Investigating the Application of Graph Neural Networks in an Intelligent System for Automated Portfolio Management

Literature Review

Kialan Pillay
pllkia010@myuct.ac.za
University of Cape Town
Cape Town, South Africa

ABSTRACT

Portfolio management is a complex decision-making process to achieve investment objectives. The initial share evaluation phase is ordinarily a manual task requiring expert knowledge. The emergence of Graph neural networks (GNNs) allows for modelling dynamic systems with non-temporal dependencies. This paper reviews the literature to examine deep learning solutions to portfolio management and identify GNN architectures to assess their applicability. Our review identifies two significant weaknesses in the literature. The first is the deficit of deep learning frameworks for share evaluation and unified portfolio management. The second is the deficit of GNN applications in the financial domain. We conclude that additional research is required to investigate the applicability of GNN models to the presented problem.

CCS CONCEPTS

• **Computing methodologies → Neural networks**.

KEYWORDS

share evaluation, portfolio management, deep learning, graph neural networks

1 INTRODUCTION

Over the last ten years, deep learning models have superseded classical machine learning techniques as a powerful tool for recognition, detection and prediction in a range of domains. Deep learning involves multi-layer computational processing models that effectively capture the latent features of the data at multiple abstraction levels [41]. The performance of these models has been aided by improvements in parallel computing technologies and big data collection techniques [59]. These advances allow supervised learning models to be effectively and efficiently trained on large datasets. Graph neural networks (GNN) are a subclass of deep learning models that are applied to data that is naturally represented by graph structures [60], or data that has both temporal and spatial dependencies [79].

Portfolio management is a complex phased decision-making process that aims to achieve a set of investment objectives. The initial phase involves share evaluation: identifying securities for inclusion in a portfolio. Share evaluation requires expert domain-specific knowledge [15] and is not widely automated. Graph neural networks are an emergent model class that could be applied to automate portfolio management sub-problems.

This literature review aims to investigate the applicability of GNNs for price prediction and share evaluation. By reviewing the literature in this field, we seek to gain an understanding of the current state-of-the-art techniques, how these have applied to the financial market time series in the context of portfolio management, and the gaps existing in the current research, both application and architectural. This review will also assist in understanding how GNN models can be incorporated into intelligent systems to support enhanced decision-making.

This literature review will explore five key areas which are of particular importance to our research objectives. First, we include a discussion on equity markets and share evaluation. This section will be followed by a review of time series prediction tasks and deep learning-based techniques. The third section investigates deep learning approaches to portfolio management sub-problems. We then present a detailed treatment of GNNs, in which we will compare state-of-the-art architectures and discuss their application to portfolio management. The final section surveys metrics and procedures for model validation. In closing, we discuss our findings and their related importance and conclude with key points extracted from the reviewed literature.

2 EQUITY MARKETS

An equity market is a financial market that facilitates the exchange of ownership shares of listed financial securities [15]. Each share grants the holder fractional ownership of the company and is a representation of equity interest. Market participants to purchase shares in a company through exchanges and trade these amongst other participants. Supply and demand are the key determinants of share prices, and an investor aims to earn a profit through the acquisition of and successive sale of shares. This is termed as the return on investment. Excess returns, or alpha, is a key performance measure [27] defined as the ratio of portfolio return to a benchmark return.

2.1 Portfolio Theory

In this subsection, we present the Efficient Market Hypothesis (EMH) and the implications of counterarguments for investing and portfolio management. The EMH is an investment theory which states that the financial market is efficient in reflecting current information and security prices are immediately and unbiasedly reflective of available information [18]. The implication is that neither analysis of financial information nor historical prices would provide greater returns for an investor than a randomly diversified portfolio of stocks [50].

However, financial markets are not perfectly efficient as asserted by Fama [18]. Subsequent research has proved the existence of persistent market inefficiencies through both theoretical and empirical means [14] [58] [2] [51]. Barriers in a financial market are limiting factors to the previously hypothesised market efficiency. Jensen defined market efficiency with respect to information as the inability to earn an economic profit based on trading on that information [32]. The converse implication is that information and the corresponding market inefficiency present opportunities for investors to earn a profit.

2.2 Share Evaluation

In this subsection, we introduce portfolio management, discuss share evaluation and current investor strategies for evaluation.

