parameter.

Figure 4.4 showcases the results in terms of the Sharpe Ratio for NASDAQ, FTSE, and HS300. The parameters $t_{\text{rebalance}} = 84$ and $t_{\text{lookback}} = 126$ are fixed, which are optimized in-sample from section 4.3.2, and the outcomes of the out-of-sample period are presented. For each subplot, the top row and bottom row represent experiments without and with 20bps transaction costs, with the bottom row being displayed in a symmetrical log scale for enhanced visualization and comparison. The Peripheral Topological Portfolio (PTP) in blue is the principal portfolio selected by SR-IFN. By varying the ConfLv of SR-IFN, we illustrate the performance with respect to different numbers of assets. Given that PTP is constructed from the disconnected assets of the correlation graph, its counterpart, Central Topological Portfolio (CTP) in yellow, represented by the connected assets from the correlation graph, is also exhibited as a supplement to portray the nearly symmetrical gain and loss. Since the number of assets is not a direct parameter in the algorithm but is controlled by the ConfLv, PTP and CTP do not have the exact same number of assets when comparing the two curves. Moreover, to showcase the efficacy of PTP, we also present the results for a randomly sub-sampled portfolio with the same number of assets as PTP, denoted as Random Benchmark Portfolio (RBP) in green, a Peripheral Benchmark Portfolio (PBP) in red that is constructed by selecting the assets with the least sum of pairwise correlation, as well as a simple long hold strategy represented by a dashed line.

In all three markets, the two peripheral portfolios, PTP and PBP, both yield superior performance compared to RBP and CTP, suggesting a clear advantage in adopting the peripheral portfolio as discussed in section 2.4.3. For out-of-sample experiments, PTP surpasses PBP when the number of assets is relatively large with no transaction cost, and if a 20bps transaction cost is applied, the range where PTP outperforms PBP extends. These findings corroborate that PTP is superior to the benchmark PBP with statistical significance and consistency across markets and conditions. Specifically, SR-IFN provides a more robust mechanism to identify assets with the most/least correlation than the simple empirical correlation method, and this effect is more likely to persist in the future period. Furthermore, by con-

trasting PTP and CTP, the gains and losses are roughly symmetrical around the Long/Hold dashed line, suggesting that the gain in PTP predominantly arises from selecting the peripheral assets as opposed to other factors.

To further refine our discussion, we place a restriction on our portfolio size to include more than 50 and less than 100 assets, aiming to mitigate the high variance at the tail of the performance distribution. When examining the out-of-sample period without transaction fees, the average Sharpe Ratio for NASDAQ is 1.10, while with the inclusion of 20bps transaction fees, it marginally decreases to 1.08. This is compared against a Long/Hold (L/H) benchmark of 1.07. In a similar vein, the FTSE index records an average Sharpe Ratio of 0.28 without transaction fees and 0.24 with these fees, against an L/H benchmark of -0.09. The HS300 index exhibits a Sharpe Ratio of 0.65 without transaction fees and 0.62 with the inclusion of 20bps transaction fees, compared to an L/H benchmark of 0.47. Thus, the net gain in the out-of-sample Sharpe Ratio equates to approximately 3% and 1% in the case of NASDAQ, 411%, and 367% for FTSE, and 38% and 32% for HS300, without and with transaction fees respectively.

The noteworthy performance observed in the FTSE and HS300 indices can likely be attributed to the specific market dynamics during the chosen period, characterized by a highly negative skew and a substantial maximum drawdown. For instance, during the 'golden period' of NASDAQ, a high beta and generally high correlation among the index component stocks result in a low signal-to-noise ratio when identifying the least correlated stocks. In contrast, in the more turbulent and less bullish market dynamics observed in the FTSE and HS300 indices, where the correlation among component stocks presents greater diversity, the underlying SR-IFN results in more significant findings in terms of correlation filtering and inference. In essence, SR-IFN for portfolio selection generally has a positive impact on the selection of the least correlated assets, leading to improved portfolio performance. This performance is more pronounced when the underlying market dynamic is less bullish and subject to more extreme losses.

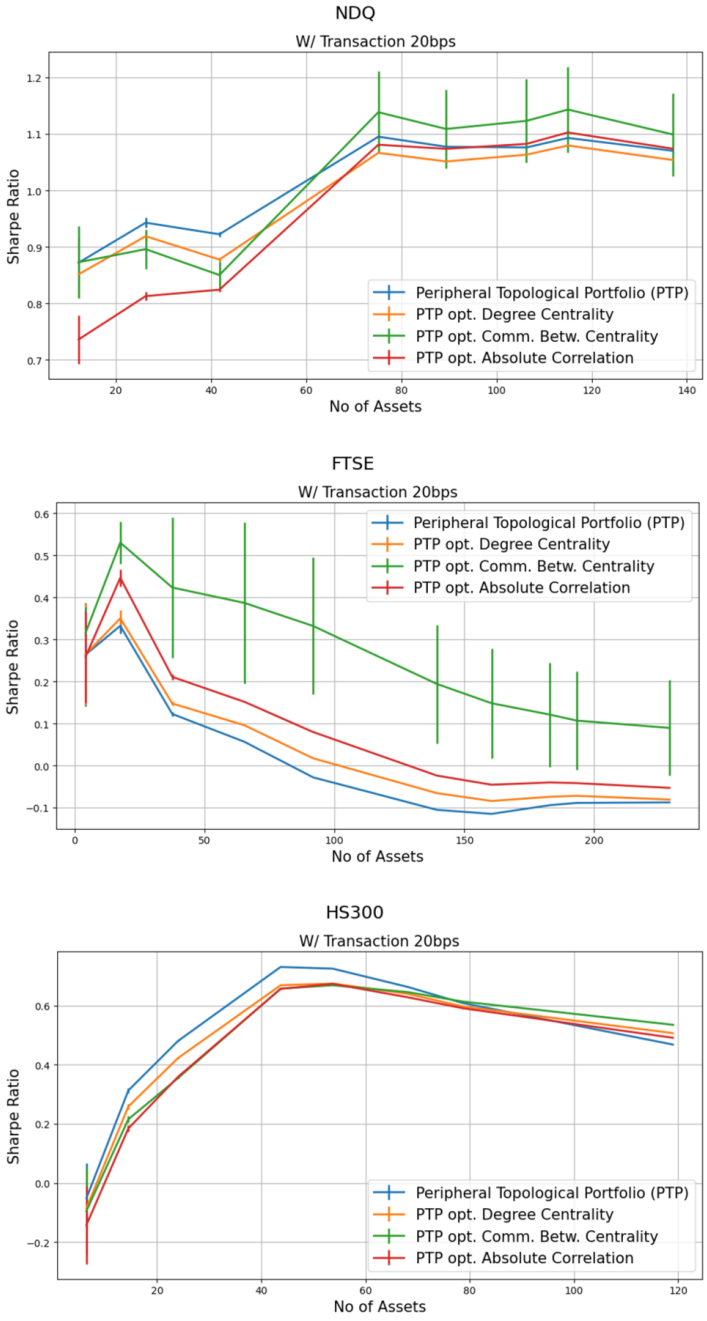


Figure 4.5: Portfolio optimization results in Sharpe Ratio across NASDAQ (top left), FTSE (top right) and HS300 (bottom) with 20bps transaction fee. We report the original Peripheral Topological Portfolio of equal weights as the benchmark (blue), and the optimised PTPs whose weights are inversely proportional to Degree Centrality (yellow), Communicability Betweenness Centrality (green) and Absolute Correlation (red). For each subplot, the left, middle and right columns represent the full, in-sample (before 2017) and off-sample (2017-2020) periods. The rebalance window is fixed at 84 days and the lookback window is fixed at 126 days, optimised in the in-sample period.

4.4.2 Topological Portfolio Optimization

In the preceding section, we have demonstrated the significant advantage of a peripheral portfolio selection strategy in the context of high drawdown periods. This strategy, founded on the least correlated portfolio, can be further refined by incorporating other topological properties to optimize the weighting of the selected portfolio. Herein, we continue to underscore the merit of a more peripheral portfolio, characterized by reduced correlation and superior performance, by assigning weights that are inversely proportional to centrality measures within the previously selected Peripheral Topological Portfolio (PTP). Degree Centrality, as one of the simplest measures of centrality, and Communicability Betweenness Centrality, a more complex but well-documented measure, are included in our study. Furthermore, given its intuitive nature and alignment with the overall theme of decorrelation, Absolute Correlation is also incorporated into our experiments.

Figure 4.5 depicts the performance across the three markets in terms of the Sharpe Ratio. Within each market, we plot PTP, serving as the benchmark, and three optimised versions of PTP wherein weights are inversely proportional to centrality measures, including Degree Centrality (yellow), Communicability Betweenness Centrality (green), and Absolute Correlation (red). Maintaining the same comparative framework, we restrict our analysis to portfolios comprising more than 50 and less than 100 assets, in order to mitigate the high variance at the tail of the performance distribution. For brevity, we limit our analysis to experiments incorporating 20 bps transaction fees. For the NASDAQ index, the average Sharpe Ratio is improved from 1.08 to 1.12, representing an approximate 4% improvement when optimized by Communicability Betweenness Centrality. Absolute Correlation yields an equal 1.08, while Degree Centrality results in a slightly inferior 1.06. For the FTSE index, Communicability Betweenness Centrality optimization improves the Sharpe Ratio from 0.24 to 0.42, an impressive 75% enhancement, while Absolute Correlation and Degree Centrality yield improvements to 0.31 (29%) and 0.25 (4%), respectively. For the HS300 index, the average Sharpe Ratio improves from 0.62 to 0.65 (approximately 5%) when optimized by Communicability Betweenness Cen-

trality, while remaining unchanged under the other two methods.

	NASDAQ			FTSE			HS300		
	L/H	PTP	PTP+CBC	L/H	PTP	PTP+CBC	L/H	PTP	PTP+CBC
Ann. Return	16.6%	16.2%	15.7%	-1.6%	1.1%	5.5%	8.7%	11.5%	12.1%
Ann. Std.Dev.	15.5%	15.0%	14.1%	18.4%	19.5%	17.7%	18.6%	18.7%	18.8%
Sharpe R.	1.07	1.08	1.12	-0.09	0.24	0.42	0.47	0.62	0.65
Max. Drawdown	-23.2%	-22.1%	-18.1%	-43.5%	-48.5%	-42.4%	-30.0%	-28.0%	-29.0%

Table 4.2: Aggregated performance statistics of L/H benchmark, Peripheral Topological Portfolio (PTP) and PTP optimised by Communicability Betweenness Centrality (CBC) in NASDAQ, FTSE and HS300. The table reports averaged statistics for portfolios with a number of assets between 50 and 100, including the annualized mean return, annualized return standard deviation, annualized Sharpe Ratio, daily return skewness and maximum drawdown.

Furthermore in Table 4.2, we report the aggregated performance for the Long/Hold benchmark (L/H), PTP and PTP optimised by CBC (PTP+CBC), the statistics are averaged across portfolios with a number of assets between 50 and 100. Apart from a superior Sharpe Ratio, our PTP+CBC demonstrates significant improvement in the risk matrices. PTP+CBC has reduced annualized return standard deviation by 1.4% in NASDAQ, 0.7% in FTSE and kept similar in HS300, as well as shrank the maximum drawdown by 5.1% in NASDAQ, 1.1% in FTSE and 1% in HS300.

This section illustrates the impact of weighting the portfolio inversely proportional to different centrality measures. We provide quantitative evidence of a robust improvement over the simple, equally-weighted PTP. Furthermore, Figure 4.5 demonstrates that, apart from the FTSE index where the effect is consistently dominant across all asset numbers, the effect is particularly pronounced for portfolios with larger asset numbers. One plausible explanation is that, for smaller PTPs, the assets are already optimally selected and much of the topological information has been extracted. As a result, additional optimization may suffer from a low signal-to-noise ratio, as the application of infinitesimally small weights effectively equates to deselection. This hypothesis aligns with the more pronounced effect observed in the FTSE index, given its effectively larger pool of component stocks compared to the other two indices.

4.5 Summary

In this study, we presented a novel, statistically robust bootstrapping method designed to enhance the information filtering network for modelling multivariate time series, hereby referred to as the Statistically Robust Information Filtering Network (SR-IFN). This method improves upon the existing Information Filtering Network (IFN) by reducing redundant edges formed due to applied graphical constraints. The SR-IFN accepts multivariate time-series observations as inputs and outputs a sparse similarity matrix and a network, both of which are subsequently employed for portfolio selection and optimization with constant rebalancing. Our experiments spanned a decade-long history across three distinct markets, utilizing the first 70% of the data to select parameters such as rebalancing frequency and lookback window size. The results reported are based on the off-sample data from the remaining three years. Our in-sample grid search for parameter tuning demonstrated consistent outperformance of the benchmark, mirroring the findings in the off-sample period, thereby reinforcing the robustness of the proposed method in even the most challenging financial applications.

Our findings indicate that the deployment of such an innovative approach results in a Sharpe Ratio improvement of 1%, 367%, and 32% with 20bps transaction costs for market indices. This is achieved by simply selecting a subset of composite stocks in the US, UK, and China markets, respectively. Moreover, the performance can be further amplified by optimizing the portfolio weights based on the centrality measures of the output network, yielding additional improvements of 4%, 75%, and 5%. The cumulative improvement derived from both approaches enhances the results by 5%, 567%, and 38% for the NASDAQ, FTSE, and HS300 indices, respectively. The disparities in the magnitude of improvement are likely attributed to the market dynamics of the selected period. For instance, NASDAQ was in its 'golden period', while the other two markets underwent significant drawdowns. Consequently, further improvement of an already efficient system (NASDAQ) proved more challenging than the other two, which serves as a testament to the method's resilience under extreme market conditions. Furthermore, despite a

marginal boost in the risk-adjusted reward in NASDAQ compared to the other two markets, the risk metrics are notably reduced in both annualized standard deviation of 1.4% and maximum drawdown of 5.1%. Additional findings reveal that the underlying method performs well with large-dimension data (number of assets) with computational efficiency.

Chapter 5

Network Filtering of Spatial-temporal GNN for Multivariate Time-series Prediction

5.1 Introduction

Intra-series temporal patterns and inter-series correlations are jointly the two cores in multivariate time-series forecasting. Recent advancement in deep learning has enabled strong temporal pattern mining. Recurrent Neural Network (RNN) [206], Long Short-Term Memory (LSTM) network [207] and Gated Recurrent Units (GRU) [208] demonstrate promising results in temporal modelling. Advanced work with attention mechanism, e.g., transformer [209, 210], further improves the performance and efficiency in temporal modelling by being able to prioritize certain temporal sequence instead of the entire history while enabling parallel computation in attention calculation [211]. However, existing methods don't take into account the correlation or other interdependencies matrices of time series explicitly, instead, they aim to learn such a relationship from input data, e.g., via embeddings. Historical attempts have been made to input covariance/correlation structure into neural networks, as these structures can be used as priors to reducing learning complexity. Matrix-based neural networks have been discussed [212, 213], but this approach is not specifically designed for the covariance/correlation matrix, and therefore fails to

directly and explicitly address the dependency in the covariance/correlation structure inside the calculation.

A graph is a mathematical structure to model relations between objects. The permutation-invariant, local connectivity and compositionality of graphs present a perfect data structure to simulate the correlation/covariance matrix. In fact, network science literature has long been including (sparse) covariance/correlation as a special network for analysis [214, 215, 216], and many network properties of covariance/correlation matrix contribute greatly to analytical and predictive tasks in the financial market [217, 218, 219]. Recently, graph neural networks (GNN) have been leveraged to incorporate the topological structure between entities. Hence, modeling inter-series correlation via graph learning is a natural extension to analyzing covariance/correlation matrix from a network perspective. Each variable (series) from a multivariate time series is a node in the graph, and the edge represents their latent inter-dependency. By propagating information between neighboring nodes, the graph neural network enables each time-series to be aware of correlated context.

Spatial-temporal graph neural network is the most used network structure for multivariate time-series problems in the literature [220, 139], as the temporal part extracts patterns in each uni-variate series with a LSTM/RNN/GRU, while the spatial part (GNN) models the relationship between series with a pre-defined topology or a graph representation learning algorithm. On one hand, existing GNNs heavily utilize a pre-defined topology structure which is not explicit in multivariate time-series, and does not reflect the temporal dynamical nature of time-series. On the other hand, many graph representation learning methods focus more on generating node embeddings rather than topological structure, and most of the embeddings depend on a pre-defined topological prior or attention mechanism [221, 220].

In this chapter, we propose an end-to-end framework termed Filtered Sparse Spatial-temporal GNN (FSST-GNN) for sales volume prediction of 50 products in 10 stores. By integrating modern spatial-temporal GNN with traditional matrix filtering/sparsification methods, we demonstrate the direct use of the (inverse) correlation matrix in GNN. Correlation filtering techniques generate a sparse inverse

correlation matrix from multivariate time-series, which can be inverted to a filtered correlation matrix. Both the (inverse) correlation can be used as a pre-defined topological structure or prior for further representation learning. With the designed architecture, we further illustrate that filtered graphs generates a positive impact in multivariate time-series learning, and sparse graphs acts as a contributing prior to guide attention mechanism in GNN.

5.2 Model Implementation

We first elaborate on the general framework of our model. As illustrated in Figure 5.1, the model consists of 5 main building blocks. A correlation graph generator is able to transform the multivariate time-series into a correlation graph where each node represents a single time-series and each edge between two nodes denotes their correlation. A standard transformation generates a full (inverse) correlation graph with (inverse) correlation edges between each node. In addition, correlation-filtering based transformation generates a full correlation graph with filtered correlation edges, or a sparse inverse correlation graph. We employ covariance shrinkage, graphical models and information filtering network as the three main correlation-filtering based graph generators. The feature generator generates initial input features for each node based on the multivariate time-series, and the details of the feature generator in the specific case study will be discussed in Section 5.3.1. The generated graph and the features from the two generators are then fed into a GNN to learn meaningful node embeddings as the spatial information. Similarly, the multivariate time-series is also fed into a LSTM to extract temporal information. Then, the spatial and temporal information are input in a multi-layer perceptron (MLP) as the read-out layer for the final output, the predicted sales volume. As the feature generator, LSTM and read-out MLP are general and based on standard settings, the sections following focus on the correlation graph generator and the employed GNN.

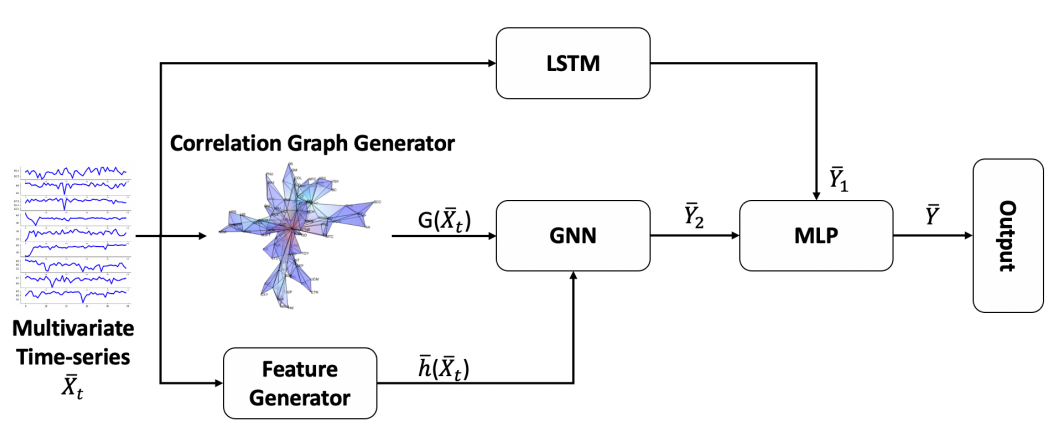


Figure 5.1: The overall model architecture, where $\hat{X}_t$ represents the input multivariable time series of t dimensions, $\hat{Y}_1$ is the output of LSTM, $\hat{Y}_2$ is the output of GNN, and the final output $\hat{Y}$ combined $\hat{Y}_1$ and $\hat{Y}_2$ through a MLP.