2.2.1 Overview. Portfolio management is a decision-making process that aims to achieve a set of investment objectives. These objectives are selected by the decision-maker or stakeholders and principally revolve around risk minimisation [85] and value maximisation [67]. Share evaluation is the initial phase of this process. It involves identifying shares that have suitable risk-return characteristics [15] for inclusion in a portfolio of financial instruments. This identification process is achieved through detailed research and analysis. The decision-maker has a reasonable expectation that the selected shares will generate excess returns and profitability for the portfolio manager. The shares identified during this phase are input into the second phase of the portfolio management process: portfolio construction.

Portfolio construction (commonly termed selection) refers to determining the optimal allocation and diversification of a set of shares to maximise returns [21]. A solution to this portfolio selection problem was pioneered by Markowitz, who introduced the Mean-Variance model in his seminal paper [52]. This model is a mathematical framework for constructing a portfolio such that the expected return is maximised for a certain level of variance or market risk. Once this second decision-making phase is complete, the total available capital is divided and invested according to the identified percentages to construct the optimal portfolio.

2.2.2 Investing Strategies. There are two general investing approaches employed by market participants: technical analysis and fundamental analysis. Edwards et al. [16] define technical analysis of financial markets as a technique that provides asset price forecasts on the basis of visual inspection of historical price action. Brown and Jennings [7] present a similar definition, instead referring to future price as private information. Technical analysis is based on the fundamental assumption that historical price patterns reoccur in the future and that successive price changes are dependant [15]. Drake comments that technical approaches do not hold in the long run since these techniques assume that asset prices are based purely on speculation rather than economic fundamentals [15].

Fundamental analysis is the evaluation of financial statement data [1] and factors other than price to guide investment decisions and predict future price movements. The fundamental analysis strategy can be further subdivided into two distinct approaches: value and growth investing. The value investing strategy is based on the premise that undervalued stocks exist relative to economic fundamentals; the share price is below its intrinsic value [43]. These shares are expected to deliver an abnormal return as the price reverses towards its intrinsic value. Fama and French [19] pioneered a formalised methodology for an investment strategy using the value approach. The growth investing strategy is premised on investing in high-valued securities that are deemed to have significant future growth prospects. These shares are typically associated with companies with strong earnings growth and innovative technology [15].

A review study by Chan and Lakonishok concluded that the majority of empirical research indicated that the value investing approach earns higher returns than a growth investing strategy. However, Lev and Srivastava [43] argue that over the last 10-12 years, value investing is failing to deliver excess returns. Despite inconsistent performance, both strategies are exceedingly popular and often employed in tandem by portfolio managers [15].

3 TIME SERIES PREDICTION

In this section of the review, we begin with a brief introduction to time series, categorise prediction tasks and conclude with a review of deep learning forecasting techniques. The universal procedure for time series prediction consists of the following six steps: (1) data partitioning, (2) parameter selection or estimation, (3) model building, (4) current value prediction, (5) performance evaluation and (6) future value prediction [56].

Prediction techniques primarily differ in the second and third stages of this process. The class of parametric methods, including linear models, require *a priori* information about the distribution that governs the data. On the other hand, non-parametric techniques, the focus of this review, initialise and update the model parameters in an iterative process to minimise the predictive error.

3.1 Background

A univariate time series is a set of random variables $\{X_t, t \in T\}$, where T is a discrete or continuous index set. Informally, a time series is a sequence of observations ordered in time. If T is discrete, then $T = \{1, 2, ..., N\}$, else the index set is a finite interval $T = \{t \mid 0 \leq t \leq N\}$. Each random variable has a probability distribution F_t, and this collection of random variables is a stochastic process. For each $t \in T$, X_t is a single realization from this distribution, and thus a time series is a observation of a stochastic process [57]. A multivariate time series is defined as a set of M univariate series.

3.2 Prediction Tasks

This subsection presents a brief overview of two prediction task classes: point prediction and trend prediction. Trend prediction can be further subdivided into two general sub-problems: multi-step-ahead prediction and sequence movement prediction.

3.2.1 Point Prediction. Point or one-step-ahead prediction is concerned with predicting the value of a single future observation conditional on the historical temporal observations. The majority of conventional linear models [4] and machine learning (ML) techniques [3] apply to this prediction task. Point prediction can be formulated as a regression problem; a model is fit to the time series data and outputs a continuous value.

3.2.2 Movement Prediction. Sezer et. al [61] formulate movement prediction in terms of a classification problem, in which the objective of the model is to determine the direction of value change. Their review found that studies formulate this problem either as a dual-class or tri-class classification problem; the latter considers neutral future trends in designing the output layer of the model. The fitted model outputs a class label corresponding to the predicted movement of the global trend.

3.2.3 Multi-Step-Ahead Prediction. Multi-step-ahead prediction is the task of predicting a sequence of values in a time series instead of the value of a single future observation. Multi-step-ahead prediction is applicable in multiple domains but involves additional complexity due to error accumulation. Error accumulation is the propagation of past error into future predictions. [12] which decreases the model's predictive accuracy.

3.2.4 Trend Prediction. The trend of a time series is defined as $T = \{< l_1, s_1 >, ..., < l_n, s_n >\}$, computed through piece-wise linear approximation [37] of the original time series X_t. The task is formulated as the prediction of the trend component $< l_{n+1}, s_{n+1} >$. We describe this as a quasi-combination of both multi-step-ahead and movement prediction.

3.3 Deep Neural Network Techniques

Sezer et al. [61] note that over the last decade, deep neural network (DNN) architectures have surpassed the performance of conventional ML models. DNNs are also non-parametric data-driven models [54] that extract information without *a priori* knowledge of the underlying probability distribution. This subsection presents an overview of relevant DNN model classes and then discusses DNNs for trend prediction. Point prediction is discussed in a successive section in the context of financial time series.

3.3.1 Overview. A DNN is a composition of L parametric functions called layers. Each layer l_i consists of one or more neurons, which are computational units that transform the input into the layer's output. Each of these transformations is a non-linear parametric function. The parameter vector Θ (weights) link a previous layer's input to the current layer's output. The final output is calculated through the composition of these non-linear computations. Deep Multi-layer Perceptrons (DMLP) are a sub-class of DNNs that are feed-forward only; the architecture contains no loops.

A Convolutional Neural Network (CNN) consists of convolutional layers that transform the input using convolutions. Fawaz et al. [20] define a convolution as applying a non-linear transformation ω (termed a filter) to an input. CNNs are state-of-the-art techniques for image recognition tasks [40] [68]. However, subsequent research has shown that convolutions can be adapted for time series prediction by reducing the dimensionality of the filters from two to one [20].

A Recurrent Neural Network (RNN) is a DNN that contains connections between non-successive layers, forming a cycle. These residual connections enable to network to store internal state and learn temporal dependencies. These networks can also be trained using the backpropagation algorithm [26]. LTSM models are a subclass of RNN models that can recall values over variable ranges [20]. LSTM differs by including input and output gates in each computational unit that prevent error from propagating through the rest of the network [26]. This improves predictive accuracy and increases efficacy on long-range sequences.

3.3.2 Deep Neural Networks for Trend Prediction. Lin et. al [46] propose a hybrid deep neural network model (TreNet) that combines a CNN and LSTM model to predict the future trend. The CNN component consumes raw time series data whilst the LSTM model consumes segmented historical trends. These models extract local and global features to produce an output trend [39]. Lin et. al demonstrate that their model outperforms existing models. However, Kouassi and Moodley [39] critique the robustness of the validation procedures employed by the authors.

Kouassi and Moodley build on the previous work by evaluating an MLP, CNN and LSTM against the TreNet architecture. Instead of standard cross-validation with a single test set, Kouassi and Moodley utilise a walk-forward validation procedure that preserves the temporal nature of the input data by partitioning the data into overlapping subsets [38]. Kouassi and Moodley find that TreNet only infrequently outperforms the general DNN architectures and by no significant margin. Kouassi and Moodley also [39] explore the combined algorithm selection and parameter optimisation (CASH) problem [22] for deep learning models for trend prediction. Their research output is an experimental platform for AutoML based on the Bayesian optimisation and Hyperband framework [17]. The authors demonstrate that AutoML can effectively automate the search for an optimal model-parameter configuration.

Fawaz et al. [20] published an exhaustive review of deep learning techniques for time series classification. The authors trained nine DNN models on 97 datasets consisting of both uni- and multivariate series. The results indicate that the Residual Network (ResNet) [77] architecture outperforms other evaluated models for both univariate and multivariate forecasting. ResNet is an 11-layer architecture comprised primarily of residually-connected convolutional layers. This reduces the effect of the vanishing gradient problem and increases the efficiency of model training [28]. The vanishing gradient problem refers to partial derivatives computed by the backpropagation algorithm tending to zero. This effectively halts the learning procedure as the weights are no longer iteratively updated. The network depth allows the model to generalise effectively on unseen data [20], although He et al. [28] note that a significant amount of training data is usually required to achieve suitable accuracy.