5.2.1 Correlation Graph Generator

5.2.1.1 Graphical LASSO

Graphical lasso is a statistical approach for deducing the structure of an undirected Gaussian graphical model, focusing on the concentration or precision matrix, which is the inverse of the covariance matrix. As a sparse penalized maximum likelihood estimator, its goal is to maximize likelihood while ensuring the resulting matrix has a high level of sparsity, characterized by a large number of zeros. This sparsity aids in pinpointing the most crucial variable relationships.

In managing the number of non-zero elements in the precision matrix, graphical lasso utilizes l_1 regularization, also known as lasso regularization. This method introduces a penalty proportional to the absolute value of the coefficients, favoring solutions with fewer non-zero elements. Consequently, graphical lasso is effective in unraveling a network's structure by determining direct relationships between variables, thus leading to a model that is both simpler and easier to understand.

Graphical LASSO is epressed in equation 2.1. We leverage Python's **sklearn.covariance.GraphicalLasso** library for implementation, and **sklearn.covariance.GraphicalLassoCV** [222] for cross valiation and regularization constant λ selection. Graphical LASSO sparsifies an inverse correlation matrix

which can be directly transformed into a sparse inverse correlation graph, while a full but filtered correlation graph can be obtained through the matrix inversion of the inverse correlation.

5.2.1.2 Maximally Filtered Clique Forest

We implement Maximally Filtered Clique Forest (MFCF), an information filtering network, for sparse precision matrix filtering. By setting the minimum and maximum clique size to 4, we simplify our solution to a TMFG-equivalent model discussed in Section 2.2.2. It generates sparse inverse correlation, which will undergoes similar transformation as Graphical LASSO to obtain inverse correlation and correlation graphs.

5.2.2 GNN

5.2.2.1 GCN

The Graph convolution network is proposed by Kipf and Welling in 2017 [223], which generates embeddings for each node in the graph. It takes original features in each node as the initial embeddings, then aggregates neighboring feature representations and updates the node embeddings through a message-passing-like network with the adjacency matrix. The layer-wise propagation rule updates the node features, which can be expressed as:

$$\mathbf{H}^{(l+1)} = \phi\left(\mathbf{D}^{(-1/2)}\mathbf{A}\mathbf{D}^{(-1/2)}\mathbf{H}^{l}\mathbf{W}^{l}\right) \tag{5.1}$$

where $\mathbf{H}^{(l)} \in \mathbb{R}^{F_l \times N}$ (where N is the number of nodes, and F_l is the number of features at layer l) is the matrix of node features at layer l, $\mathbf{W}^{l} \in \mathbb{R}^{F_{l+1} \times F_L}$ is the weight matrix for layer l, ϕ is the non-linear activation function, $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix with added self-connections, and $\mathbf{D}$ is the diagonal degree matrix of $\mathbf{A}$.

In the experiments, we have replaced the graph information, adjacency matrix, expressed in equation 5.1 by (inverse) correlation matrix, adjacency matrix and Laplacian matrix obtained by thresholding the (inverse) correlation matrix. The em-

pirical results suggest the superiority by simply employing the (inverse) correlation matrix. It can be seen as weighted adjacency matrix, where correlation coefficients are naturally scaled/normalized.

5.2.2.2 GAT

The implementation of GCN limits the model to be used only with static graphs. The embedding update is static across time, which assumes non-stationarity in time-series. Graph attention network uses masked multi-head attention mechanism to solve this issue by dynamically assigning attention coefficients between nodes. The normalized attention coefficient scalar a_{ij} is computed for nodes i and j based on their features (embeddings):

$$\alpha_{i,j} = \frac{\exp\left(\text{LeakyReLU}(\hat{a}^\top[\mathbf{W}h_i||\mathbf{W}h_j])\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}(\hat{a}^\top[\mathbf{W}h_i||\mathbf{W}h_k])\right)} \tag{5.2}$$

where $\mathbf{W} \in \mathbb{R}^{F_{l+1}\times F_l}$ is the shared weight matrix defined similarly to GCN, $h_i \in \mathbb{R}^{F_l}$, is the feature vector of node i in the graph, $\hat{a} \in \mathbb{R}^{F_{l+1}}$ is a learnable weight vector representing the attention mechanism to perform self-attention on each node and $||$ represents concatenation operation, $\mathcal{N}_i$ is the set of neighboring nodes of node i

The multi-head attention mechanism has been proposed by Vaswani et al. [209] which demonstrates superior and robust performance in network training. GAT incorporates the masked multi-head attention where attention is only computed between neighbouring nodes, and the output feature representation is expressed as:

$$h_i^{l+1} = \phi\left(\frac{1}{k}\sum_{k=1}^{K}(\sum_{j\in\mathcal{N}_i}\alpha_{i,j}\mathbf{W}^k h_j^l)\right) \tag{5.3}$$

where $\mathbf{W} \in \mathbb{R}^{F_{l+1}\times F_l}$ is the shared weight matrix, $h_i^{l+1} \in \mathbb{R}^{F_{l+1}}$ is the updated feature vector for node i in layer $l+1$, and the final representation is averaged between the K number of multi-head attention layers and applied a non-linearity ϕ.

Graph	Filtering	RMSE	MAE	MAPE
FSST-GNN (GCN)				
Cor	Empirical	10.12 ± 0.53	7.77 ± 0.39	$17.28\% \pm 1.18\%$
Cor	Shrinkage	9.99 ± 0.17	7.72 ± 0.09	$17.34\% \pm 0.61\%$
Cor	GLASSO	$\mathbf{9.76}^{*} \pm \mathbf{0.47}$	$\mathbf{7.52}^{*} \pm \mathbf{0.39}$	$\mathbf{16.96\%} \pm \mathbf{1.49\%}$
Cor	MFCF	9.80 ± 0.61	7.66 ± 0.43	$17.27\% \pm 1.13\%$
Inv Cor	Empirical	11.74 ± 0.99	8.69 ± 0.46	$20.08\% \pm 0.90\%$
Inv Cor	Shrinkage	10.59 ± 1.04	8.26 ± 0.78	$19.15\% \pm 2.21\%$
Inv Cor	GLASSO	$\underline{\mathbf{9.67}^{***} \pm \mathbf{0.41}}$	$\underline{\mathbf{7.55}^{**} \pm \mathbf{0.34}}$	$\underline{\mathbf{17.51\%}^{***} \pm \mathbf{1.54\%}}$
Inv Cor	MFCF	10.07 ± 0.59	7.99 ± 0.44	$17.87\% \pm 1.11\%$
Zeros	/	13.51 ± 0.16	10.28 ± 0.12	$22.04\% \pm 1.01\%$
Ones	/	12.33 ± 1.12	10.00 ± 0.98	$24.76\% \pm 2.33\%$
Identity	/	11.78 ± 0.52	9.23 ± 0.46	$21.01\% \pm 1.78\%$
LSTM				
/	/	16.34 ± 0.44	12.56 ± 0.25	$26.40\% \pm 0.98\%$

Table 5.1: Summary of forecasting results with different models, graphs and filtering methods. Highlighted in bold are the optimal RMSE, MAE and MAPE in each graph, and underlined is the absolute optimal results in the table. A LSTM results is attached as the baseline. A pairwise T-test has been performed, and the p-values for the best-performing results in each graph against the empirical graph results are highlighted next to the best-performing results, where * denotes 5% significance, ** for 1% and *** for 0.1% respectively.

5.3 Experiments

5.3.1 Setup

We test our model on a Kaggle playground code competition, Store Item Demand Forecasting Challenge [224]. The dataset consists of 5-year sales time-series data of 50 products in 10 different stores. For simplicity, we re-formulate the problem as 50 mini-problems, each focusing on 1 product in 10 different stores. At each time stamp, the temporal component regresses each of the 10-time series individually based on its historical value. The dependency between them is reflected by the final embeddings generated from the spatial component. The outputs from each component are subsequently concatenated and, by a read-out layer, to generate final daily forecasting for the product. We assume stationarity in the time series, therefore, we separate the training and testing data as 80% and 20% of the raw dataset.

The temporal component of the FSST-GNN is an LSTM, which has an input size of (t_{lb}, 10) where $t_{lb} = 14$ is the look-back window size of historical sales,

and 10 is the number of different stores. The feature generator produces node features (initial embeddings). We employ the four moments (mean, standard deviation, skewness and kurtosis) of the sales time-series distribution based on each 14-day look back window. The correlation graph generator generates a graph with edges that represents the correlation between any two of the 14-day sample time series in the 10 stores. Then, the generated node features and edges are input into the GNN. In the experiments, a GCN and a GAT have been used as the spatial component.

To understand the effect of filtering and sparsification for multivariate time-series graph learning, we perform 4 sets of experiments: 1) FSST-GNN (GCN) on different filtered correlation graphs; 2) FSST-GNN (GCN) on different filtered inverse correlation graphs; 3) FSST-GNN (GCN) on GLASSO-filtered and MFCF-filtered inverse correlation graph with different levels of sparsity; and 4) FSST-GNN (GAT) on GLASSO-filtered and MFCF-filtered inverse correlation graph with different levels of sparsity. Each experiment has been re-computed 10 times with different random seeds, and the final results are averaged for statistical robustness.

Sparsity	RMSE	MAE	MAPE	Sparsity	RMSE	MAE	MAPE
GLASSO				MFCF			
77.2%	10.24 ± 0.61	8.03± 0.47	18.89%±1.35%	76.6%	11.35 ± 0.76	8.76 ± 0.61	20.36%±1.97%
71.0%	10.34 ± 0.79	8.05 ± 0.58	18.66%±1.29%	72.3%	10.23 ± 0.26	8.06 ± 0.44	18.13%±1.14%
66.6%	9.80 ± 0.41	7.66 ± 0.33	17.75%±0.87%	68.6%	10.68 ± 0.80	8.32 ± 0.52	19.56%±1.30%
60.0%	**9.67*** ± **0.41**	**7.55***± **0.34**	**17.51%** ± **1.54**%	61.3%	**10.07**** ± **0.59**	**7.99**** ± **0.44**	**17.87%***** ± **1.11**%
56.5%	9.86 ± 0.58	7.74 ± 0.53	18.24%±1.66%	58.0%	10.19 ± 0.41	8.12 ± 0.29	18.29%±0.49%
51.3%	10.06 ± 0.57	7.86 ± 0.45	17.68%±1.20%	54.8%	10.31 ± 0.52	8.09 ± 0.39	18.22%±1.26%
43.3%	9.99 ± 0.27	7.88 ± 0.19	17.60%±0.47%	43.7%	10.75 ± 0.68	8.16 ± 0.40	18.44%±0.55%

Table 5.2: Summary of forecasting results of FSST-GNN (GCN) with different filtering methods and sparsity on inverse correlation graph. Highlighted in bold are the optimal RMSE, MAE and MAPE in each model-sparsity combination, and underlined is the absolute optimal results in the table. A pairwise T-test has been performed, and the p-values for the best-performing results in each graph against the second-best-performing results are highlighted next to the best-performing results, where * denotes 5% significance, ** for 1% and *** for 0.1% respectively.

5.3.2 Results

We compute the root mean square error (RMSE), mean average error (MAE) and mean average percentage error (MAPE) of the predicted sales number of all 50 products in 10 stores with the ground truth label in Table 5.1 as the evaluation matrix to analyze the effectiveness of filtering methods over FSST-GNN with GCN on

Sparsity	RMSE	MAE	MAPE	Sparsity	RMSE	MAE	MAPE
GLASSO				MFCF			
77.2%	9.88 ± 0.59	7.58 ± 0.43	16.17%±0.53%	76.6%	10.47 ± 0.75	8.09 ± 0.68	17.61%±2.64%
71.0%	10.03 ± 0.65	7.73 ± 0.52	16.56%±1.07%	72.3%	10.27 ± 0.62	7.86 ± 0.46	17.24%±0.99%
66.6%	9.63 ± 0.36	7.42 ± 0.25	15.82%±0.25%	68.6%	9.90 ± 1.17	7.60 ± 0.95	16.01%±1.78%
60.0%	**9.58*** ± **0.31**	**7.37**± **0.20**	**15.62%*** ± **0.26%**	61.3%	**9.46*** ± **0.68**	**7.31**** ± **0.45**	**15.44%***** ± **0.19%**
56.5%	9.64 ± 0.34	7.41 ± 0.23	15.80%±0.47%	58.0%	9.65 ± 0.46	7.53 ± 0.32	15.63%±0.54%
51.3%	9.75 ± 0.33	7.55 ± 0.31	17.28%±1.18%	54.8%	9.81 ± 0.53	7.53 ± 0.37	16.03%±0.54%
43.3%	10.12 ± 0.53	7.77 ± 0.39	17.28%±1.18%	43.7%	9.90 ± 0.38	7.63 ± 0.30	16.31%±0.92%

Table 5.3: Summary of forecasting results of FSST-GNN (GAT) with different filtering methods and sparsity on inverse correlation graph. Highlighted in bold are the optimal RMSE, MAE and MAPE in each model-sparsity combination, and underlined is the absolute optimal results in the table. A pairwise T-test has been performed, and the p-values for the best-performing results in each graph against the empirical graph results are highlighted next to the best-performing results, where * denotes 5% significance, ** for 1% and *** for 0.1% respectively.

the correlation and inverse correlation graph respectively. Since all filtering methods are parametric, the table reports the optimal results from covariance shrinkage (Shrinkage), graphical LASSO (GLASSO) and MFCF, which are obtained through grid-search. We also include a fully connected graph of a matrix of ones, two fully disconnected graphs of a matrix of zeros and an identity matrix as benchmarks for comparison. In addition, a plain LSTM is also presented as the baseline model where no graphical/spatial information is input.

In Table 5.1, it is evident that all FSST-GNN (GCN) outperforms the plain LSTM, which confirms the efficacy of considering the spatial information in multivariate time-series problems. Other benchmarks of fully connected/disconnected graphs are also presented, and their results in all three measurements are effectively inferior to any (inverse) correlation-based graph methods. These results further assert the information gain from meaningful spatial graphs. The significance in the inverse correlation graph may also suggest that the local and global patterns captured by inverse covariance would result a more effective filtering and sparsification in relative to the local effect of correlation graph.

Highlighted in each column of Table 5.1 are the best results in first two experiment: 1) FSST-GNN (GCN) on the correlation graph; 2) FSST-GNN (GCN) on the inverse correlation graph. In correlation graph cases, filtered correlation graphs demonstrate superior results than the original Empirical correlation graph. Both MFCF and GLASSO filtering are operated on the inverse correlation for sparsifi-

cation, and then inverted back to a full correlation graph, while Shrinkage operates directly on correlation. Therefore, the superior results in MFCF and GLASSO than Shrinkage may suggest a stronger filtering effect behind graph/network-based methods, and inversion does not affect filtering.

To understand the effect in filtering and sparsification, results from the same setup with inverse correlation graphs are compared, where full and Shrinkage-filtered inverse correlation graphs are full graphs and GLASSO-filtered and MFCF-filtered graphs are sparse graphs. In this case, Shrinkage filters a correlation and inverts it to an inverse correlation. Comparably to the correlation graph case, Shrinkage consistently yields better result than the Empirical, which further validates that the filtering mechanism is hardly impacted by inversion operation. Furthermore, we observe even more significant results from two sparse graphs filtered by GLASSO and MFCF. This advantage could possibly come from both the filtering, the sparsification, as well as their combined effect. To further investigate the sole efficacy of sparsification, we perform the third and fourth sets of experiments: 3) FSST-GNN (GCN) on GLASSO-filtered and MFCF-filtered inverse correlation graph with different levels of sparsity; and 4) FSST-GNN (GAT) on GLASSO-filtered and MFCF-filtered inverse correlation graph with different levels of sparsity.

Presented in Table 5.2 and Table 5.3 are the results with different levels of sparsity. We select the parameter to match the sparsity level between GLASSO and MFCF for comparison. It is seen that at around 60% sparsity, the highlighted best results are achieved for both MFCF and GLASSO in FSST-GNN (GCN) and FSST-GNN (GAT) models. Moreover, as the sparsity deviates away from this local minimum, the three errors start to increase, which may suggest an optimal sparsity structure of the inverse correlation graph in our experimental case. In addition, this optimal structure is independent of the chosen model. Furthermore, as illustrated in equation 5.3, GAT by default does not account for edge weights in weighted graphs (correlation graphs) as GCN. Hence, the sparse inverse correlation graph serves as a thresholded adjacency matrix, where 0 entries are interpreted as disconnection between nodes. Then, attention, which is only calculated between linked nodes, acts

as the edge weights. Namely, the superior performance in Table 5.2 is a mixture of filtering and sparsity, but the performance in Table 5.3 is merely determined by the sparsity of the input graph without a filtering mechanism.

5.4 Summary

The academic literature has presented many GNN-based graph sparsification methods. However, none of them explicitly addresses the filtering and sparsification from a time-series perspective. In small sample time-series problems, especially in finance, graph structure learning models, e.g., graph representation learning, are highly prone to noise. In this chapter, we designed an end-to-end filtered sparse spatial-temporal graph neural network for time-series forecasting. Our model leverages and integrates traditional matrix filtering methods with modern graph neural networks to achieve robust results, and show the use of a simple and efficient architecture. We employed three different matrix filtering methods, covariance shrinkage, graphical LASSO and information filtering network-maximally filtered clique forest to show a positive gain in graph filtering to graph learning. The results from the three methods surpass all of the benchmark approaches, including an LSTM with no graphical information, the same FSST-GNN architecture with fully connected, disconnected graphs and unfiltered graphs.

In the experiments, we found the sparse graph in GAT serves only as an indication of which pairs of nodes require attention calculation, and the advantages from sparsity are significant. The filtered correlation matrix in GCN is interpreted and used as a weighted adjacency matrix for direct graph convolution, where the efficacy of filtering is also obvious. Furthermore, the optimal combined effect of filtering and sparsification in FSST-GNN (GCN) with inverse correlation implies the two contributing factors are complementary. Therefore, by incorporating weighted graphs in GAT like Grassia & Mangioni [225], we may further improve the performance of attention-based graph neural networks.

Current work is based on a synthetic dataset from a Kaggle competition for sales prediction. Further work will be applied with real-world financial data for

practical problems, e.g., portfolio optimization, risk management and price forecasting. The temporal and spatial components of the current architecture are designed to compute in parallel and combined in the end. Therefore, temporal information does not directly contribute to the spatial filtered graph generation or graph node feature generation. In the next phase of this study, we aim to develop a stacked architecture, where temporal signals contribute to spatial graph filtering/sparsification.