Long, Lu and Cui [47] focus specifically on movement prediction using multivariate time series data market. The authors propose Multi-Filter Neural Network (MFNN), a hybrid DNN architecture similar to ResNet [77] that consists of both convolution and recurrent computational units. Long et al. evaluated model performance against baseline RNN, LSTM and CNN models and conventional ML techniques. The results demonstrate that MFNN outperforms all baseline models, measured using both error and profitability metrics. In addition to conventional validation, Long et al. perform a market simulation to evaluate model performance. We highlight the robustness of this evaluation methodology. The inclusion of profit metric-based evaluation increases the strength of the authors' claims and dually illustrates its importance for comprehensively measuring the performance and stability of model applications in finance. This is the topic of the next section of our review.

4 DEEP LEARNING FOR PORTFOLIO MANAGEMENT

To understand the current techniques, we review the literature on price prediction, portfolio selection and share evaluation using deep learning approaches. A copious review of 124 papers by Jiang [33] illustrates the significant body of published literature in this field, although the majority is focused on forecasting rather than portfolio management sub-tasks. Also, we comment on the absence of unified nomenclature. Share evaluation is commonly referred to as asset or stock (pre)selection, and portfolio selection is termed optimisation in several papers.

4.1 Price Prediction

Whilst forecasting the future price of a security is not a conventional component of the portfolio management process, it nonetheless provides basic decision-support for share evaluation. In their review, Sezer et al. [61] analyse deep learning model classes for financial time series forecasting. In contrast to Fawaz et al. work, Sezer et al. only identify frequently applied model classes without evaluating specific architectures. The absence of experimental results is a limitation of the study, although it does not detract from the comprehensive survey of existing deep learning research in the financial domain.

Sezer et al. found that the majority of reviewed literature preferred RNN, LSTM, and Hybrid models. The prevalence of the application of these models is consistent across different financial markets, including equity, indices, bond and cryptocurrency. Sezer et al. conclude that approximately half of the reviewed papers evaluate the RNN and LSTM models. We note that hybrid models that combine the RNNs with an LSTM or CNN or RNNs combined with CNNs have also increased in popularity. Tealab [71] found that hybrid models produced more accurate predictions compared to standalone models, although we note his contribution is not a comprehensive review of the literature. An important aspect of prediction is the length of the forecasting horizon [3]. Our review of the literature illustrates the proposed models are successively applied to both one-step- [10] and multi-step-ahead [29] [9] prediction tasks.

4.2 Portfolio Selection

In this subsection, we discuss deep learning techniques for portfolio selection. Recall that portfolio management is defined as the decision-making process of proportional allocation and redistribution [35] of investable capital into a set of securities to maximise return. We can thus formulate portfolio selection as an optimisation problem. Although a recent study by Tsang et al. [72] proposes a DNN-based sub-network architecture, we find that current techniques presented in the recent literature are predominately based on deep reinforcement learning (DRL) frameworks [35] [31] [75].

RL is a multi-stage decision-making process in which an RL agent learns to map states to action to maximise a reward. This finding is unsurprising; reinforcement learning can be defined as an optimisation technique that seeks to maximise a numerical performance [69] measure to find an optimal solution. This is consistent with the objectives of the portfolio management process.

In their two recent contributions [34] [35], Jiang and Liang propose a deep reinforcement learning framework for cryptocurrency asset portfolio management. The authors assume a liquid market and negligible investment market influence. A CNN model computes a portfolio weight vector is input into the objective function that is iteratively maximised to compute the optimal policy. This diverges from previously discussed CNN architectures that perform point or trend prediction. The experimental results indicate that the CNN produced the maximal portfolio return, although a DNN exhibited more stable performance over eight experimental runs than the CNN. Jiang et al. [35] build on their previous work by constructing the policy functions (formulated previously directly as a weight vector) using CNN, RNN and LSTM models. The study results demonstrate that the proposed framework outperformed all surveyed portfolio selection methods, including their previous DRL framework. We note that the scope of the studies is constrained to the cryptocurrency market, which may exhibit different dynamics to other financial markets.