Chapter 6

Homological Neural Networks: A Sparse Architecture for Multivariate Time-series

6.1 Introduction

In this work, we propose a novel deep learning architecture that can be used for multivariate time-series prediction that keeps into account higher-order interactions in the dependency structure as topological priors. Higher-order graphs are networks that connect not only vertices with edges (i.e. low-order 1-dimensional simplices) but also higher-order simplices (A N-simplex is a shape in N-dimensional space formed by connecting $N+1$ vertices, e.g., 0-simplex is a point, 1-simplex is an edge, 2-simplex is a triangle, and 3-simplex is a tetrahedron) [226]. Indeed, any higher-order component can be described as a combination of lower-order components (i.e. edges connecting two vertices, triangles connecting three edges, ...). The transformation between a network representation in terms of a set of lower-order components to a set of higher-order components is called homology. In this work, we propose a novel multi-layer deep learning unit capable of fully representing the homological structure of data and dub it Homological Neural Network (HNN). This is a feed-forward unit where the first layer represents the vertices, the second the edges, the third the triangles, and so on. Each layer connects with the next

homological level accordingly to the network's topology representing dependency structures of the underlying input dataset. Information only flows between connected structures at different order levels, and homological computations are thus obtained. Neurons in each layer have a residual connection to a post-processing readout unit. The HNN's weights are updated through backward propagation using a standard gradient descent approach. Given the higher-order representation of the dependency structure in the data, this unit should provide better computational performances than those of fully connected multi-layer architectures. Furthermore, given the network representation's intrinsic sparsity, this unit should be computationally more efficient, and results should be more intuitive to interpret. We test these hypotheses by evaluating the HNN unit on two application domains traditionally challenging for deep learning models: tabular data and time series regression problems.

This work builds upon a vast literature concerning complex network representation of data dependency structures [227, 228]. Networks are excellent tools for representing complex systems both mathematically and visually, they can be used for both qualitatively describing the system and quantitatively modeling the system properties. A dense graph with everything connected with everything else (complete graph) does not carry any information, conversely, too sparse representations are oversimplifications of the important relations. There is a growing recognition that, in most practical cases, a good representation is provided by structures that are locally dense and globally sparse. In this chapter we use a family of network representations, named Information Filtering Networks (IFNs), that have been proven to be particularly useful in data-driven modeling [229, 17, 230]. The proposed methodology exploits the power of a specific class of IFNs, namely the Triangulated Maximally Filtered Graph (TMFG), which is a maximally planar chordal graph with a clique-three structure made of tetrahedra [231]. The TMFG is a good compromise between sparsity and density and it is computationally efficient to construct. It has the further advantage of being chordal (every cycle of four or more vertices has a chord) which makes it possible to directly implement probabilistic graphical mod-

eling on its structure (chordal graph is decomposable, easily factorizable, and easier to compute marginal distributions and conditional probabilities from a joint distribution) [17].

6.2 A novel representation for higher order networks and its use for HNN construction

The representation of undirected graphs explicitly accounts for the vertices and their connections through edges and, instead, does not explicitly account for other, higher-order, structures such as triangles, tetrahedra, and, in general, d-dimensional simplexes. Indeed, usually, an undirected graph is represented as a pair of sets, $\mathscr{G} = \{V, E\}$: the vertex set $V = \{v_1, ..., v_p\}$ and the edge set E which is made of pairs of edges $V = \{v_i, v_j\}$. The associated graphical representation is a network where vertices, represented as points, are connected through edges, represented as segments. This encoding of the structure accounts only for the edges skeleton of the network. However, in many real-world scenarios, higher-order sub-structures are crucial for the functional properties of the network and it is therefore convenient – and sometimes essential – to use a representation that accounts for them explicitly.

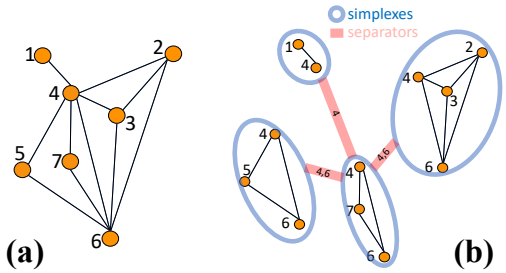


Figure 6.1: **(a)** Visual example of a higher order network made of 7 vertices, 11 edges, 6 triangles, and 1 tetrahedron. **(b)** This higher-order network is a clique tree made of four cliques (maximal cliques highlighted in the circles) connected through three separators (the tick red edges). One can observe that the separator constituted by the vertex '4' has multiplicity 1, while the separator constituted of the edge '4-6' has multiplicity 2 and indeed it appears twice.

A simple higher-order representation can be obtained by adding triplets (triangles), quadruplets (tetrahedra), etc. to the sets in $\mathscr{G}$. However, the associated higher-order network is hard to handle both visually and computationally. In this

chapter, we propose an alternative approach, which consists of a layered representation that explicitly takes into account the higher-order sub-structures and their interconnections. Such a representation is very simple, highly intuitive, of practical applicability as computational architecture, and, to the best of our knowledge, it has never been proposed before.

The proposed methodology is entirely based on a special class of networks: chordal graphs. These networks are constituted only of cliques organized in a higher order tree-like structure (also referred to as 'clique tree'). This class of networks is very broad and it has many useful applications, in particular for probabilistic modelling [26]. A visual example of a higher-order chordal network (a clique-tree), with 7 vertices, 11 edges, 6 triangles, and 1 tetrahedron, is provided in Figure 6.1. In the figure, the maximal cliques (largest fully-connected subgraphs) are highlighted and reported, in the right panel, as clique-tree nodes. Such nodes are connected to each other with links that are sub-cliques called separators. Separators have the property that, if removed from the network, they disconnect it into a number of components equal to the multiplicity of the separator minus one. In higher-order networks, cliques are the edge skeletons of simplexes. A 2-clique is a 1-dimensional simplex (an edge); 3-clique is a 2-dimensional simplex (a triangle); and so on with $(d+1)$-cliques being the skeleton of d-dimensional simplexes.

To represent the complexity of a higher-order network we propose to adopt a layered structure where nodes in layer d represent d-dimensional simplexes. The structures start with the vertices in layer 0; then a couple of vertices connect to edges represented in layer 1; edges connect to triangles in layer 2; triangles connect into tetrahedra in layer 3, and so on. This is illustrated in Figure 6.2. Such representation has a one-to-one correspondence with the original network but shows explicitly the simplexes and sub-simplexes and their interconnection in the structure. All information about the network at all dimensions is explicitly encoded in this representation including elements such as maximal cliques, separators, and their multiplicity (see caption of Figure 6.2).

It is worth noting the resemblance of this layered structure with the layered

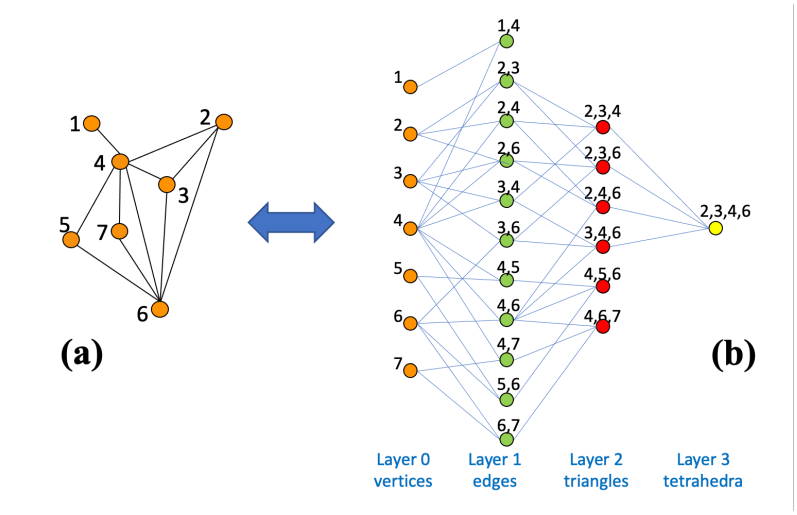


Figure 6.2: Higher order homological representation of the chordal graph in Figure 6.1 (reproduced in **(a)**). **(b)** Nodes in each layer, L_d, represent the d-dimensional simplexes in the structure. The links between nodes in layers d and $d+1$ are the connections between d to $d+1$ simplexes in the network. The degree on the left of nodes in L_d is always equal to d. The degree on the right of nodes in L_d can instead vary. The d-dimensional simplexes with no connections towards $d+1$ are the maximal cliques in the network (i.e. the nodes in the clique tree in Figure 6.1(b)).

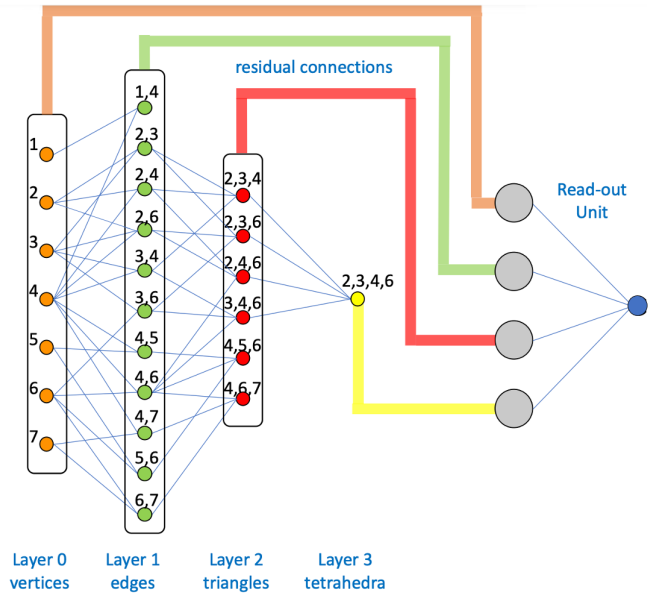


Figure 6.3: The Homological Neural Network (HNN) unit is constructed by using as input layer 0 of the homological representation of the dependency structure (see Figure 6.2(b)) and then feeding forward through the homological layers. The output is produced by a readout unit that connects all neurons in the layers. The HNN is essentially a sparse MLP unit with residual connections.

architecture of deep neural networks. Indeed, we leverage this novel higher-order network representation as the neural network architecture of the HNN unit. In our

experiments, the HNN is implemented from the TMFG generated from correlations. TMFG is computationally efficient, and can thus be used to dynamically re-configure the HNN according to changeable system conditions [140]. The HNN architecture is illustrated in Figure 6.3. Essentially it is made by the layered representation of Figure 6.2 with the addition of the residual connections linking each neuron in each simplex layer to a final read-out layer. Such HNN is a sparse MLP-like neural network with extra residual connections and it can be employed as a modular unit. It can directly replace fully connected MLP layers in several neural network architectures. In this chapter, the HNN unit is implemented using the standard PyTorch deep learning framework, while the sparse connection between layers is obtained thorugh the "sparselinear"[1] PyTorch library.

6.3 Design of neural network architectures with HNN units for time series studies

We investigate the performances of HNN units in two traditionally challenging application domains for deep learning: tabular data and time series regression problems. To process tabular data, the HNN unit can be directly fed with the data and it can be constructed from correlations by using the TMFG. In this case, the HNN unit acts as a sparsified MLP. This architecture is schematically shown in Figure 6.4. Instead, in spatio-temporal neural networks, the temporal layers are responsible for handling temporal patterns of individual series, whereas the spatial component learns their dependency structures. Consequently, the temporal part is usually modeled through the usage of recurrent neural networks (e.g. RNNs, GRUs, LSTMs), while the spatial component employs convolutional layers (e.g. CNNs) or aggregation functions (e.g. MLPs, GNNs).

Figure 6.5 presents the spatio-temporal neural network architecture employed in our multivariate time series experiments. The architecture consists of an LSTM for the temporal encoding of each time series and a graph generation unit that takes into account the correlation between different time series. This unit models time

[1] `https://github.com/hyeon95y/SparseLinear`

series as nodes and pairwise correlations as edges by imposing the topological constraints typical of the TMFG: planarity and chordality. The HNN is built based on the resulting sparse TMFG and aggregates each of the encoded time series from the LSTM, generating the final output.

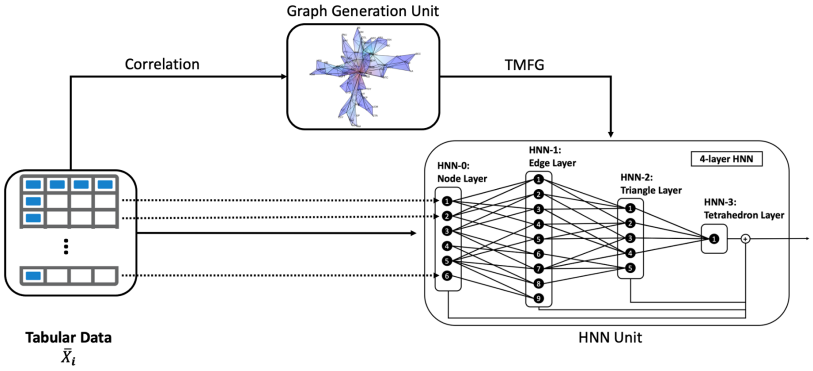


Figure 6.4: General HNN architecture is a sparsified MLP. The input data is processed by a Graph Generation Unit to construct a prior sparse graph to represent spatial interdependencies between the feature columns. The prior graph guides the design of the HNN unit which then processes and transforms the feature columns into the final output.

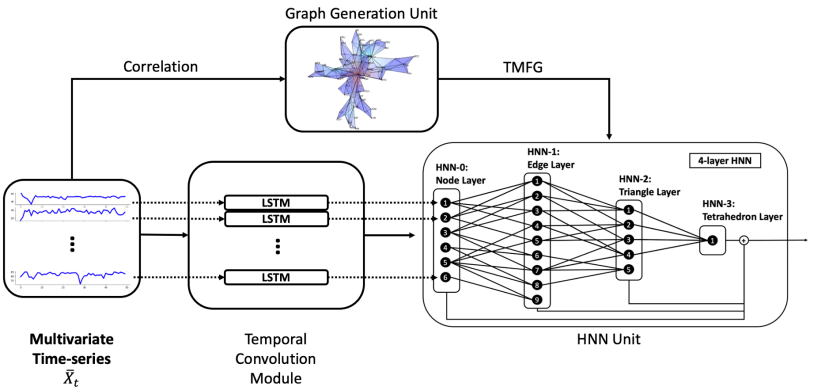


Figure 6.5: LSTM-HNN architecture for time-series data. The multivariate time-series is processed by a Graph Generation Unit to construct a prior sparse graph to represent spatial interdependencies, and each of the multivariate time series is processed separately by LSTM in the Temporal Convolution Module to harness the temporal information. The prior graph guides the design of the HNN unit which then aggregates the single temporal representations from LSTMs into the final output.

		solar-energy				exchange-rates			
		Horizon (days)				Horizon (days)			
Model	Metrics	3	6	12	24	3	6	12	24
LSTM-HNN	RSE	**0.190****	**0.270****	**0.354****	**0.446****	**0.022****	**0.027*****	**0.040****	**0.049****
	CORR	**0.981***	**0.964****	**0.942****	**0.902*****	**0.976*****	**0.968*****	**0.956****	**0.938****
LSTM-MLP-HNN	RSE	0.207	0.292	0.365	0.454	0.028	0.034	0.046	0.054
	CORR	0.980	0.959	0.936	0.893	0.965	0.957	0.945	0.928
LSTM-MLP-res	RSE	0.245	0.340	0.409	0.501	0.031	0.035	0.052	0.059
	CORR	0.972	0.944	0.905	0.898	0.850	0.829	0.835	0.828
LSTM-MLP	RSE	0.307	0.361	0.425	0.697	0.029	0.037	0.054	0.056
	CORR	0.956	0.937	0.898	0.723	0.845	0.838	0.834	0.824

Table 6.1: Relative Standard Error (RSE) and CORR (correlation). The best-performing results in a given metric and horizon are highlighted in bold. In addition, a paired T-test has been performed, and the p-values for the LSTM-HNN against the second-best-performing model (LSTM-MLP-res) in the given metrics and horizon are highlighted next to the best-performing results, where * denotes 5% significance, ** for 1% and *** for 0.1% respectively. The absence of * indicates statistical equivalence between the best-performing and LSTM-HNN models.

		solar-energy				exchange-rates			
		Horizon (days)				Horizon (days)			
Model	Metrics	3	6	12	24	3	6	12	24
LSTM-HNN	RSE	0.190	0.270	0.354	0.446	0.022	0.027	0.040	0.049
	CORR	0.981	0.964	0.942	0.902	0.976	0.968	0.956	**0.938**
MTGNN	RSE	**0.177****	**0.234*****	**0.310****	**0.427***	0.019	0.025	0.034	0.045
	CORR	**0.985**	**0.972****	**0.950****	0.903	0.978	0.970	0.955	0.937
TPA-LSTM	RSE	0.180	0.234	0.323	0.438	**0.017****	**0.024**	**0.034**	**0.044**
	CORR	0.985	0.974	0.948	**0.908****	**0.979***	0.970	**0.956**	0.938
LSTNet-skip	RSE	0.184	0.255	0.325	0.464	0.022	0.028	0.035	0.044
	CORR	0.984	0.969	0.946	0.887	0.973	0.965	0.951	0.935
RNN-GRU	RSE	0.193	0.262	0.416	0.485	0.019	0.026	0.040	0.062
	CORR	0.982	0.967	0.915	0.882	0.978	**0.971**	0.953	0.922
GP	RSE	0.225	0.328	0.520	0.797	0.023	0.027	0.039	0.058
	CORR	0.975	0.944	0.851	0.597	0.871	0.819	0.848	0.827
VARMLP	RSE	0.192	0.267	0.424	0.684	0.026	0.039	0.040	0.057
	CORR	0.982	0.965	0.905	0.714	0.860	0.872	0.828	0.767
AR	RSE	0.243	0.379	0.591	0.869	0.022	0.027	0.035	0.044
	CORR	0.971	0.926	0.810	0.531	0.973	0.965	0.952	0.935

Table 6.2: Relative Standard Error and correlation. The best-performing results in a given metric and horizon are highlighted in bold. In addition, a paired T-test has been performed, and the p-values for the best-performing result against LSTM-HNN in the given metrics and horizon are highlighted next to the best-performing results, where * denotes 5% significance, ** for 1% and *** for 0.1% respectively. The absence of * indicates statistical equivalence between the best-performing and LSTM-HNN models. When LSTM-HNN is the best-performing result, then the t-test is conversely performed against the second best-performing result.

6.4 Results

The HNN module can be used as a portable component along with different types of neural networks to manage various input data structures and downstream tasks.

In this Section, we apply HNN to process dependency structures in time series modelling after temporal dependencies are handled through the LSTM architecture. We use two different datasets which have been extensively investigated in the multivariate time-series literature [139]: the solar-energy dataset from the National Renewable Energy Laboratory, which contains the solar-energy power output collected from 137 PV plants in Alabama State in 2007; and a financial dataset containing the daily exchange-rates rates of eight foreign countries including Australia, British, Canada, Switzerland, China, Japan, New Zealand, and Singapore in the period from 1990 to 2016 (see Table A.9 in Appendix for further details).

Analogously with the tabular data, we first compare the outcomes of LSTM-HNN with those obtained with adapted MLP units. Specifically, LSTM units plus an MLP (LSTM-MLP); LSTM units plus an MLP with added residual connections to the final read-out layer (LSTM-MLP-res); and LSTM units plus a sparse MLP of the same layout as HNN without residual connections (LSTM-MLP-HNN). We then compare the LSTM-HNN results with traditional and state-of-the-art spatio-temporal models for multivariate time-series problems: auto-regressive model (AR) [131]; a hybrid model that exploits both the power of MLP and auto-regressive modelling (VARMLP) [232]; a Gaussian process (GP) [132]; a recurrent neural network with fully connected GRU hidden units (RNN-GRU) [139]; a LSTM recurrent neural network combined with a convolutional neural network (LSTNet) [133]; a LSTM recurrent neural network with attention mechanism (TPA-LSTM) [134]; and a graph neural network with temporal and graph convolution (MTGNN) [139].

We evaluate performances of the LSTM-HNN and compare them with the ones achieved by benchmark methodologies by forecasting the solar-energy power outputs and the exchange-rates values at different time horizons with performances measured in terms of relative standard error (RSE) and correlation (CORR) (see Table 6.1). We underline that LSTM-HNN significantly outperforms all MLP-based models. On solar-energy data, LSTM-HNN reduces RSE by 38%, 25%, 17%, and 36% from LSTM-MLP and 8%, 7%, 3%, and 2% from LSTM-MLP-res across four

horizons. On exchange-rates data, LSTM-HNN reduces RSE by 23%, 28%, 26%, and 13% from LSTM-MLP and 19%, 20%, 14%, and 10% from LSTM-MLP-res across four horizons.

We also notice that the residual connections from each layer to the final read-out layer are effective both in the HNN architecture (i.e. LSTM-HNN outperforms LSTM-MLP-HNN) and within the MPL models (i.e. LSTM-MLP-res outperforms LSTM-MLP). In order to illustrate the significance of the gain, a paired t-test of LSTM-HNN against LSTM-MLP-res has been performed revealing that all differences are significant at 1% or better with the only exception for the correlation at horizon 3 in the solar-energy output data.

The comparison between the results for LSTM-HNN and the other benchmark models is reported in Table 6.2. Results reveal that LSTM-HNN consistently outperforms all three non-RNN-based methods (AR, VARMLP and GP) on both datasets. It also outperforms LSTNet-skip results. LSTM-HNN outperforms RNN-GRU for all datasets and horizons except for the correlation in the exchange rates at horizon 6 where it returns an equivalent result accordingly with the paired t-test that was conducted between LSTM-HNN and the best-performing model. LSTM-HNN is instead slightly outperformed by MTGNN in most results for solar-power and by TPA-LSTM in several results for exchange-rates. It must be however noticed that these are massive deep-learning models with a much larger number of parameters (respectively 1.5 and 2.5 times larger than LSTM-HNN for the solar-energy datasets and 10 and 26 times larger for the exchange-rates datasets, see Table A.10).

6.5 Summary

In this chapter we introduced Homological Neural Networks (HNNs), a novel deep-learning architecture for multivariate time-series prediction based on a higher-order network representation of multivariate data dependency structures. This architecture can be seen as a sparse MLP with extra residual connections and it can be applied in place of any fully-connected MLP unit in composite neural network models. We test the effectiveness of HNNs on tabular and time-series heteroge-

neous datasets. Results reveal that HNN, used either as a standalone model or as a modular unit within larger models, produces better results than MLP models with the same number of neurons and layers. We compare the performance of HNN with both fully-connected MLP, MLP sparsified with the HNN layered structure, and fully-connected MLP with additional residual connections and read-out unit. We design an experimental pipeline that verifies that the sparse higher-order homological layered structure on which HNN is built is the main element that eases the computational process. Indeed, we verify that the sparsified MLP with the HNN structure (MLP-HNN) over-performs all other MLP models. We also verify that the residual links between layers and the readout unit consistently improve HNN performances. Noticeably, although residual connections also improve fully-connected MLP performances, results are still inferior to the ones achieved by sparse MLP-HNN. We demonstrate that HNNs' performances are in line with state-of-the-art best-performing computational models, however, it must be considered that they have a much smaller number of parameters, and their processing architecture is easier to interpret.

In this chapter, we built HNNs from TMFG networks computed on pure correlations. TMFG are very convenient chordal network representations that are computationally inexpensive and provide opportunities for dynamically self-adjusting neural network structures. Future research work on HNN will focus on developing an end-to-end dynamic model that addresses the temporal evolution of variable interdependencies. TMFG is only one instance of a large class of chordal higher-order information filtering networks [27] which can be used as priors to construct HNN units. The exploration of this larger class of possible representations is a natural expansion of the present HNN configuration and will be pursued in future studies.

Chapter 7

General Conclusions

7.1 Summary of Contributions

This thesis presents a novel approach by employing complex networks to model multivariate time series, where each temporal sequence is interpreted as a node and the bilateral dependency represents the edge uniting these nodes. Consequently, the linear and nonlinear constituents of the multivariate system are illustrated by the network or graph, facilitating subsequent analysis and transformation for diverse subsequent tasks. The pivotal components of the four main chapters rely on information filtering methodologies to extract statistically significant data from the network to yield a sparse yet meaningful representation, which is subsequently leveraged to mitigate noise and jumps in financial time series, and facilitate the designs of novel neural network architecture for time-series data. This depiction is then used for cross-asset portfolio modelling and plays a crucial role in the formulation of time-series neural network architectures. Empirical results obtained consistently validate the efficacy of this approach that employs network/graph models for multivariate time series.

In Chapter 3, we derive the sparse inverse covariance of the underlying assets in a portfolio, and demonstrate its superiority over the full inverse covariance for Markowitz portfolio optimization, due to its reduced noise content. The sparse inverse covariance further finds application as a component of a distance metric for time-series clustering, enabling the segmentation of the market into two states char-

acterized by high and low log-likelihood. We demonstrate that portfolios optimized using data from the cluster that performs better yield superior outcomes than portfolios derived from the complete dataset or the alternative state. The successful execution of portfolio construction using clustering is significant, as it explicitly handles the time-series jumps so that the estimated sparse inverse covariance is closer to a true reflection of the underlying, without which the benefits derived from the applied methodologies would likely be obscured by the noise from the empirical inverse covariance.

In order to further mitigate the noise in a sparse representation, we propose a statistically robust bootstrapping framework in Chapter 4, designed to eliminate redundant edges created due to the imposition of topological constraints in conventional network filtering stages. This novel methodology is used to construct diversified portfolios by selecting the least correlated assets, which are disconnected and devoid of significant edges in the correlation network. Additionally, the bootstrapping framework produces sub-networks through repetition, and the ensemble centrality measure, calculated by averaging the centralities associated with each sub-network, is more robust and can be utilized to optimize the weight in the aforementioned portfolio for further diversification. Thus, the proposed method enhances extant information filtering techniques in the network representation of multivariate time series.

Given the promising results achieved with financial datasets, we subsequently focus on broader multivariate time series modelling. With the advent of artificial intelligence, numerous time series are processed using machine learning models, especially the spatial-temporal neural network architecture which effectively processes multivariate time series as the temporal pattern of individual time series is captured by RNN or LSTM, and the interdependency among different ones is handled by convolutional operations, such as GNN, CNN. In Chapter 5, we explore the feasibility of incorporating graphical priors from the sparse network representations into the GNN within a GNN+LSTM framework. Positive outcomes suggest the viability of replacing a fully connected GNN with a sparse network topology derived

from the filtered correlation structure of underlying datasets.

The achievement with the sparse GNN representation prompts two questions: Can higher-order interactions be directly modelled in graph-based models? Can the sparse topology generated from the input dataset guide the design of neural networks? In response, we introduce the Homological Neural Network (HNN) in Chapter 6, which is a sparse MLP-like neural network structure derived from the sparse network topology of the input data, with each layer representing a simplicial level of the network. Our experiments reveal that it outperforms a fully connected MLP and is equivalent in performance to sophisticated state-of-the-art transformer-based models.

Our research underscores the wide-ranging applicability of the selection of multivariate time series, conducting experiments with a variety of data sources in each chapter. These include financial data such as equity returns from the US, UK, and China, synthesized data, as well as machine learning benchmark datasets. It has been established that networks can efficaciously encode dependency structures between multivariate time series. The proposed approach promotes the use of sparse inverse covariance in issues related to portfolio optimization and encourages the innovative design of neural networks. These findings present promising avenues for future research and exploration in the field.

7.2 Future Work

In Chapter 3, inverse covariance is effectively utilized to distinguish market regimes. However, the fundamental methodology only employs sparse inverse covariance as a distance metric, while Briola et. al. [233] propose that key market transition points can be identified through network centrality measures derived from the correlation network of assets. Consequently, the method for clustering market regimes may be significantly improved by combining it with Chapter 4, where we present a more precise estimation of centrality.

Network topological features, e.g., bootstrapped centrality in Chapter 4, have been utilized during the portfolio optimization stage. Current implementation scale

weights with centrality in an almost linear fashion, and without further additional conditions/constraints such as MVO. Combining topological features into MVO constraints would be an interesting further work and an analysis of the interaction between different conditions would be novel and valuable. Furthermore, real-life conditions could be also integrated into the optimizer, e.g., turnover, factor exposure, and trading/holding constraints.

Correlation/covariance networks presented in this thesis are all based on raw returns. In the quantitative trading field, residual returns (raw return subtract factor returns) are equivalently or something more important than the raw return as they capture the idiosyncratic nature of individual stocks. Hence, a correlation network based on residual return can be analyzed alongside or against the raw-return-based network, as it should be able to identify undefined systematic risk on top of the known factors.

Taking inspiration from Zhang et.al [234], portfolios can be optimized using a deep learning architecture specifically designed to optimize the Sharpe ratio. The two primary applications of this dissertation - portfolio optimization and neural networks - demonstrate their relevance in design and methodology by incorporating sparse representation in similarity matrices, such as inverse covariance. As such, the subsequent logical progression in research would be an integrated design combining the sparse topology of neural networks with sparse inverse covariance in portfolio optimization.

Current financial deep learning would train multiple models based on a large set of parameters and random seeds, and ensemble the models in the end. The correlation between each of the models is also certainly an interesting structure. By considering the correlation structure during the ensemble stage, a more efficient and effective ensembling can be achieved than simply aggregating them based on equal weighting or performance weighting.

In Chapter 6, the Homological Neural Network (HNN) is proposed, which represents a sparse MLP-like architecture derived from the sparse topology of the input data. Ongoing work is focusing on applying a similar philosophy of direct

modelling of higher-order interaction from the topology of input data using a convolutional architecture, designated as the Homological Convolution Neural Network (HCNN). Future endeavours will also include exploring similar topological design architecture in other categories of neural networks, such as RNNs.

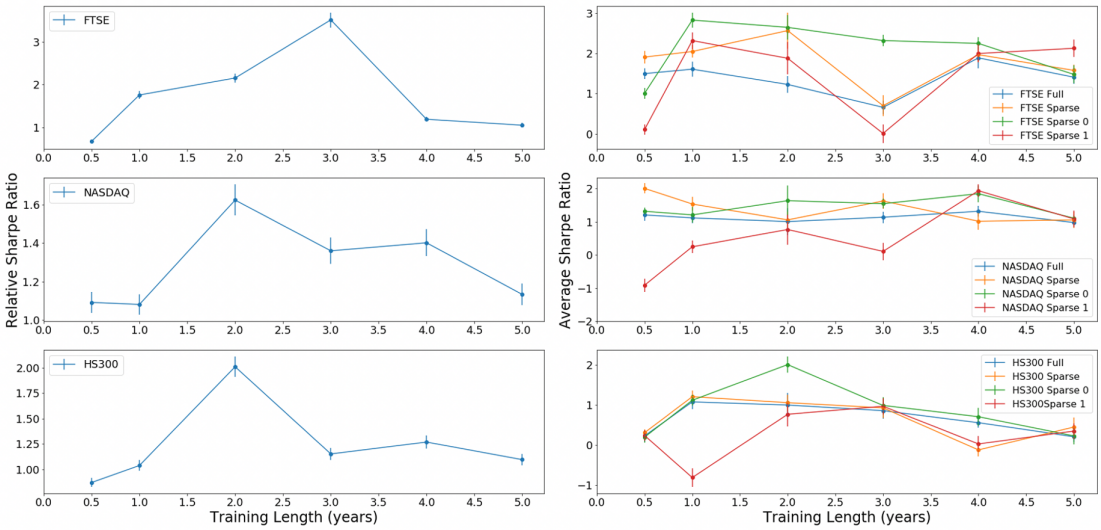
Appendix A

Appendices

A.1 Appendix 1 for Chapter 3

A.1.1 Off sample log-likelihood and performances for Student-t log-likelihood construction

In this appendix, we investigate the effect of the length of the training set on the Sharpe Ratio performance and off-sample (test set) log-likelihood using 100 randomly selected stocks drawn from NASDAQ, FTSE and HS300. They are in a similar format as Figure 3.1 and 3.2 and demonstrate that identical patterns exist regardless of underlying assets and capital markets. In Figure A.2, it is noticeable that the green bars in general sit above 0 and the red is below 0, which indicates the Sparse 0 has better off-sample log-likelihood than the Full, as illustrated in Figure 3.2. In this appendix, we compute Student-t likelihoods with $\mu = 2.1$.



(a) Relative difference in Sharpe Ratio between the Sparse 0 and the Full

(b) Sharpe Ratio of the Full, Sparse, Sparse 0 and 1

Figure A.1: Sharpe Ratio for portfolios with constituent stocks of three indices optimized using different training set durations by using Student-t log-likelihood for ICC clustering. The right subplot reports the average Sharpe Ratios (SR) with 1 standard deviation for states, statistics is on 100 training-testing periods chosen at random within the 10-year dataset. The left subplot reports instead the relative Sharpe Ratios between Sparse 0 and Full, $SR_{Sparse0}/SR_{Full}$.

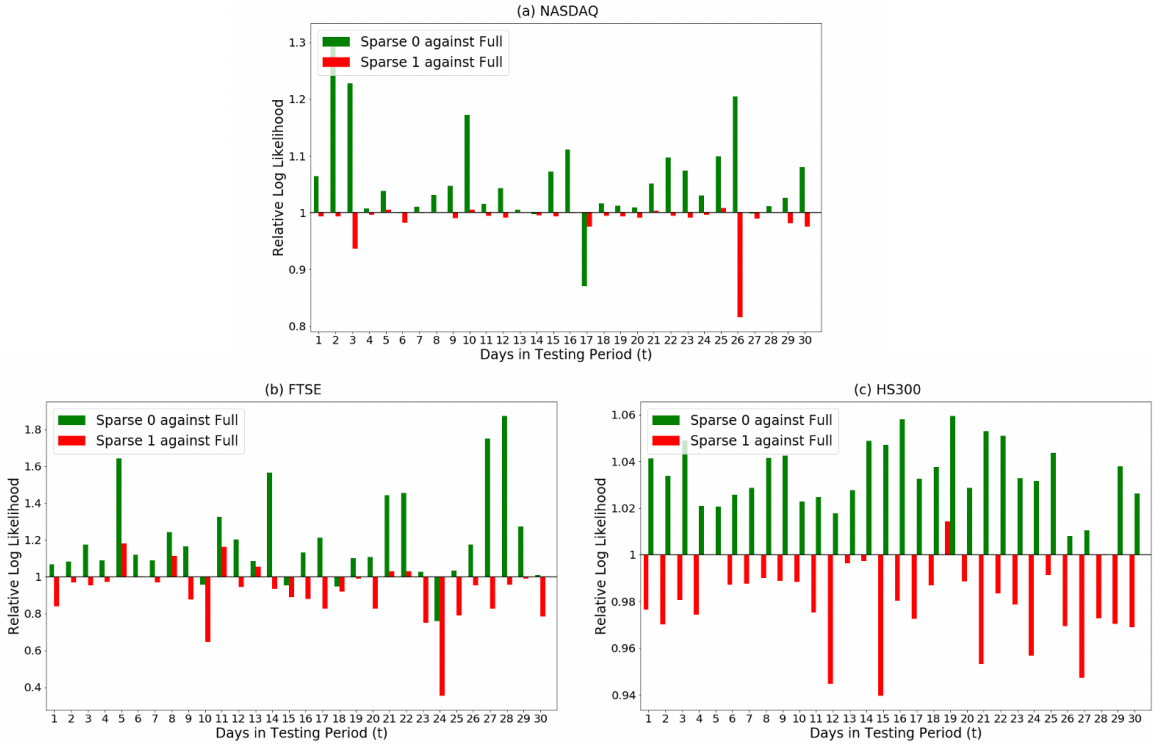


Figure A.2: Student-t log-likelihood for constituent stocks of a) NASDAQ, b) FTSE and c) HS300 Composite v.s. number of days in the test period after training. Each bar represents the average gain of the Sparse 0 (green) or 1 (red) with respect to the Full in each day. Averages are over 100 re-samplings.

A.1.2 Portfolio Performances

In this appendix, we extend the results in the main paper including 10, 20, 30 and 100-day investment horizons based on Student-t log-likelihood. Differently from the main text, portfolios are constructed with 100 random stocks drawn from NASDAQ, FTSE and HS300. They serve as complement and comparison for table 3.1. It is noticeable that although a shorter testing period yields a numerically larger Sharpe Ratio due to a possible low Volatility and an overestimation of annualized Return on a small sample, the relative difference, namely, the gain between the Sparse 0 and the Full remains roughly the same. This consistency further confirms the generality of our model. Besides, the patterns in the three tables are generally consistent with the findings in section 3.4.2.