Huang et al. [31] propose a similar DRL framework although their study differs in using an algorithm that consists of dual DNNs to learn the optimal reward function. The authors compare four DRL-selected portfolios with a Chinese stock market benchmark. The results demonstrate that the agent-selected portfolios generate excess returns. The proposed architecture also produced greater returns than classical equal-weight and mean-variance optimisation [52] strategies. The authors were able to sufficiently demonstrate the robustness of their DRL framework, but not generalisability; the experiments were only conducted using Chinese stock market securities. An important limitation to note of the reviewed publications is manual asset pre-selection, effectively bypassing the share evaluation phase of the portfolio management process. Additionally, the conditions of market liquidity and negligible capital impact may not consistently hold.

4.3 Share Evaluation

In this subsection, we discuss techniques for identifying shares for portfolio inclusion. This process is also defined as preselection in the literature.

Fu et al. [25] conducted a study of both ML and DNN models. To label the candidate securities, the study draws inspiration from the Sharpe Ratio, a measure of a company's profit to risk [62]. The authors constructed features representing both technical and fundamental signals and apply a Genetic Algorithm (GA) to conduct a feature-space optimal subset search, a method first proposed by Yang and Honavar [83]. GAs are a family of meta-heuristic ML models inspired by evolution [53] that iteratively optimise a set of candidate solutions. Yang et al. [82] devise a similarly inspired framework for share evaluation, but combine both fundamental factors and predicted returns of the candidate securities. However, their research is less relevant as only optimisation techniques [13], and not DNNs, are applied to price prediction and share evaluation. Fu et al. found that a stacked DNN-based model predicts the maximally returning portfolio, measured using excess return. However, the study is limited by only evaluating the models on Chinese stock market data. Furthermore, the research fails to evaluate other deep learning architectures.

4.4 Unified Portfolio Management

Our search has illustrated that recent contributions have begun to focus on end-to-end portfolio management. Leon et al. [42] conduct both share evaluation and portfolio selection, although their study evaluates only unsupervised learning techniques. Leon et al. [42] use clustering algorithms to identify an optimal set of shares and apply optimisation techniques to select a portfolio with maximised returns.

Both Ta et al. [70] and Wang et al. [76] use an LSTM model to perform share evaluation based on the predicted price. This prediction is performed using identified technical features. The results of both studies demonstrated that the LSTM model for share evaluated combined with Mean-Variance Optimization (MVO) for the portfolio selection phase produced the highest returns. Both sets of authors use square- and absolute-error metrics to evaluate the performance of the LSTM model for price prediction. Return metrics are used to evaluate the performance of the predicted portfolios.

In our search of the literature, we state, to the best of our knowledge, that Ma et al. [49] have produced the most comprehensive study of DNN models for both share evaluation and portfolio selection. Building on the methodology of the aforementioned papers, Ma et al. compared a DMLP, LSTM and CNN to baseline ML and ARIMA models. Diverging from the previous work, Ma et al. use only historical price data to predict future return. Whilst the results indicated that LSTM outperforms the other DNN-based models, a Random Forest [5] ML model produced the overall lowest predictive error. The authors evaluate both MVO and the Omega ratio [36] for optimisation of the portfolio weights, similarly finding that MVO selects the optimal portfolio.

We comment on several observations about the literature discussed. Firstly, a deep learning approach is applied only to the share evaluation component, using a price forecasting strategy. The experimental results provide evidence contrary to the hypothesis of DNN outperformance. Secondly, each study is limited by its model evaluation; whilst it is robust, data from a single financial market fails to demonstrate the generality of the models. Finally, we note that the current literature provides an empirical foundation for further investigation of price prediction as a proxy for share evaluation.

5 GRAPH NEURAL NETWORKS

Graph Neural Networks (GNN) are a subclass of deep learning models first proposed by Scarselli et al. [60]. We review the literature to understand state-of-the-art architectures and GNN applications to portfolio management.

5.1 Background

Data and the relationships between objects can be modelled graphically in almost every domain. This observation of naturally occurring graph structures in systems motivated Scarselli et al. [60] to introduce a novel neural network architecture that extended existing supervised neural network methods to process graphically-encoded non-Euclidean data. Historically, RNNs [24] [66] and Markov chains [73] applied to graph-based problems, notably exploited by Brin and Page [6] for search engine creation. In their recent review on GNN methods and applications, Zhou et al. [88] comment that the aforementioned techniques, although successful, relied on converging to a solution, which constrains their generalisability to other systems in which convergence is impossible. Moreover, Zhou et al. note that recent advances in the field of deep learning have supported the development of GNN-based architectures. Wu et al. attribute the success of GNN models to its properties of permutation-invariance, local connectivity and compositionality [78].

Mathematically, a graph is a pair $G = (V, E)$, where V the set of nodes (data objects) and E the set of edges (relationships between data). A feature vector X_v is associated with each node $v \in V$. The labels of a node v, edge $e = (v, u)$ or graph G are real-valued vectors. The adjacency matrix $A \in \mathbb{R}^{N \times N}$ is a mathematical representation of a graph G, with $A_{ij} > 0$ for $(v_i, v_j) \in E$ and $A_{ij} = 0$ for $(v_i, v_j) \notin E$.

GNN models are designed to perform two distinct predictive tasks: node classification and graph classification. Node classification refers to the model objective of learning a representation vector h_v for $v \in V$ and a function f, such that the predicted node label is $y_v = f(h_v)$ [81]. In node classification applications, f is conditional on v, and hence the classification or regression is dependant on the node properties [60]. Graph classification tasks instead learn a representation vector h_G and a function g to predict the graph label y_G [81]. The function g is independent of any node $v \in V$ and thus unconditional on the node-level properties of the graph.

The representation vector h_G can be described as the global features of the graph, and similarly, h_v are the node-level local features. The learning algorithm for GNNs implements a neighbourhood aggregation strategy [81]. The representation vector h_v is initialised to the feature vector X_v and iteratively by aggregating the representation vectors of neighbouring nodes $N(v) = \{v \in V \mid (v, u) \in E\}$. This recursive update procedure of each vector h_v for $v \in V$ captures the network structure and neighbourhood-level local features.

5.2 Architectures

RNN-based approaches [45] are vulnerable to the vanishing gradient problem [28] and ineffective on long-range temporal sequences. Diffusion Convolutional Recurrent Neural Network (DCRNN) [45] is one such example of a hybrid architecture, extracting spatial information and passing the output into an RNN to learn temporal dependencies [88]. CNN models can be adapted to learn temporal dependencies using one-dimensional convolutions but require an exceptionally deep network for learning to be effective [79]. Spatial-temporal graph convolutional networks (STGCN) [86] [65] models are CNN-based approaches frequently applied to spatial-temporal graph modelling tasks.

Existing literature incorrectly assumes that the graph reflects real dependencies [79]. In practical systems, a dependency can exist if an edge is absent from the graph, nor does an edge always imply a relation between nodes. Additionally, prior knowledge of this graph structure is assumed and the models rely on a fixed adjacency matrix. This approach is unsuitable for systems with hidden or implicit spatial dependencies, like financial markets. In this subsection, we present a detailed comparison of three graph neural network architectures that overcome these limitations: Graph WaveNet [79], MTGNN [78] and StemGNN [8].

5.2.1 Graph WaveNet. Graph WaveNet (GWN) is a hybrid deep learning architecture for spatial-temporal graph modelling. Wu et al. define a spatial-temporal graph G as consisting of a set of nodes V, where each node v has dynamic input features [79]. Zhou et al. define graphs with time-varying input features as dynamic graphs [88].

The GWN architecture consists of temporal convolution (Gated TCN) and graph convolution (GCN) modules. The GCN module consists of a self-adaptive adjacency matrix that requires no prior knowledge and is learnt using the stochastic gradient descent algorithm, although can be seeded with prior spatial knowledge (graph structure) if available. The TCN module consists of one-dimensional convolutions that capture long-range temporal sequences [55] whilst avoiding the vanishing gradient problem. The GCN module extracts node-level features using neighbourhood feature aggregation. The model outputs the predicted sequence over the entire prediction horizon.

Wu et al. evaluate the architecture on traffic network datasets [45]. The study results demonstrate that GWN outperformed both the temporal models and spatial-temporal models, although the latter by a reduced margin. The authors find that the model trained with the prior graph structure produces the lowest error. Further, the self-adaptive adjacency matrix without prior information produced reasonable performance. We note that results tentatively suggest that GWN suitable for application to problems without knowledge of the graphical structure. Schleifer, McCreery and Chitters [63] propose several modifications to the GWN architecture (subsequently termed GWN++) and model hyperparameters to improve its performance. Their results demonstrate that minor adjustments positively impact performance with only nominal degradation of computational efficiency. We remark that the evaluation methodology is a significant limitation of both studies. GWN and GWN++ are compared against a minimal set of spatial-temporal models and on relatively small-scale datasets. The authors fail to demonstrate the applicability of the GWN architecture in multiple domains.