Market	Solver	State	Return (5, 95) percentile	Volatility (5, 95) percentile	Sharpe (5, 95) percentile
NASDAQ		$1/n$ Naive	(-171,206)	(14.0,40.0)	(-8.7,12.5)
NASDAQ	SLS	Full	(-210,138)	(13.0,69.0)	(-7.2,9.1)
NASDAQ	SLS	Sparse	(-213,122)	(12.0,59.0)	(-7.6,9.1)
NASDAQ	SLS	Sparse 0	(-157,125)	(9.0,52.0)	(-5.1,10.4)
NASDAQ	SLS	Sparse 1	(-352,153)	(13.0,49.0)	(-7.1,7.6)
NASDAQ	CLA	Full	(-198,190)	(13.0,57.0)	(-6.6,10.1)
NASDAQ	CLA	Sparse	(-172,223)	(13.0,51.0)	(-8.4,10.3)
NASDAQ	CLA	Sparse 0	(-181,185)	(10.0,49.0)	(-5.2,13.4)
NASDAQ	CLA	Sparse 1	(-198,200)	(13.0,66.0)	(-8.3,7.4)
FTSE		$1/n$ Naive	(-125,140)	(9.0,28.0)	(-9.0,17.0)
FTSE	SLS	Full	(-91,147)	(7.0,27.0)	(-8.1,20.5)
FTSE	SLS	Sparse	(-89,125)	(7.0,26.0)	(-8.8,15.5)
FTSE	SLS	Sparse 0	(-64,150)	(8.0,22.0)	(-5.9,18.8)
FTSE	SLS	Sparse 1	(-113,116)	(9.0,26.0)	(-8.9,11.6)
FTSE	CLA	Full	(-147,137)	(8.0,22.0)	(-6.9,17.6)
FTSE	CLA	Sparse	(-138,122)	(7.0,25.0)	(-9.0,18.0)
FTSE	CLA	Sparse 0	(-119,129)	(7.0,23.0)	(-6.4,21.5)
FTSE	CLA	Sparse 1	(-171,149)	(9.0,22.0)	(-10.5,16.3)
HS300		$1/n$ Naive	(-228,198)	(10.0,60.0)	(-7.7,10.8)
HS300	SLS	Full	(-250,216)	(12.0,42.0)	(-8.3,15.8)
HS300	SLS	Sparse	(-283,252)	(11.0,44.0)	(-8.1,15.3)
HS300	SLS	Sparse 0	(-181,284)	(11.0,46.0)	(-6.0,16.0)
HS300	SLS	Sparse 1	(-317,192)	(13.0,58.0)	(-7.9,9.9)
HS300	CLA	Full	(-250,216)	(12.0,42.0)	(-8.3,15.8)
HS300	CLA	Sparse	(-283,252)	(11.0,44.0)	(-8.1,15.3)
HS300	CLA	Sparse 0	(-194,284)	(11.0,41.0)	(-4.9,14.2)
HS300	CLA	Sparse 1	(-277,223)	(13.0,53.0)	(-9.2,8.6)

Table A.1: Portfolio performances obtained by using Student-t log-likelihood for ICC clustering. We report annualized return, annualized volatility and annualized Sharpe Ratio computed on 10 days investment period after the 1 year training set. The values are averages and 5th and 95th percentiles computed over 10-day investment horizon from obtained from 100 re-sampling of consecutive training-investment periods chosen at random within the 10 years dataset. The underlying assets are constituent stocks of NASDAQ, FTSE and HS300. Highlight in bold are return, volatility and Sharpe Ratio indicating the optimal state in each market solver combination, while highlights in 5th return and 95th volatility showcase the extreme behaviours (excluding the state Market). The state $1/n$ Naive is the equally weighted un-optimised portfolio and it is reported as benchmark.

Market	Solver	State	Return (5, 95) percentile	Volatility (5, 95) percentile	Sharpe (5, 95) percentile
NASDAQ		$1/n$ Naive	(-112,180)	(14.0,84.0)	(-4.7,5.9)
NASDAQ	SLS	Full	(-126,131)	(14.0,85.0)	(-4.4,6.5)
NASDAQ	SLS	Sparse	(-133,127)	(13.0,53.0)	(-5.1,6.5)
NASDAQ	SLS	Sparse 0	(-124,120)	(12.0,83.0)	(-4.4,7.5)
NASDAQ	SLS	Sparse 1	(-198,78)	(15.0,50.0)	(-6.1,4.5)
NASDAQ	CLA	Full	(-147,127)	(13.0,84.0)	(-4.7,5.4)
NASDAQ	CLA	Sparse	(-149,135)	(13.0,53.0)	(-4.6,6.4)
NASDAQ	CLA	Sparse 0	(-101,144)	(12.0,46.0)	(-2.8,7.9)
NASDAQ	CLA	Sparse 1	(-187,88)	(15.0,62.0)	(-5.4,4.1)
FTSE		$1/n$ Naive	(-77,111)	(11.0,26.0)	(-4.7,8.1)
FTSE	SLS	Full	(-58,94)	(9.0,26.0)	(-4.5,11.4)
FTSE	SLS	Sparse	(-59,95)	(10.0,25.0)	(-4.9,10.6)
FTSE	SLS	Sparse 0	(-39,96)	(9.0,17.0)	(-3.4,11.4)
FTSE	SLS	Sparse 1	(-72,75)	(9.0,22.0)	(-5.4,7.5)
FTSE	CLA	Full	(-82,84)	(10.0,25.0)	(-5.6,10.3)
FTSE	CLA	Sparse	(-62,81)	(10.0,20.0)	(-6.2,10.1)
FTSE	CLA	Sparse 0	(-59,79)	(9.0,22.0)	(-4.3,11.7)
FTSE	CLA	Sparse 1	(-100,80)	(10.0,23.0)	(-6.0,9.0)
HS300		$1/n$ Naive	(-102,236)	(11.0,44.0)	(-4.4,9.3)
HS300	SLS	Full	(-133,246)	(15.0,42.0)	(-5.1,10.7)
HS300	SLS	Sparse	(-125,234)	(14.0,42.0)	(-5.0,10.3)
HS300	SLS	Sparse 0	(-101,218)	(13.0,40.0)	(-2.8,11.3)
HS300	SLS	Sparse 1	(-142,202)	(13.0,46.0)	(-5.2,7.9)
HS300	CLA	Full	(-133,246)	(15.0,42.0)	(-5.1,10.7)
HS300	CLA	Sparse	(-125,234)	(14.0,42.0)	(-5.0,10.3)
HS300	CLA	Sparse 0	(-62,247)	(12.0,41.0)	(-2.8,10.2)
HS300	CLA	Sparse 1	(-131,187)	(14.0,48.0)	(-5.0,8.2)

Table A.2: Portfolio performances obtained by using Student-t log-likelihood for ICC clustering. We report annualized return, annualized volatility and annualized Sharpe Ratio computed on 20 20-day investment period after the 1-year training set. The values are averages and 5th and 95th percentiles computed over a 20-day investment horizon obtained from 100 re-samplings of consecutive training-investment periods chosen at random within the 10-year dataset. The underlying assets are constituent stocks of NASDAQ, FTSE and HS300. Highlight in bold are return, volatility and Sharpe Ratio indicating the optimal state in each market solver combination, while highlights in the 5th return and 95th volatility showcase the extreme behaviours (excluding the state Market). The state $1/n$ Naive is the equally weighted un-optimised portfolio and it is reported as a benchmark.

Market	Solver	State	Return (5, 95) percentile	Volatility (5, 95) percentile	Sharpe (5, 95) percentile
NASDAQ		$1/n$ Naive	(-110,62)	(13.0,35.0)	(-4.5,5.7)
NASDAQ	SLS	Sparse	(-133,115)	(14.0,73.0)	(-4.7,4.7)
NASDAQ	SLS	Sparse 0	(-99,112)	(14.0,71.0)	(-2.9,6.4)
NASDAQ	SLS	Sparse 1	(-128,72)	(14.0,63.0)	(-4.4,4.4)
NASDAQ	CLA	Full	(-87,121)	(16.0,69.0)	(-3.0,5.8)
NASDAQ	CLA	Sparse	(-86,130)	(15.0,72.0)	(-2.6,6.2)
NASDAQ	CLA	Sparse 0	(-40,134)	(14.0,74.0)	(-2.5,6.5)
NASDAQ	CLA	Sparse 1	(-101,86)	(15.0,73.0)	(-3.3,4.0)
FTSE		$1/n$ Naive	(-69,90)	(11.0,28.0)	(-3.0,6.6)
FTSE	SLS	Full	(-63,80)	(10.0,26.0)	(-4.7,8.0)
FTSE	SLS	Sparse	(-56,73)	(10.0,22.0)	(-4.8,8.3)
FTSE	SLS	Sparse 0	(-52,87)	(9.0,20.0)	(-3.5,7.3)
FTSE	SLS	Sparse 1	(-68,62)	(11.0,22.0)	(-4.5,6.3)
FTSE	CLA	Full	(-56,79)	(10.0,24.0)	(-4.8,8.0)
FTSE	CLA	Sparse	(-53,73)	(10.0,20.0)	(-4.6,9.1)
FTSE	CLA	Sparse 0	(-47,72)	(9.0,20.0)	(-4.2,9.0)
FTSE	CLA	Sparse 1	(-81,64)	(11.0,24.0)	(-5.8,7.1)
HS300		$1/n$ Naive	(-90,172)	(11.0,38.0)	(-3.3,7.1)
HS300	SLS	Full	(-127,173)	(16.0,40.0)	(-4.0,6.3)
HS300	SLS	Sparse	(-98,160)	(15.0,36.0)	(-4.1,7.2)
HS300	SLS	Sparse 0	(-65,172)	(13.0,43.0)	(-2.9,7.3)
HS300	SLS	Sparse 1	(-118,142)	(15.0,45.0)	(-3.8,5.7)
HS300	CLA	Full	(-127,173)	(16.0,40.0)	(-4.0,6.2)
HS300	CLA	Sparse	(-98,160)	(15.0,36.0)	(-4.1,7.2)
HS300	CLA	Sparse 0	(-64,173)	(13.0,38.0)	(-2.7,7.4)
HS300	CLA	Sparse 1	(-98,142)	(15.0,36.0)	(-4.2,5.4)

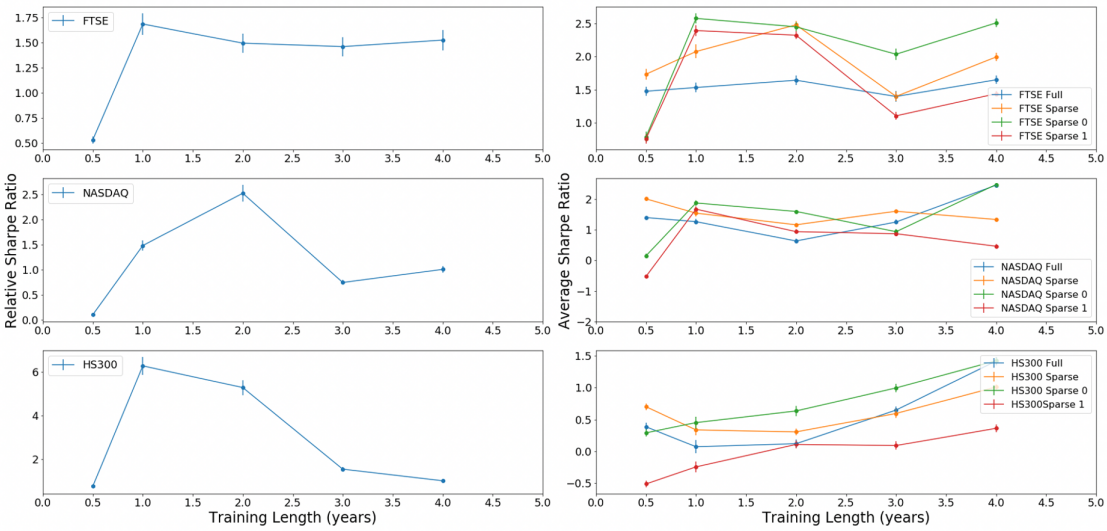
Table A.3: Portfolio performances obtained by using Student-t log-likelihood for ICC clustering. We report annualized return, annualized volatility and annualized Sharpe Ratio computed on 30 days investment period after the 1 year training set. The values are averages and 5th and 95th percentiles computed over 30-day investment horizon from obtained from 100 re-sampling of consecutive training-investment periods chosen at random within the 10 years dataset. The underlying assets are constituent stocks of NASDAQ, FTSE and HS300. Highlight in bold are return, volatility and Sharpe Ratio indicating the optimal state in each market solver combination, while highlights in 5th return and 95th volatility showcase the extreme behaviours (excluding the state Market). The state $1/n$ Naive is the equally weighted un-optimised portfolio and it is reported as benchmark.

Market	Solver	State	Return (5, 95) percentile	Volatility (5, 95) percentile	Sharpe (5, 95) percentile
NASDAQ		$1/n$ Naive	(-167,172)	(11.0,48.0)	(-7.6,8.4)
NASDAQ	SLS	Full	(-210,138)	(13.0,69.0)	(-7.2,9.1)
NASDAQ	SLS	Sparse	(-213,122)	(12.0,59.0)	(-7.6,9.1)
NASDAQ	SLS	Sparse 0	(-189,160)	(11.0,41.0)	(-5.4,8.4)
NASDAQ	SLS	Sparse 1	(-290,114)	(13.0,66.0)	(-8.5,5.6)
NASDAQ		Full	(-198,190)	(13.0,57.0)	(-6.6,10.1)
NASDAQ	CLA	Sparse	(-172,223)	(13.0,51.0)	(-8.4,10.3)
NASDAQ	CLA	Sparse 0	(-165,187)	(11.0,46.0)	(-6.1,14.2)
NASDAQ	CLA	Sparse 1	(-257,166)	(13.0,62.0)	(-6.8,6.6)
FTSE		$1/n$ Naive	(-186,140)	(8.0,22.0)	(-10.3,23.0)
FTSE	SLS	Full	(-91,147)	(7.0,27.0)	(-8.1,20.5)
FTSE	SLS	Sparse	(-89,125)	(7.0,26.0)	(-8.8,15.5)
FTSE	SLS	Sparse 0	(-75,161)	(7.0,21.0)	(-7.4,29.2)
FTSE	SLS	Sparse 1	(-110,123)	(9.0,27.0)	(-9.5,13.6)
FTSE	CLA	Full	(-147,137)	(8.0,22.0)	(-6.9,17.6)
FTSE	CLA	Sparse	(-138,122)	(7.0,25.0)	(-9.0,18.0)
FTSE	CLA	Sparse 0	(-80,145)	(8.0,23.0)	(-5.7,24.2)
FTSE	CLA	Sparse 1	(-194,138)	(9.0,25.0)	(-12.0,16.3)
HS300		$1/n$ Naive	(-228,198)	(10.0,60.0)	(-7.7,10.7)
HS300	SLS	Full	(-250,216)	(12.0,42.0)	(-8.3,15.8)
HS300	SLS	Sparse	(-283,252)	(11.0,44.0)	(-8.1,15.3)
HS300	SLS	Sparse 0	(-237,249)	(11.0,50.0)	(-5.6,16.5)
HS300	SLS	Sparse 1	(-237,193)	(14.0,45.0)	(-8.4,9.3)
HS300	CLA	Full	(-250,216)	(12.0,42.0)	(-8.3,15.8)
HS300	CLA	Sparse	(-283,252)	(11.0,44.0)	(-8.1,15.3)
HS300	CLA	Sparse 0	(-186,298)	(10.0,44.0)	(-7.6,13.9)
HS300	CLA	Sparse 1	(-302,186)	(13.0,56.0)	(-7.5,9.5)

Table A.4: Portfolio performances obtained by using Student-t log-likelihood for ICC clustering. We report annualized return, annualized volatility and annualized Sharpe Ratio computed on 100 days investment period after the 1 year training set. The values are averages and 5th and 95th percentiles computed over 100-day investment horizon from obtained from 100 re-sampling of consecutive training-investment periods chosen at random within the 10 years dataset. The underlying assets are constituent stocks of NASDAQ, FTSE and HS300. Highlight in bold are return, volatility and Sharpe Ratio indicating the optimal state in each market solver combination, while highlights in 5th return and 95th volatility showcase the extreme behaviours (excluding the state Market). The state $1/n$ Naive is the equally weighted un-optimised portfolio and it is reported as benchmark.

A.1.3 Normal Log-likelihood: training duration and Off-sample Log-likelihood

This Appendix C section includes Sharpe ratio against training duration plots and off-sample Normal log-likelihood plots of 100 random stocks drawn from NASDAQ, FTSE and HS300. They are in the similar format as Figure 3.1 and 3.2 and demonstrate that identical patterns exist regardless underlying assets and capital markets. In Figure A.2, it is noticeable that the green bars in general sit above 0 and the red are below 0, which indicates the Sparse 0 has better off-sample log-likelihood than the Full, as illustrated in Figure 3.2.



(a) Relative difference in Sharpe Ratio between the Sparse 0 and the Full

(b) Sharpe Ratio of the Full, Sparse, Sparse 0 and 1

Figure A.3: Sharpe Ratio for portfolios with constituent stocks of three indices optimized using different training set durations by using Normal log-likelihood for ICC clustering. The right subplot reports the average Sharpe Ratios (SR) with 1 standard deviation for states, statistics is on 100 training-testing periods chosen at random within the 10 years dataset. The left subplot report instead the relative Sharpe Ratios between Sparse 0 and Full, $SR_{Sparse0}/SR_{Full}$.

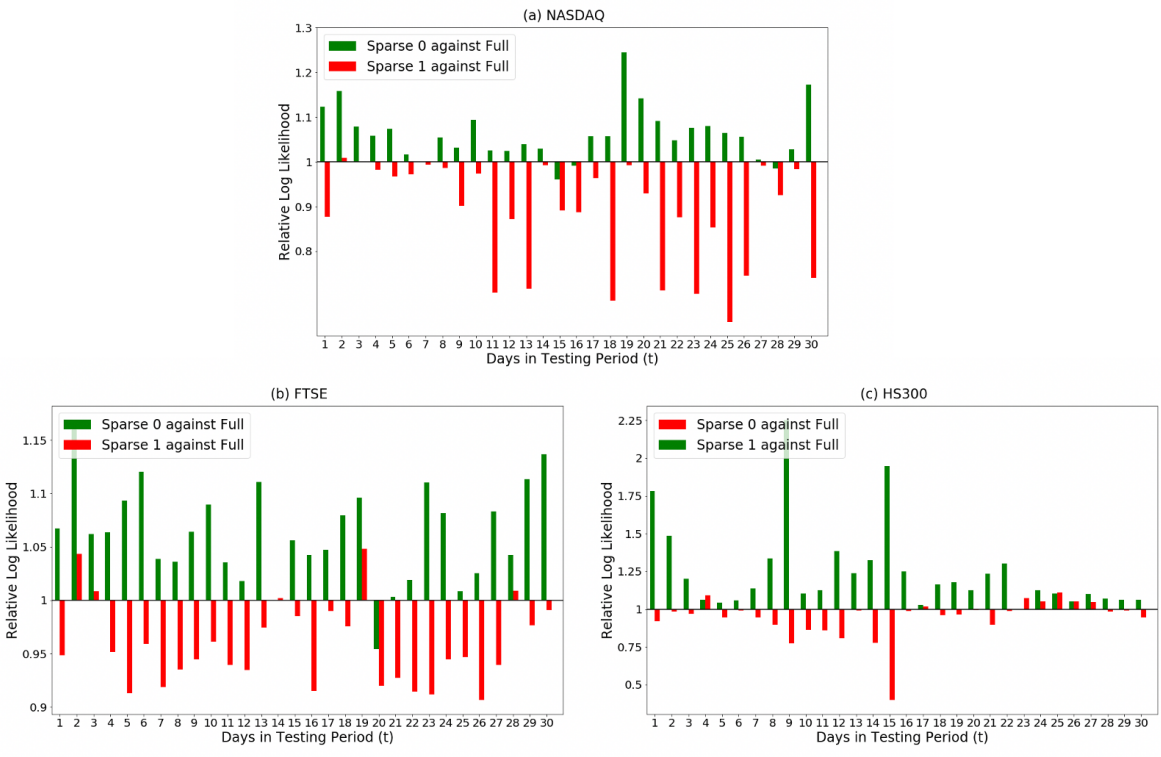


Figure A.4: Normal log-likelihood for constituent stocks of a) NASDAQ, b) FTSE and c) HS300 Composite v.s. number of days in the test period after training. Each bar represents the average gain of the Sparse 0 (green) or 1 (red) with respect to the Full in each day. Averages are over 100 re-samplings.

A.1.4 Off sample log-likelihood and performances for Normal log-likelihood construction

In this appendix we perform the same kind of investigations as in the previous appendix but ICC is computed using Normal log-likelihood. We notice similar patterns but the Student-t log-likelihood result are more significant. However, the Normal log-likelihood performs better in risk matrices.

Market	Solver	State	Return (5, 95) percentile	Volatility (5, 95) percentile	Sharpe (5, 95) percentile
NASDAQ		$1/n$ Naive	(-171,206)	(14.0,40.0)	(-8.7,12.5)
NASDAQ	SLS	Full	(-192,190)	(13.0,57.0)	(-5.4,10.1)
NASDAQ	SLS	Sparse	(-160,223)	(13.0,51.0)	(-7.4,10.3)
NASDAQ	SLS	Sparse 0	(-144,174)	(11.0,49.0)	(-5.0,12.6)
NASDAQ	SLS	Sparse 1	(-181,218)	(14.0,59.0)	(-8.7,7.8)
NASDAQ	CLA	Full	(-192,198)	(13.0,57.0)	(-5.4,10.1)
NASDAQ	CLA	Sparse	(-160,223)	(13.0,51.0)	(-7.4,10.3)
NASDAQ	CLA	Sparse 0	(-169,171)	(12.0,36.0)	(-5.0,14.2)
NASDAQ	CLA	Sparse 1	(-256,144)	(14.0,67.0)	(-6.5,6.5)
FTSE		$1/n$ Naive	(-161,117)	(7.0,34.0)	(-9.2,15.4)
FTSE	SLS	Full	(-163,116)	(8.0,28.0)	(-8.7,14.1)
FTSE	SLS	Sparse	(-148,108)	(8.0,33.0)	(-8.9,14.2)
FTSE	SLS	Sparse 0	(-111,138)	(7.0,22.0)	(-7.1,18.8)
FTSE	SLS	Sparse 1	(-199,118)	(9.0,40.0)	(-12.1,13.0)
FTSE	CLA	Full	(-163,116)	(8.0,28.0)	(-8.7,14.1)
FTSE	CLA	Sparse	(-148,108)	(8.0,33.0)	(-8.9,14.2)
FTSE	CLA	Sparse 0	(-111,146)	(7.0,21.0)	(-8.3,18.0)
FTSE	CLA	Sparse 1	(-176,123)	(8.0,36.0)	(-11.0,13.7)
HS300		$1/n$ Naive	(-228,198)	(10.0,60.0)	(-7.7,10.8)
HS300	SLS	Full	(-250,216)	(12.0,42.0)	(-8.3,15.8)
HS300	SLS	Sparse	(-283,252)	(11.0,44.0)	(-8.1,15.3)
HS300	SLS	Sparse 0	(-165,276)	(11.0,42.0)	(-5.6,15.2)
HS300	SLS	Sparse 1	(-289,176)	(13.0,44.0)	(-8.3,7.9)
HS300	CLA	Full	(-250,216)	(12.0,42.0)	(-8.3,15.8)
HS300	CLA	Sparse	(-283,252)	(11.0,44.0)	(-8.1,15.3)
HS300	CLA	Sparse 0	(-131,250)	(11.0,45.0)	(-5.3,18.8)
HS300	CLA	Sparse 1	(-331,218)	(12.0,58.0)	(-8.9,9.9)

Table A.5: Portfolio performances obtained by using Normal log-likelihood for ICC clustering. We report annualized return, annualized volatility and annualized Sharpe Ratio computed on 10 days investment period after the 1 year training set. The values are averages and 5th and 95th percentiles computed over 10-day investment horizon from obtained from 100 re-sampling of consecutive training-investment periods chosen at random within the 10 years dataset. The underlying assets are constituent stocks of NASDAQ, FTSE and HS300. Highlight in bold are return, volatility and Sharpe Ratio indicating the optimal state in each market solver combination, while highlights in 5th return and 95th volatility showcase the extreme behaviours (excluding the state Market). The state $1/n$ Naive is the equally weighted un-optimised portfolio and it is reported as benchmark.

Market	Solver	State	Return (5, 95) percentile	Volatility (5, 95) percentile	Sharpe (5, 95) percentile
NASDAQ		$1/n$ Naive	(-129,147)	(14.0,36.0)	(-4.9,7.8)
NASDAQ	SLS	Full	(-147,124)	(13.0,84.0)	(-4.7,5.4)
NASDAQ	SLS	Sparse	(-149,135)	(13.0,53.0)	(-4.6,6.4)
NASDAQ	SLS	Sparse 0	(-98,133)	(12.0,64.0)	(-3.7,8.4)
NASDAQ	SLS	Sparse 1	(-147,101)	(14.0,71.0)	(-5.6,5.3)
NASDAQ	CLA	Full	(-147,127)	(13.0,84.0)	(-4.7,5.4)
NASDAQ	CLA	Sparse	(-149,135)	(13.0,53.0)	(-4.6,6.4)
NASDAQ	CLA	Sparse 0	(-95,127)	(11.0,51.0)	(-3.1,7.8)
NASDAQ	CLA	Sparse 1	(-149,111)	(14.0,69.0)	(-4.8,5.5)
FTSE		$1/n$ Naive	(-83,104)	(10.0,32.0)	(-6.9,9.1)
FTSE	SLS	Full	(-79,100)	(9.0,30.0)	(-5.5,9.5)
FTSE	SLS	Sparse	(-63,84)	(9.0,27.0)	(-5.6,9.6)
FTSE	SLS	Sparse 0	(-49,82)	(9.0,27.0)	(-5.7,11.7)
FTSE	SLS	Sparse 1	(-92,94)	(11.0,34.0)	(-5.6,8.0)
FTSE	CLA	Full	(-79,100)	(9.0,30.0)	(-5.5,9.5)
FTSE	CLA	Sparse	(-63,84)	(9.0,27.0)	(-5.6,9.6)
FTSE	CLA	Sparse 0	(-68,82)	(9.0,23.0)	(-5.2,10.7)
FTSE	CLA	Sparse 1	(-110,101)	(9.0,29.0)	(-7.3,8.3)
HS300		$1/n$ Naive	(-102,236)	(11.0,44.0)	(-4.4,9.3)
HS300	SLS	Full	(-133,246)	(15.0,42.0)	(-5.1,10.7)
HS300	SLS	Sparse	(-125,234)	(14.0,42.0)	(-5.0,10.3)
HS300	SLS	Sparse 0	(-92,231)	(13.0,45.0)	(-3.8,11.4)
HS300	SLS	Sparse 1	(-143,204)	(14.0,43.0)	(-5.6,7.4)
HS300	CLA	Full	(-133,246)	(15.0,42.0)	(-5.1,10.7)
HS300	CLA	Sparse	(-125,234)	(14.0,42.0)	(-5.0,10.3)
HS300	CLA	Sparse 0	(-94,223)	(12.0,46.0)	(-3.0,10.0)
HS300	CLA	Sparse 1	(-146,209)	(12.0,39.0)	(-5.5,8.0)

Table A.6: Portfolio performances obtained by using Normal log-likelihood for ICC clustering. We report annualized return, annualized volatility and annualized Sharpe Ratio computed on 20 days investment period after the 1 year training set. The values are averages and 5th and 95th percentiles computed over 20-day investment horizon from obtained from 100 re-sampling of consecutive training-investment periods chosen at random within the 10 years dataset. The underlying assets are constituent stocks of NASDAQ, FTSE and HS300. Highlight in bold are return, volatility and Sharpe Ratio indicating the optimal state in each market solver combination, while highlights in 5th return and 95th volatility showcase the extreme behaviours (excluding the state Market). The state $1/n$ Naive is the equally weighted un-optimised portfolio and it is reported as benchmark.

Market	Solver	State	Return (5, 95) percentile	Volatility (5, 95) percentile	Sharpe (5, 95) percentile
NASDAQ		$1/n$ Naive	(-112,137)	(15.0,41.0)	(-3.3,7.0)
NASDAQ	SLS	Full	(-105,113)	(16.0,71.0)	(-3.0,5.0)
NASDAQ	SLS	Sparse	(-135,120)	(15.0,73.0)	(-3.2,5.5)
NASDAQ	SLS	Sparse 0	(-52,116)	(12.0,78.0)	(-2.6,5.6)
NASDAQ	SLS	Sparse 1	(-169,86)	(16.0,74.0)	(-3.9,3.3)
NASDAQ	CLA	Full	(-105,113)	(16.0,71.0)	(-3.0,5.0)
NASDAQ	CLA	Sparse	(-135,120)	(15.0,73.0)	(-3.2,5.5)
NASDAQ	CLA	Sparse 0	(-61,116)	(14.0,71.0)	(-2.5,6.2)
NASDAQ	CLA	Sparse 1	(-146,78)	(15.0,79.0)	(-4.1,3.7)
FTSE		$1/n$ Naive	(-46,68)	(11.0,31.0)	(-3.0,5.7)
FTSE	SLS	Full	(-45,75)	(11.0,26.0)	(-2.9,6.8)
FTSE	SLS	Sparse	(-48,69)	(11.0,24.0)	(-3.3,7.7)
FTSE	SLS	Sparse 0	(-32,68)	(11.0,21.0)	(-2.6,8.2)
FTSE	SLS	Sparse 1	(-69,67)	(11.0,29.0)	(-3.8,6.6)
FTSE	CLA	Full	(-45,75)	(11.0,26.0)	(-3.0,6.8)
FTSE	CLA	Sparse	(-48,69)	(11.0,24.0)	(-3.3,7.7)
FTSE	CLA	Sparse 0	(-43,67)	(11.0,22.0)	(-3.0,7.6)
FTSE	CLA	Sparse 1	(-56,58)	(12.0,29.0)	(-3.2,5.3)
HS300		$1/n$ Naive	(-91,165)	(11.0,51.0)	(-3.3,6.0)
HS300	SLS	Full	(-127,168)	(16.0,42.0)	(-4.0,6.3)
HS300	SLS	Sparse	(-94,163)	(15.0,38.0)	(-4.1,7.1)
HS300	SLS	Sparse 0	(-78,182)	(12.0,38.0)	(-3.0,6.5)
HS300	SLS	Sparse 1	(-121,135)	(14.0,55.0)	(-3.9,5.5)
HS300	CLA	Full	(-127,168)	(16.0,42.0)	(-4.0,6.3)
HS300	CLA	Sparse	(-94,163)	(15.0,38.0)	(-4.1,7.1)
HS300	CLA	Sparse 0	(-54,165)	(12.0,51.0)	(-2.3,7.8)
HS300	CLA	Sparse 1	(-110,138)	(14.0,43.0)	(-4.2,6.0)

Table A.7: Portfolio performances obtained by using Normal log-likelihood for ICC clustering. We report annualized return, annualized volatility and annualized Sharpe Ratio computed on 30 days investment period after the 1 year training set. The values are averages and 5th and 95th percentiles computed over 30-day investment horizon from obtained from 100 re-sampling of consecutive training-investment periods chosen at random within the 10 years dataset. The underlying assets are constituent stocks of NASDAQ, FTSE and HS300. Highlight in bold are return, volatility and Sharpe Ratio indicating the optimal state in each market solver combination, while highlights in 5th return and 95th volatility showcase the extreme behaviours (excluding the state Market). The state $1/n$ Naive is the equally weighted un-optimised portfolio and it is reported as benchmark.

Market	Solver	State	Return (5, 95) percentile	Volatility (5, 95) percentile	Sharpe (5, 95) percentile
NASDAQ		$1/n$ Naive	(-25,42)	(15.0,33.0)	(-1.3,2.6)
NASDAQ	SLS	Full	(-33,52)	(17.0,53.0)	(-1.4,2.8)
NASDAQ	SLS	Sparse	(-24,56)	(16.0,36.0)	(-1.2,3.1)
NASDAQ	SLS	Sparse 0	(-17,38)	(14.0,37.0)	(-0.8,2.3)
NASDAQ	SLS	Sparse 1	(-41,51)	(17.0,51.0)	(-1.9,2.1)
NASDAQ	CLA	Full	(-33,52)	(17.0,53.0)	(-1.4,2.8)
NASDAQ	CLA	Sparse	(-24,56)	(16.0,36.0)	(-1.2,3.1)
NASDAQ	CLA	Sparse 0	(-24,52)	(15.0,36.0)	(-1.2,2.4)
NASDAQ	CLA	Sparse 1	(-34,34)	(16.0,49.0)	(-1.7,1.7)
FTSE		$1/n$ Naive	(-37,49)	(11.0,30.0)	(-2.3,5.3)
FTSE	SLS	Full	(-34,55)	(11.0,26.0)	(-2.2,5.0)
FTSE	SLS	Sparse	(-38,55)	(11.0,26.0)	(-2.1,5.0)
FTSE	SLS	Sparse 0	(-27,44)	(10.0,18.0)	(-1.9,6.3)
FTSE	SLS	Sparse 1	(-39,59)	(12.0,30.0)	(-2.3,5.1)
FTSE	CLA	Full	(-34,55)	(11.0,26.0)	(-2.2,5.0)
FTSE	CLA	Sparse	(-38,55)	(11.0,26.0)	(-2.1,5.0)
FTSE	CLA	Sparse 0	(-28,57)	(11.0,18.0)	(-1.9,7.2)
FTSE	CLA	Sparse 1	(-45,44)	(11.0,34.0)	(-2.6,4.5)
HS300		$1/n$ Naive	(-47,112)	(15.0,51.0)	(-1.9,5.0)
HS300	SLS	Full	(-68,77)	(17.0,52.0)	(-2.0,3.8)
HS300	SLS	Sparse	(-50,96)	(16.0,43.0)	(-1.8,4.9)
HS300	SLS	Sparse 0	(-60,109)	(15.0,46.0)	(-1.8,5.5)
HS300	SLS	Sparse 1	(-74,114)	(17.0,48.0)	(-2.0,4.1)
HS300	CLA	Full	(-68,77)	(17.0,52.0)	(-2.0,3.8)
HS300	CLA	Sparse	(-50,96)	(16.0,43.0)	(-1.8,4.9)
HS300	CLA	Sparse 0	(-54,86)	(15.0,51.0)	(-1.4,5.3)
HS300	CLA	Sparse 1	(-71,91)	(17.0,46.0)	(-2.0,4.3)

Table A.8: Portfolio performances obtained by using Normal log-likelihood for ICC clustering. We report annualized return, annualized volatility and annualized Sharpe Ratio computed on 100 days investment period after the 1 year training set. The values are averages and 5th and 95th percentiles computed over 100-day investment horizon from obtained from 100 re-sampling of consecutive training-investment periods chosen at random within the 10 years dataset. The underlying assets are constituent stocks of NASDAQ, FTSE and HS300. Highlight in bold are return, volatility and Sharpe Ratio indicating the optimal state in each market solver combination, while highlights in 5th return and 95th volatility showcase the extreme behaviours (excluding the state Market). The state $1/n$ Naive is the equally weighted un-optimised portfolio and it is reported as benchmark.

A.1.5 Portfolio Optimization

In the original **Markowitz's mean variance optimization** approach, the portfolio weights $\mathbf{W} = (w_1, ..., w_n) \in \mathbb{R}^{1\times n}$ are chosen in order to minimize portfolio's variance $\sigma_p^2 = \mathbf{W}\Sigma\mathbf{W}^\top$ for a given value, of the portfolio's expected return $\mu\mathbf{W}^\top = \bar{r}_p$. Specifically,

$$\begin{aligned} \mathbf{W}^* = \min_{\mathbf{W}} \quad & \mathbf{W}\Sigma\mathbf{W}^\top \\ \text{s.t} \quad & \mathbb{1}\mathbf{W}^\top = 1, \\ \text{and} \quad & \mu\mathbf{W}^\top = \bar{r}_p, \end{aligned} \tag{A.1}$$

The exact solution can be obtained analytically by setting to zero the derivatives with respect to $\mathbf{W}$, using the Lagrange multiplier technique to account for the constraints. Namely the minimum of the following Lagrangian is computed

$$L(\mathbf{W}, \lambda) = \mathbf{W}\Sigma\mathbf{W}^\top + \lambda_1\mu\mathbf{W}^\top + \lambda_2\mathbb{1}\mathbf{W}^\top, \tag{A.2}$$

and the solution is

$$\mathbf{W}^* = \Sigma^{-1}(\lambda_1\mu + \lambda_2\mathbb{1})^\top, \tag{A.3}$$

where λ_1 and λ_2 are the Lagrange multipliers.

The **sequential least square quadratic programming (SLS)** [185, 186, 187] is considered to be one of the most efficient computational method to solve general nonlinear constrained optimization problems. Jackson et al. and Cesarone et al. demonstrate its effectiveness in finance [188, 189]. SLS solves the optimization problem iteratively with a gradient descent strategy starting with an initial setting $\mathbf{W}^0$, and updating $\mathbf{W}^{k+1}$ from $\mathbf{W}^k$ by:

$$\mathbf{W}^{k+1} = \mathbf{W}^k + \alpha^k\mathbf{d}^k \tag{A.4}$$

where $\mathbf{d}^k$ is the search direction at the k-th step and α^k is the associated step size. In each iteration, the descent search direction, $\mathbf{d}$, is determined by the solution of a

sub-problem. Given the loss function

$$f(\mathbf{W}) = \mathbf{W}\Sigma\mathbf{W}^{\top} \tag{A.5}$$

that we want to minimize under a set of non-liner constraints $g_j(\mathbf{W}) = 0$ for $j \in [1, m_e]$ and $g_j(\mathbf{W}) \geq 0$ for $j \in [m_e + 1, m]$, at each iteration, the problem of finding the optimal descent direction can be addressed by solving the standard quadratic programming sub-problem [235]:

$$\begin{aligned} \mathbf{d}^{k+1} = \min_{\mathbf{d}} \quad & \frac{1}{2}\mathbf{d}\nabla^2 L(\mathbf{W}^k, \lambda)\mathbf{d}^{\top} + \nabla f(\mathbf{W}^k)\mathbf{d}^{\top} \\ \text{s.t} \quad & \nabla g_j(\mathbf{W}^k)\mathbf{d}^{\top} + g_j(\mathbf{W}^k) = 0, \quad j = 1, ..., m_e \\ & \nabla g_j(\mathbf{W}^k)\mathbf{d}^{\top} + g_j(\mathbf{W}^k) \geq 0, \quad j = m_e + 1, ..., m \end{aligned} \tag{A.6}$$

where $L(\mathbf{W}, \lambda)$ is the associated Lagrangian

$$L(\mathbf{W}, \lambda) = f(\mathbf{W}) - \sum_{j=1}^{m} \lambda_j g_j(\mathbf{W}). \tag{A.7}$$

A step size $\alpha = 1$ is optimal near a local optimum, but when far from the optimum, the step size will need to be modified to guarantee a global convergence. Han [236], Powell [237], Schittkowski [238] and Rockafellar [239] have introduced the use of penalty functions in the nonlinear programming to control the step size.

The **Critical Line Algorithm (CLA)** is an efficient alternative to the quadratic optimizer for mean-variance model, as it is specifically designed for inequality portfolio optimization. It was already originally introduced in the Markowitz Portfolio Selection paper [1], and its computational implementation has become increasingly popular [190, 191]. CLA also solves constrained problems with conditions in inequalities, but unlike SLS, it divides a constrained problem into series of unconstrained sub-problems by invoking the concept of turning point. A turning point is a constrained minimum variance portfolio whose vicinity contains other constrained minimum variance portfolios of different free assets.