5.2.2 MTGNN. MTGNN is a generic framework proposed by Wu et al. [78] with an explicit focus on multivariate time series forecasting. Wu et al. posit that existing spatial-temporal graph neural networks are not suitable for modelling multivariate time series due to two distinct factors: unavailability of prior structural information and the sub-optimality of the graph structure [78]. GWN can accommodate unknown graph structures, although the architecture does not update the graph structure during model training. MTGNN consists of a distinct graph learning layer that extracts the adaptive graph adjacency matrix. The MTGNN architecture exhibits several similarities to GWN. A neighbourhood aggregation strategy learns spatial dependencies and the TCN module extracts temporal dependencies. MTGNN avoids the vanishing gradient problem by including residual and skip intra- and inter-layer connections. However, model learning uses a strategy that splits the input into subgroups and is thus more computationally efficient than GWN. We note that this becomes more apparent on larger datasets, unlike those used by Wu et al. in model evaluation. Wu et al. [78] conduct a robust evaluation of MTGNN against a selection of non-GNN models for one-step-ahead forecasting and both RNN-based and STGNN-based models for multi-step-ahead forecasting, including DCRNN [45], STGCN [86] and GWN [79]. MTGNN outperforms the evaluated models for one-step-ahead forecasting tasks. We note that the performance of MTGNN is consistent with the explicit spatial dependencies exhibited by the datasets. Further, the results demonstrate the model's generalisability. The results indicate that whilst MTGNN is performant, it is unable to achieve state-of-the-art [11] [87] performance for the multistep-ahead predictive task. However, MTGNN achieves parity with the performance of DCRNN and STGCN without the prior availability of an adjacency matrix. This is a significant finding given that DCRNN [45] and STGCN [86] rely on a fixed adjacency matrix. We note that MTGNN and GWN can perform sufficiently without prior knowledge of the graph structure.

5.2.3 StemGNN. Spectral Temporal Graph Neural Network [8] is a hybrid architecture for multivariate time series forecasting that captures inter-series correlations and temporal dependencies in the spectral domain. The spectral component of the model diverges from previously discussed architectures which extract spatial dependencies. Spectral-Temporal GNN models are trained on the spectral representation of the graph [88] using graph signal processing techniques first introduced by Shuman et al. [64]. A graph signal x is transformed into the spectral domain by a Fourier transform $\mathcal{F}$, a convolution operator is applied to the spectral signal, and the inverse Fourier transform $\mathcal{F}^{-1}$ is applied to transform the signal into its original representation [88]. StemGNN consists of a Latent Correlation Layer to automatically learn correlations between time series without a pre-defined graph structure. However, StemGNN improves on previous architectures in the literature that ignore inter-series correlations and thus do not jointly capture temporal and multivariate dependencies.

Cao et al. comprehensively evaluate StemGNN against baseline and state-of-the architectures, including GWN, on nine separate datasets. The study results demonstrate that StemGNN consistently outperforms the evaluated models on all datasets without prior knowledge of the graph. In the included analysis on traffic forecasting applications, Cao et al. also note that StemGNN produces excellent predictive performance without reducing interpretability [8]. We note that the experimental results illustrate the generalisability of the StemGNN architecture. Additionally, compared to prior graph knowledge, the correlation layer captures hidden structural dependencies more effectively, evidenced by the superior performance of StemGNN against models such as DCRNN [45] and STGCN [86] that require a fixed adjacency matrix. The authors conclude that StemGNN produces state-of-the-art performance on these datasets. However, we note that the study is limited by excluding MTGNN [78] from the model evaluation; the authors' claim is less justified. We remark that evaluating MTGNN presents a more representative comparison given its specific focus on multivariate time series.