Similar to quadratic programming, an initial solution is required on the con-

strained minimum variance frontier. To construct the initial solution, assets are ranked with respect to their expected returns. Then, one increases the weight of the first asset of the highest expected returns, w_1, from a defined lower bound $l_1 = 0$ to an upper bound u_1 if $w_1 \leq 1$. Subsequently, the following assets have their weights increased until $\sum_i w_i = 1$. Typically, the weights of the first and the last few assets are set to the upper and lower bound which are called bounded assets, while only one in the middle has its weight between bounds and referred as the free asset. The free weight is expressed as:

$$w_f = 1 - \sum_{i \in \mathbb{U}} w_i - \sum_{i \in \mathbb{L}} w_i \tag{A.8}$$

where $\mathbb{U}$ and $\mathbb{L}$ represents two sets of upper and lower bounded weights. Then in the following iterations, by decreasing the Lagrange multiplier for the constraint on expected portfolio return, λ to move to the next lower turning point, two cases need to be considered to compute $\mathbf{W}$. A formally free asset moves to its bound, or vice versa, a bounded asset wants to become free. In both situations, the maximum threshold λ_{inside} and $\lambda_{outside}$ for the former and the later will be found. Subsequently, the larger one characterises the new turning point, and the asset is moved accordingly, and weights are re-assigned. As the free and bounded assets do not interchange between turning points, the constrained solution between two turning points is in fact the solution of unconstrained optimization on only the free assets. Therefore, the constrained problem reduces to solving the unconstrained problem on the free assets. When no new threshold can be found, the lowest turning point is said to be reached and the algorithm is terminated for the optimized $\mathbf{W}$.

A.2 Appendix 2 for Chapter 5

Dataset	No. Features	No. Samples	Sample Rate
solar-energy	137	52560	10 minutes
exchange-rates	8	7588	1 day

Table A.9: Multivariate time-series dataset statistics, including the number of features, number of samples and sample rate in the solar-energy-energy and exchange-rates-rates datasets [139].

	solar-energy	exchange-rates
LSTM-MLP	452901	13011
LSTM-MLP-HNN	509208	13203
LSTM-MLP-res	509208	13203
LSTM-HNN	239061	12795

Table A.10: Number of parameters in each model in solar-energy and exchange-rates datasets, comparing the sparse LSTM-HNN with the fully connected models.

	solar-energy	exchange-rates
LSTM-skip	337112	19478
TPA-LSTM	613987	132172
MTGNN	347665	337345
LSTM-HNN	239061	12795

Table A.11: Number of parameters in each model in solar-energy and exchange-rates datasets, comparing the LSTM-HNN with respect to state-of-art models.

Bibliography

[1] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1), 1952.

[2] Francesca Pozzi, Tiziana Di Matteo, and Tomaso Aste. Spread of risk across financial markets: better to invest in the peripheries. *Scientific Reports*, 3, 2013.

[3] P. J. Young and Stephen Shellswell. Time series analysis, forecasting and control. *IEEE Transactions on Automatic Control*, 17:281–283, 1972.

[4] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015.

[5] Liheng Zhang, Charu C. Aggarwal, and Guo-Jun Qi. Stock price prediction via discovering multi-frequency trading patterns. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[6] Christian Kascha. A comparison of estimation methods for vector autoregressive moving-average models. *Econometric Reviews*, 31:297 – 324, 2012.

[7] Theodore W. Anderson. Maximum likelihood estimation for vector autoregressive moving average models. 1978.

[8] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018.

[9] Shun-Yao Shih, Fan-Keng Sun, and Hung yi Lee. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, pages 1–21, 2019.

[10] Syama Sundar Rangapuram, Matthias W. Seeger, Jan Gasthaus, Lorenzo Stella, Bernie Wang, and Tim Januschowski. Deep state space models for time series forecasting. In *NeurIPS*, 2018.

[11] Rajat Sen, Hsiang-Fu Yu, and Inderjit S. Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. In *NeurIPS*, 2019.

[12] Nicolai Meinshausen and Peter Buhlmann. High-dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34:1436–1462, 2006.

[13] Ming Yuan and Yi Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94:19–35, 2007.

[14] Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation. *ArXiv*, abs/0707.0704, 2007.

[15] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, 9 3:432–41, 2008.

[16] Rahul Mazumder and Trevor J. Hastie. The Graphical Lasso: New insights and alternatives. *Electronic journal of statistics*, 6:2125–2149, 2012.

[17] Wolfram Barfuss, Guido Previde Massara, T. Di Matteo, and Tomaso Aste. Parsimonious modeling with information filtering networks. *Physical Review E*, 94(6), Dec 2016.

[18] Jaroslav Nešetřil, Eva Milková, and Helena Nešetřilová. Otakar Boruvka on minimum spanning tree problem translation of both the 1926 papers, comments, history. *Discrete mathematics*, 233(1-3):3–36, 2001.

[19] Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. 1956.

[20] Robert C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[21] M. Tumminello, T. Aste, T. Di Matteo, and R. N. Mantegna. A tool for filtering information in complex systems. *Proceedings of the National Academy of Sciences*, 102(30):10421–10426, 2005.

[22] T. Aste and T. Matteo. Sparse causality network retrieval from short time series. *Complex.*, 2017:4518429:1–4518429:13, 2017.

[23] Guido Previde Massara, T. Matteo, and T. Aste. Network filtering for big data: Triangulated maximally filtered graph. *ArXiv*, abs/1505.02445, 2017.

[24] Guido Previde Massara and Tomaso Aste. Learning clique forests. *ArXiv*, 1905.02266, 2019.

[25] Qawi K. Telesford, S. Simpson, J. Burdette, S. Hayasaka, and P. Laurienti. The brain as a complex system: Using network science as a tool for understanding the brain. *Brain connectivity*, 1 (4):295–308, 2011.

[26] Tomaso Aste. Topological regularization with information filtering networks. *arXiv preprint arXiv:2005.04692*, 2020.

[27] Guido Previde Massara and Tomaso Aste. Learning clique forests. *ArXiv*, abs/1905.02266, 2019.

[28] Olivier Ledoit and Michael Wolf. Honey, I shrunk the sample covariance matrix. *Capital Markets: Asset Pricing & Valuation*, 2003.

[29] Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88:365–411, 2004.

[30] Yilun Chen, Ami Wiesel, Yonina C. Eldar, and Alfred O. Hero. Shrinkage algorithms for MMSE covariance estimation. *IEEE Transactions on Signal Processing*, 58:5016–5029, 2010.

[31] Laurent Laloux, Pierre Cizeau, Marc Potters, and Jean-Philippe Bouchaud. Random matrix theory and financial correlations. *International Journal of Theoretical and Applied Finance*, 03:391–397, 2000.

[32] Noureddine El Karoui. Spectrum estimation for large dimensional covariance matrices using random matrix theory. *Annals of Statistics*, 36:2757–2790, 2006.

[33] Szilárd Pafka, Marc Potters, and Imre Kondor. Exponential weighting and random-matrix-theory-based filtering of financial covariance matrices for portfolio optimization. *Research Papers in Economics*, 2004.

[34] Responsible Officer. The distribution of stock returns. *Journal of the American Statistical Association*, 67:807–812, 1972.

[35] Eckhard Limpert and Werner A. Stahel. Problems with using the normal distribution – and ways to improve quality and efficiency of data analysis. *PLoS ONE*, 6, 2011.

[36] Uwe Küchler, Kirsten Neumann, Michael Sørensen, and Arnfried Streller. Stock returns and hyperbolic distributions. *Mathematical and Computer Modelling*, 29:1–15, 1999.

[37] Student. On the probable error of the mean. *Biometrika*, 6:1–25, 1908.

[38] Amado Peiró. The distribution of stock returns: international evidence. *Applied Financial Economics*, 4(6):431–439, 1994.

[39] Eckhard Platen and Renata Rendek. Empirical evidence on Student-t Log-Returns of diversified world stock indices. *Journal of Statistical Theory and Practice*, 2(2):233–251, 2008.

[40] E. B. Wilson. First and second laws of error. *Journal of the American Statistical Association*, 18:841–851, 1923.

[41] R. M. Norton. The double exponential distribution: Using calculus to find a maximum likelihood estimator. *The American Statistician*, 38:135–136, 1984.

[42] T. Eltoft, Taesu Kim, and Te-Won Lee. On the multivariate Laplace distribution. *IEEE Signal Processing Letters*, 13:300–303, 2006.

[43] B. Mandelbrot. The PARETO-LEVY law and the distribution of income*. *International Economic Review*, 1:79, 1960.

[44] H. F. Coronel-Brizio and A. Hernández-Montoya. On fitting the Pareto-Levy distribution to stock market index data: selecting a suitable cutoff value. *Physica A-statistical Mechanics and Its Applications*, 354:437–449, 2005.

[45] A. Stuart and H. Markowitz. Portfolio selection: Efficient diversification of investments. *A Quarterly Journal of Operations Research*, 10:253, 1959.

[46] Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the royal statistical society series b-methodological*, 58:267–288, 1996.

[47] Jerome H. Friedman, Trevor J. Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, 9 3:432–41, 2008.

[48] T. Carsey and Jeffrey J. Harden. Monte carlo simulation and resampling methods for social science. CSAGE Publications, Inc., 2013.

[49] B. Manly. Randomization, bootstrap and monte carlo methods in biology. Chapman and Hall/CRC, 2020.

[50] Philip G. Berger and E. Ofek. Diversification's effect on firm value. *Journal of Financial Economics*, 37:39–65, 1995.

[51] Victor DeMiguel, Lorenzo Garlappi, and R. Uppal. Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *Review of Financial Studies*, 22:1915–1953, 2009.

[52] Jonas Schmitt. Portfolio selection efficient diversification of investments. 2016.

[53] James D. Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57(2):357–384, 1989.

[54] L. Ramchand and R. Susmel. Cross correlations across major international markets. *Journal of Empirical Finance*, 5(4):397–416, 1998.

[55] Andrew Ang and Geert Bekaert. How do regimes affect asset allocation? Working Paper 10080, National Bureau of Economic Research, November 2003.

[56] Steven L. Scott. Bayesian methods for hidden markov models. *Journal of the American Statistical Association*, 97:337 – 351, 2002.

[57] Tobias Rydén. Em versus markov chain monte carlo for estimation of hidden markov models: a computational perspective. *Bayesian Analysis*, 3:659–688, 2008.

[58] Randal Douc, Aurélien Garivier, Éric Moulines, and Jimmy Olsson. Sequential monte carlo smoothing for general state space hidden markov models. *Annals of Applied Probability*, 21:2109–2145, 2011.

[59] Emily B. Fox, Erik B. Sudderth, Michael I. Jordan, and Alan S. Willsky. An hdp-hmm for systems with state persistence. In *International Conference on Machine Learning*, 2008.

[60] D. Reynolds. Gaussian mixture models. In *Encyclopedia of Biometrics*, 2009.

[61] I. Buckley, D. Saunders, and L. Seco. Portfolio optimization when asset returns have the Gaussian mixture distribution. *Eur. J. Oper. Res.*, 185:1434–1461, 2008.

[62] Wolfgang Ketter, John Collins, Maria L. Gini, Alok Gupta, and P. Schrater. Detecting and forecasting economic regimes in multi-agent automated exchanges. *Econometrics eJournal*, 2009.

[63] Sarah Jane Delany. k-nearest neighbour classifiers. 2007.

[64] R. Nayak, Debahuti Mishra, and A. Rath. A naïve SVM-KNN based stock market trend reversal analysis for indian benchmark indices. *Appl. Soft Comput.*, 35:670–680, 2015.

[65] I. Kumar, Kiran Dogra, Chetna Utreja, and Premlata Yadav. A comparative study of supervised machine learning algorithms for stock market trend prediction. *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 1003–1007, 2018.

[66] Eric M. Reyes and S. Ghosh. Bayesian average error-based approach to sample size calculations for hypothesis testing. *Journal of Biopharmaceutical Statistics*, 23:569 – 588, 2013.

[67] S. Hee, T. Hamborg, S. Day, J. Madan, F. Miller, M. Posch, S. Zohar, and N. Stallard. Decision-theoretic designs for small trials and pilot studies: A review. *Statistical Methods in Medical Research*, 25:1022 – 1038, 2016.

[68] S. Berry, B. Carlin, Jiun-Kae Jack Lee, and P. Müller. Bayesian adaptive methods for clinical trials. 2010.

[69] Fischer Black and Robert B Litterman. Asset allocation. *The Journal of Fixed Income*, 1(2):7–18, 1991.

[70] Fischer Black and Robert Litterman. Global portfolio optimization. *Financial Analysts Journal*, 48(5):28–43, 1992.

[71] S. Aghabozorgi, A. S. Shirkhorshidi, and Teh Ying Wah. Time-series clustering - a decade review. *Inf. Syst.*, 53:16–38, 2015.

[72] S. Zolhavarieh, S. Aghabozorgi, and Ying Wah Teh. A review of subsequence time series clustering. *The Scientific World Journal*, 2014, 2014.

[73] J. Navarro, C. Frenk, and S. White. A universal density profile from hierarchical clustering. *The Astrophysical Journal*, 490:493–508, 1997.

[74] Biplab Bhattacharjee, M. Shafi, and A. Acharjee. Network mining based elucidation of the dynamics of cross-market clustering and connectedness in asian region: An mst and hierarchical clustering approach. *J. King Saud Univ. Comput. Inf. Sci.*, 31:218–228, 2019.

[75] Peter J. Zeitsch. A jump model for credit default swaps with hierarchical clustering. *Physica A-statistical Mechanics and Its Applications*, 524:737–775, 2019.

[76] T. Madhulatha. An overview on clustering methods. *ArXiv*, abs/1205.1117, 2012.

[77] S. Dolnicar. A review of unquestioned standards in using cluster analysis for data-driven market segmentation. 2002.

[78] Ricardo J. G. B. Campello, Davoud Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *PAKDD*, 2013.

[79] Chang Liu and Yang Cao. Task re-pricing model based on density-based spatial clustering of applications. *Appl. Soft Comput.*, 96:106608, 2020.

[80] Smail Tigani, H. Chaibi, and Rachid Saadane. Gaussian mixture and kernel density-based hybrid model for volatility behavior extraction from public financial data. *Data*, 4:19, 2019.

[81] M. Aitken, P. Brown, Christine Buckland, H. Izan, and Terry S. Walter. Price clustering on the australian stock exchange. *Pacific-basin Finance Journal*, 4:297–314, 1995.

[82] Hee-Joon Ahn, Jun Cai, and Yan-Leung Cheung. Price clustering on the limit-order book: Evidence from the stock exchange of hong kong. *Journal of Financial Markets*, 8:421–451, 2005.

[83] Fang Li, H. Sheng, and Dongmo Zhang. Event pattern discovery from the stock market bulletin. In *Discovery Science*, 2002.

[84] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:881–892, 2002.

[85] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17:790–799, 1995.

[86] Khalid Alkhatib, Hassan Najadat, Ismail Hmeidi, and M. Shatnawi. Stock price prediction using k-nearest neighbor (knn) algorithm. 2013.

[87] D. Duffie and Jun Pan. An overview of value at risk. 1997.

[88] Wen-Ling Lin, R. Engle, and Takatoshi Ito. Do bulls and bears move across borders? international transmission of stock returns and volatility. *Review of Financial Studies*, 7:507–538, 1994.

[89] Andrew Ang and G. Bekaert. International asset allocation with regime shifts. *Review of Financial Studies*, 15:1137–1187, 2002.

[90] Nicolò Musmeci, T. Aste, and T. D. Matteo. What does past correlation structure tell us about the future? an answer from network filtering. *arXiv: Portfolio Management*, 2016.

[91] Tim Bollerslev. Modelling the coherence in short-run nominal exchange rates: A multivariate generalized arch model. *The Review of Economics and Statistics*, 72(3):498–505, 1990.

[92] Robert Engle. Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models. *Journal of Business Economic Statistics*, 20(3):339–350, 2002.

[93] David Hallac, Sagar Vare, Stephen Boyd, and Jure Leskovec. Toeplitz inverse covariance-based clustering of multivariate time series data. *arXiv*, 1706.03161, 2018.

[94] Pier Francesco Procacci and T. Aste. Forecasting market states. *Quantitative Finance*, 19:1491 – 1498, 2018.

[95] Paul A. Samuelson. Lifetime portfolio selection by dynamic stochastic programming. *The Review of Economics and Statistics*, 51(3):239–246, 1969.

[96] Nils H. Hakansson. Capital growth and the mean-variance approach to portfolio selection. *Journal of Financial and Quantitative Analysis*, 6(1):517–557, 1971.

[97] Robert R. Grauer and Nils H. Hakansson. On the use of mean-variance and quadratic approximations in implementing dynamic investment strategies: A comparison of returns and investment policies. *Management Science*, 39(7):856–871, 1993.

[98] Stanley R. Pliska. *Introduction to mathematical finance discrete time models*. Malden, Mass Blackwell, 1997. Includes bibliographical references (p. [254]-256) and index.

[99] Robert C. Merton. Lifetime portfolio selection under uncertainty: The continuous-time case. *The Review of Economics and Statistics*, 51(3):247–257, 1969.

[100] I. Karatzas, J. Lehoczky, and S. Shreve. Optimal portfolio and consumption decisions for a "small investor" on a finite horizon. *Siam Journal on Control and Optimization*, 25:1557–1586, 1987.

[101] Robert C Merton. Optimum consumption and portfolio rules in a continuous-time model. *Journal of Economic Theory*, 3(4):374–413, 1971.

[102] Isabelle Bajeux-Besnainou, James V. Jordan, and Roland Portait. An asset allocation puzzle: Comment. *American Economic Review*, 91(4):1170–1179, September 2001.

[103] R. C. Merton. Lifetime portfolio selection under uncertainty: The continuous-time case. *The Review of Economics and Statistics*, 51:247–257, 1969.

[104] J. Cox and Chi-Fu Huang. Optimal consumption and portfolio policies when asset prices follow a diffusion process. *Journal of Economic Theory*, 49:33–83, 1989.

[105] Tomas Björk. Arbitrage theory in continuous time. Oxford University Press, 2019.

[106] Y. Jiao and H. Pham. Optimal investment with counterparty risk: a default-density model approach. *Finance and Stochastics*, 15:725–753, 2011.

[107] I. Karatzas. Optimization problems in the theory of continuous trading. *Siam Journal on Control and Optimization*, 27:1221–1259, 1989.

[108] M. Brennan and Yihong Xia. Dynamic asset allocation under inflation. *Journal of Finance*, 57:1201–1238, 2000.

[109] T. Bielecki and Inwon Jang. Portfolio optimization with a defaultable security. *Asia-Pacific Financial Markets*, 13:113–127, 2006.

[110] Y. Jiao, Idris Kharroubi, and H. Pham. Optimal investment under multiple defaults risk: a bsde-decomposition approach. *Annals of Applied Probability*, 23:455–491, 2013.

[111] T. Zariphopoulou. A solution approach to valuation with unhedgeable risks. *Finance and Stochastics*, 5:61–82, 2001.

[112] N. Castañeda-Leyva and D. Hernández-Hernández. Optimal consumption-investment problems in incomplete markets with stochastic coefficients. *SIAM J. Control. Optim.*, 44:1322–1344, 2005.

[113] Hua He and Neil D. Pearson. Consumption and portfolio policies with incomplete markets and short-sale constraints: The infinite dimensional case. *Journal of Economic Theory*, 54:259–304, 1991.

[114] I. Karatzas, J. Lehoczky, S. Shreve, and Ganlin Xu. Martingale and duality methods for utility maximization in a incomplete market. *Siam Journal on Control and Optimization*, 29:702–730, 1991.

[115] Ming-Xia Li, Zhi-Qiang Jiang, Wen-Jie Xie, Xiong Xiong, Wei Zhang, and Wei-Xing Zhou. Unveiling correlations between financial variables and topological metrics of trading networks: Evidence from a stock and its warrant. *Physica A-statistical Mechanics and Its Applications*, 419:575–584, 2013.

[116] Angeliki Papana, Catherine Kyrtsou, Dimitris Kugiumtzis, and C. G. H. Diks. Financial networks based on granger causality: A case study. *Physica A-statistical Mechanics and Its Applications*, 482:65–73, 2017.

[117] Xiaoqian Sun, Huawei Shen, and Xue qi Cheng. Trading network predicts stock price. *Scientific Reports*, 4, 2014.

[118] Ahmet Sensoy and Benjamin Miranda Tabak. Dynamic spanning trees in stock market networks: The case of asia-pacific. *Physica A-statistical Mechanics and Its Applications*, 414:387–402, 2014.

[119] Gangjin Wang and Chi Xie. Tail dependence structure of the foreign exchange market: A network view. *Expert Syst. Appl.*, 46:164–179, 2016.

[120] Fenghua Wen, Xin Yang, and Wei-Xing Zhou. Tail dependence networks of global stock markets. *International Journal of Finance & Economics*, 2018.

[121] Yan Li, Xiongfei Jiang, Yue Tian, Sai-Ping Li, and Bo Zheng. Portfolio optimization based on network topology. *Physica A: Statistical Mechanics and its Applications*, 2019.

[122] Gustavo Peralta and Abalfazl Zareei. A network approach to portfolio selection. *Econometric Modeling: Capital Markets - Portfolio Theory eJournal*, 2016.

[123] Gian Paolo Clemente, Rosanna Grassi, and Asmerilda Hitaj. Asset allocation: new evidence through network approaches. *Annals of Operations Research*, pages 1–20, 2018.

[124] Gueorgui S. Konstantinov, Andreas Chorus, and Jonas Rebmann. A network and machine learning approach to factor, asset, and blended allocation. 2020.

[125] Harald Lohre, Jochen Papenbrock, and Muddit Poonia. The use of correlation networks in parametric portfolio policies. *Econometric Modeling: Capital Markets - Risk eJournal*, 2014.

[126] Kenneth Robinson Ahern. Network centrality and the cross section of stock returns. *ERN: Asset Pricing Models (Topic)*, 2013.

[127] Bruno Scalzo Dees, L. Stanković, Anthony G. Constantinides, and Danilo P. Mandic. Portfolio cuts: A graph-theoretic framework to diversification. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8454–8458, 2019.

[128] Marcos M. Lopez de Prado. Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management*, 42:59 – 69, 2016.

[129] Fei Ren, Yansong Lu, Sai-Ping Li, Xiongfei Jiang, Li-Xin Zhong, and Tian Qiu. Dynamic portfolio strategy using clustering approach. *PLoS ONE*, 12, 2016.

[130] Prasanna Gai and Sujit Kapadia. Contagion in financial networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 466:2401 – 2423, 2010.

[131] Eric Zivot and Jiahui Wang. Vector autoregressive models for multivariate time series. 2003.

[132] Stephen J. Roberts, Michael A. Osborne, Mark Ebden, Steven Reece, Neale P. Gibson, and Suzanne Aigrain. Gaussian processes for timeseries modelling. 2012.

[133] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2017.

[134] Shun-Yao Shih, Fan-Keng Sun, and Hung yi Lee. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, pages 1–21, 2018.

[135] Wenjie Lu, Jiazheng Li, Yifan Li, Aijun Sun, and Jingyang Wang. A cnn-lstm-based model to forecast stock prices. *Complex.*, 2020:6622927:1–6622927:10, 2020.

[136] Renzhuo Wan, Shuping Mei, Jun Wang, Min Liu, and F. Yang. Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting. *Electronics*, 2019.

[137] Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

[138] Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S. Jensen. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *ArXiv*, abs/2206.09112, 2022.

[139] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[140] Yuanrong Wang and Tomaso Aste. Network filtering of spatial-temporal gnn for multivariate time-series prediction. *Proceedings of the Third ACM International Conference on AI in Finance*, 2022.

[141] Daniele Calandriello, Ioannis Koutis, Alessandro Lazaric, and Michal Valko. Improved large-scale graph learning through ridge spectral sparsification. In *ICML*, 2018.

[142] Alireza Chakeri, Hamidreza Farhidzadeh, and Lawrence O. Hall. Spectral sparsification in spectral clustering. *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2301–2306, 2016.

[143] Yu Rong, Wen bing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *ICLR*, 2020.

[144] Arman Hasanzadeh, Ehsan Hajiramezanali, Shahin Boluki, Mingyuan Zhou, Nick G. Duffield, Krishna R. Narayanan, and Xiaoning Qian. Bayesian graph neural networks with adaptive connection sampling. *ArXiv*, abs/2006.04064, 2020.

[145] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. *ArXiv*, abs/1911.08415, 2020.

[146] Dongsheng Luo, Wei Cheng, Wenchao Yu, Bo Zong, Jingchao Ni, Haifeng Chen, and Xiang Zhang. Learning to drop: Robust graph neural network via topological denoising. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021.

[147] Dongkwan Kim and Alice H. Oh. How to find your friendly neighborhood: Graph attention design with self-supervision. In *ICLR*, 2021.

[148] Jianbo Ye, Xin Lu, Zhe L. Lin, and James Zijun Wang. Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers. *ArXiv*, abs/1802.00124, 2018.

[149] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient transfer learning. *ArXiv*, abs/1611.06440, 2016.

[150] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Snip: Single-shot network pruning based on connection sensitivity. *ArXiv*, abs/1810.02340, 2018.

[151] Ruichi Yu, Ang Li, Chun-Fu Chen, Jui-Hsin Lai, Vlad I. Morariu, Xintong Han, Mingfei Gao, Ching-Yung Lin, and Larry S. Davis. Nisp: Pruning networks using neuron importance score propagation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9194–9203, 2017.

[152] Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13:1 – 18, 2015.

[153] Dmitry Molchanov, Arsenii Ashukha, and Dmitry P. Vetrov. Variational dropout sparsifies deep neural networks. *ArXiv*, abs/1701.05369, 2017.

[154] Huiyuan Zhuo, Xuelin Qian, Yanwei Fu, Heng Yang, and X. Xue. Scsp: Spectral clustering filter pruning with soft self-adaption manners. *ArXiv*, abs/1806.05320, 2018.

[155] Dong Wang, Lei Zhou, Xueni Zhang, Xiao Bai, and Jun Zhou. Exploring linear relationship in feature map subspace for convnets compression. *ArXiv*, abs/1803.05729, 2018.

[156] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Helen Li. Learning structured sparsity in deep neural networks. *ArXiv*, abs/1608.03665, 2016.

[157] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall F. Tappen, and Marianna Pensky. Sparse convolutional neural networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 806–814, 2015.

[158] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2755–2763, 2017.

[159] Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *ArXiv*, abs/1607.03250, 2016.

[160] Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jin-Hui Zhu. Discrimination-aware channel pruning for deep neural networks. In *Neural Information Processing Systems*, 2018.

[161] Hanyu Peng, Jiaxiang Wu, Shifeng Chen, and Junzhou Huang. Collaborative channel pruning for deep networks. In *International Conference on Machine Learning*, 2019.

[162] Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through l0 regularization. *ArXiv*, abs/1712.01312, 2017.

[163] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127, 2002.

[164] Matthew J. Hausknecht, Joel Lehman, Risto Miikkulainen, and Peter Stone. A neuroevolution approach to general atari game playing. *IEEE Transactions on Computational Intelligence and AI in Games*, 6:355–366, 2014.

[165] Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H. Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of arti-

ficial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 9, 2017.

[166] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv: Learning*, 2018.

[167] Ting Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *IJCAI*, 2018.

[168] Weiqiu Chen, Ling Chen, Yu Xie, Wei Cao, Yusong Gao, and Xiaojie Feng. Multi-range attentive bicomponent graph convolutional network for traffic forecasting. *ArXiv*, abs/1911.12093, 2020.

[169] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *IJCAI*, 2019.

[170] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12018–12027, 2019.

[171] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *ArXiv*, abs/1801.07455, 2018.

[172] Valerio La Gatta, Vincenzo Moscato, Marco Postiglione, and Giancarlo Sperlí. An epidemiological neural network exploiting dynamic graph structured data applied to the covid-19 outbreak. *IEEE Transactions on Big Data*, 7:45–55, 2021.

[173] Cornelius Fritz, Emilio Dorigatti, and D. Rügamer. Combining graph neural networks and spatio-temporal disease models to predict covid-19 cases in germany. *ArXiv*, abs/2101.00661, 2021.

[174] Amol Kapoor, Xue Ben, Luyang Liu, Bryan Perozzi, Matt Barnes, Martin J. Blais, and Shawn O'Banion. Examining covid-19 forecasting using spatio-temporal graph neural networks. *ArXiv*, abs/2007.03113, 2020.

[175] Daiki Matsunaga, Toyotaro Suzumura, and Toshihiro Takahashi. Exploring graph neural networks for stock market predictions with rolling window analysis. *ArXiv*, abs/1909.10660, 2019.

[176] Xiurui Hou, Kai Wang, Cheng Zhong, and Zhi Wei. St-trader: A spatial-temporal deep neural network for modeling stock market movement. *IEEE/-CAA Journal of Automatica Sinica*, 8:1015–1024, 2021.

[177] Kialan Pillay and Deshendran Moodley. Exploring graph neural networks for stock market prediction on the jse. *Artificial Intelligence Research*, 2022.

[178] Grégoire Pacreau, Edmond Lezmi, and Jiali Xu. Graph neural networks for asset management. *SSRN*, 2021.

[179] Danial Saef, Yuanrong Wang, and Tomaso Aste. Regime-based implied stochastic volatility model for crypto option pricing. *ArXiv*, abs/2208.12614, 2022.

[180] Henrik Hult, Filip Lindskog, Ola Hammarlid, and Carl Johan Rehn. Risk and portfolio analysis. 2012.

[181] Yves-Laurent Kom Samo and Alexander Vervuurt. Stochastic portfolio theory: A machine learning perspective. *Advanced Risk & Portfolio Management® Research Paper Series*, 2016.

[182] Gah-Yi Ban, Noureddine El Karoui, and Andrew E. B. Lim. Machine learning and portfolio optimization. *Manag. Sci.*, 64:1136–1154, 2018.

[183] Felipe D. Paiva, Rodrigo T. N. Cardoso, Gustavo P. Hanaoka, and Wendel Moreira Duarte. Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Syst. Appl.*, 115:635–655, 2019.

[184] Pier Francesco Procacci and Tomaso Aste. Portfolio optimization with sparse multivariate modelling. *arXiv*, 2103.15232, 2021.

[185] D. Kraft. A software package for sequential quadratic programming. *Tech. Rep. DFVLR-FB*, 88(28), 1988.

[186] P.T. Boggs and J.W. Tolle. Sequential quadratic programming. *Acta Numerica*, 4(1), 1996.

[187] J. Nocedal and S.J. Wright. Numerical optimization. *Springer-Verlag*, 2006.

[188] M. Jackson and M.D. Staunton. Quadratic programming applications in finance using excel. *The Journal of the Operational Research Society*, 50(12), 1999.

[189] F. Cesarone, A. Scozzari, and F. Tardella. Portfolio selection problems in practice: a comparison between linear and quadratic optimization models. *Computational Management Science*, 12(3), 2015.

[190] R.H. Singh, L. Barford, and F.C. Harris. Accelerating the critical line algorithm for portfolio optimization using gpus. *Advances in Intelligent Systems*, 448, 2016.

[191] Harry M. Markowitz, David Starer, Harvey Fram, and Sander Gerber. Avoiding the Downside: A Practical Review of the Critical Line Algorithm for Mean–Semivariance Portfolio Optimization. In John B Guerard and William T Ziemba, editors, *HANDBOOK OF APPLIED INVESTMENT RESEARCH*, World Scientific Book Chapters, chapter 17, pages 369–415. World Scientific Publishing Co. Pte. Ltd., 2020.

[192] D.H. Bailey and M.L. de Prado. An open-source implementation of the critical-line algorithm for portfolio optimization. *Algorithms*, 6(1), 2013.

[193] J. Narsoo. Performance analysis of portfolio optimisation strategies: Evidence from the exchange market. *International journal of economics and finance*, 9:124–132, 2017.

[194] William F. Sharpe. The sharpe ratio. *The Journal of Portfolio Management*, 21(1):49–58, 1994.

[195] Andrew W. Lo. The statistics of sharpe ratios. *Financial Analysts Journal*, 58(4):36–52, 2002.

[196] Yahoo Fiance. Shelton capital management nasdaq-100 index fund direct shares.

[197] Yahoo Fiance. ishares core ftse 100 ucits etf gbp (dist) (isf.l).

[198] Tong Zhang. Stock picking strategy based on exploration of chip distribution indicators. In *2020 International Conference on Computing and Data Science (CDS)*, pages 276–282, 2020.

[199] Darrell Duffie and Jun Pan. An overview of value at risk. *The Journal of Derivatives*, 4(3):7–49, 1997.

[200] A. Ford Ramsey and Barry K. Goodwin. Value-at-risk and models of dependence in the u.s. federal crop insurance program. *Journal of Risk and Financial Management*, 12(2), 2019.

[201] Alan L. Stuart and Harry M. Markowitz. Portfolio selection: Efficient diversification of investments. *A Quarterly Journal of Operations Research*, 10:253, 1959.

[202] Rosario N. Mantegna. Hierarchical structure in nancial markets. 1999.

[203] Jukka-Pekka Onnela, Anirban Chakraborti, Kimmo K. Kaski, János Kertész, and Antti J. Kanto. Dynamics of market correlations: taxonomy and portfolio analysis. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 68 5 Pt 2:056110, 2003.

[204] Yuanrong Wang and Tomaso Aste. Dynamic portfolio optimization with inverse covariance clustering. 2021.

[205] Pier Francesco Procacci and Tomaso Aste. Portfolio optimization with sparse multivariate modeling. *Journal of Asset Management*, 23:445 – 465, 2021.

[206] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. 1986.

[207] Hasim Sak, Andrew W. Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH*, 2014.

[208] Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv*, abs/1412.3555, 2014.

[209] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017.

[210] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *ArXiv*, abs/1506.02025, 2015.

[211] Ailing Zeng, Mu-Hwa Chen, L. Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *AAAI Conference on Artificial Intelligence*, 2022.

[212] Junbin Gao, Yi Guo, and Zhiyong Wang. Matrix neural networks. *ArXiv*, abs/1601.03805, 2017.

[213] Deng Cai, Xiaofei He, and Jiawei Han. Learning with tensor representation. 2006.

[214] Shuo Chen, Jian Kang, Yishi Xing, Yunpeng Zhao, and Don Milton. Estimating large covariance matrix with network topology for high-dimensional biomedical data. *Comput. Stat. Data Anal.*, 127:82–95, 2018.

[215] Jeremy D. Turiel, Paolo Barucca, and Tomaso Aste. Simplicial persistence of financial markets: filtering, generative processes and portfolio risk. *arXiv: Statistical Finance*, 2020.

[216] Pier Francesco Procacci and Tomaso Aste. Forecasting market states. *Machine Learning and AI in Finance*, 2021.

[217] Tristan Millington and Mahesan Niranjan. Robust portfolio risk minimization using the Graphical Lasso. In *ICONIP*, 2017.

[218] Xin Yuan, Weiqin Yu, Zhixian Yin, and Guoqiang Wang. Improved large dynamic covariance matrix estimation with graphical Lasso and its application in portfolio selection. *IEEE Access*, 8:189179–189188, 2020.

[219] Tae-Hwy Lee and Ekaterina Seregina. Optimal portfolio using factor graphical Lasso. *arXiv: Econometrics*, 2020.

[220] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21:3848–3858, 2020.

[221] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *ArXiv*, abs/1806.08804, 2018.

[222] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[223] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907, 2017.

[224] Kaggle. Kaggle: Store item demand forecasting challenge. `https://www.kaggle.com/c/demand-forecasting-kernels-only/data`. Accessed: 2022-02-06.

[225] Marco Grassia and Giuseppe Mangioni. wsgat: Weighted and signed graph attention networks for link prediction. *ArXiv*, abs/2109.11519, 2022.

[226] Joaquín J. Torres and Ginestra Bianconi. Simplicial complexes: higher-order spectral dimension and dynamics. *Journal of Physics: Complexity*, 1, 2020.

[227] Cristina Costa-Santos, João Bernardes, Luis Filipe Coelho Antunes, and Diogo Ayres de Campos. Complexity and categorical analysis may improve the interpretation of agreement studies using continuous variables. *Journal of evaluation in clinical practice*, 17 3:511–4, 2011.

[228] Luis Gregorio Moyano. Learning network representations. *The European Physical Journal Special Topics*, 226:499–518, 2017.

[229] M. Tumminello, T. Aste, T. Di Matteo, and R. N. Mantegna. A tool for filtering information in complex systems. *Proceedings of the National Academy of Sciences*, 102(30):10421–10426, 2005.

[230] Antonio Briola and Tomaso Aste. Topological feature selection: A graph-based filter feature selection approach. *ArXiv*, abs/2302.09543, 2023.

[231] Guido Previde Massara, Tiziana di Matteo, and Tomaso Aste. Network filtering for big data: Triangulated maximally filtered graph. *J. Complex Networks*, 5:161–178, 2017.

[232] Guoqiang Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

[233] Antonio Briola, David Vidal-Tom'as, Yuanrong Wang, and Tomaso Aste. Anatomy of a stablecoin's failure: the terra-luna case. *ArXiv*, abs/2207.13914, 2022.

[234] Zihao Zhang, Stefan Zohren, and Stephen J. Roberts. Deep learning for portfolio optimization. In *The Journal of Financial Data Science*, 2020.

[235] R.B. Wilson. A Simplicical Algorithm for Concave Programming. In *PhD Dissertation*. Harvard University, 1963.

[236] S.P. Han. A globally convergent method for nonlinear programming. *J.Opt.Theory Applic.*, 22:248–256, 1997.

[237] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. Fletcher, Academic Press (London), url=https://api.semanticscholar.org/CorpusID:115696750, 1978.

[238] W. Hock and K. Schittkowski. Test examples for nonlinear programming codes. *Journal of Optimization Theory and Applications*, 30:127–129, 1980.

[239] R.T. Rockafellar. The multiplier method of hestenes and powell applied to convex programming. *J.Opt.Theory Applic.*, 12:555–562, 1973.