5.3 Graph Neural Networks for Portfolio Management

5.3.1 Existing Applications. In their discussion of GNN applications, Zhou et al. [88] subdivide applications into two categories: structural and non-structural. Structural applications involve the application of GNNs to data that exhibits an explicit relational structure. On the other hand, non-structural applications refer to GNNs that applied to data that has implicit or hidden relations [88]. We note that multivariate stock market data is a class member of non-structured data; system modelling does not expose the relational schema, nor can the data be easily encoded into a graph structure. After a thorough survey, we state to the best of our knowledge that research is yet to be published on the application of GNNs to portfolio management. Zhou et al. [88] review also fails to identify any publications that investigate this research question. Instead, they find that the primary non-structural applications are focused on image processing [30] [80] and natural language processing [84], for which there exists a significant body of recent literature. Our review did identify a GNN-based architecture for dual-class movement prediction in the stock market by Li et al. [44]. Their research is notable for their encoding of the non-structural data. This graphical structure is the topic of the subsequent section.

5.3.2 Graph Structure Formulation. LSTM Relational Graph Convolutional Network (LSTM-GCN) [44] uses the correlation between shares to capture non-temporal dependencies and encode them in a graphical structure. Li et al. assume that shares are dependant and thus the correlation is representative of the latent non-temporal structure. The correlation matrix is calculated using historical price data. Li et al. motivate this choice by noting that the historical data encompasses prior market movements [44]. The graph is combined with fundamental data and fed into the LSTM-GCN model to predict future trend movement. LSTM-GCN is evaluated against baseline ML and LSTM classification models. Although the study is limited by the scope of its evaluation, it demonstrates the potential for using a correlation matrix to represent prior relational knowledge for prediction tasks using GNN models.

5.3.3 Applicability. To assess their applicability to the finance domain, we discuss the strengths and weaknesses of the aforementioned GNN architectures. These properties are summarised in table 1. Principally, we note that none of the reviewed GNN architectures are evaluated on stock market data. MTGNN is the only architecture that is applied in the finance domain. However, the performance is evaluated on an exchange-rate series; the currency exchange market exhibits unique dynamics. As noted previously, whilst MTGNN and StemGNN are sufficiently general, they are untested on stock market time series.

Recall that in contrast to GWN and MTGNN, StemGNN extracts dependencies in the spectral domain instead of the spatial domain. However, based on the comprehensive experimental results presented for several distinct domains, we posit that this does not affect applicability. GWN, MTGNN and StemGNN are all able to automatically learn the graphical structure without prior knowledge. As discussed, this overcomes the limitations of alternative GNN architectures. We note that this adaptive capability is especially relevant for non-structural applications. A notable weakness of StemGNN is that in its explicit focus on general modelling cases, it fails to accommodate prior knowledge if available. GWN and MTGNN are both more flexible architectures and allow for the adjacency matrix to be initialised. We note that the models are uniformly evaluated using an identical set of evaluation metrics. This supports both the reproducibility of the presented results and a robust evaluation of model performance on other datasets. Without experimental results and the absence of relevant literature, we cannot infer their performance. However, based on the comprehensive review presented in the previous section, we posit that the identified architectures apply to stock market times series data. Furthermore, the studies demonstrate that GWN, MTGNN and StemGNN can perform both one-step-ahead and multi-step-ahead predictive tasks. This facility is consistent with the predictive tasks performed by DNN-based solutions for price forecasting.

6 MODEL VALIDATION

An important component of any research is the evaluation of the performance and robustness of a system or model. Suitable evaluation metrics and validation procedures are required to validate the out-of-sample performance and accurately compare the model against alternative solutions. Zhou et al. [88] remark that inconsistent evaluation procedures can result in disparate performance results. This subsection briefly surveys the preferred validation and evaluation metrics in the literature.

6.1 Validation Procedures

Splitting the dataset into a train, validation and test set is a ubiquitous practice in the evaluation of non-linear models. Whilst the precise ratio differs in the literature recent contributions [79] [78] [74] [8] use a 60:20:20 split of the dataset. The purpose of the validation set is to optimise the model hyperparameters before evaluating the out-of-sample performance on the test set. Conventional validation procedures are not suited to time series prediction due to their failure to preserve temporal dependencies [38]. Walk-forward validation is the preferred procedure that uses a sliding window over the dataset to partition the data. There are two variants of walk-forward validation widely used in the literature [33]: rolling and successive. Rolling validation uses distinct segments of the dataset for training and validation whilst consecutive validation procedures use overlapping subsets, incrementally increasing the window size and concurrently the training set.

6.2 Evaluation Metrics

Conventional evaluation metrics can be subdivided into two categories: classification and regression. These metric classes corresponds to the associated prediction task. In their review, Sezer et al. [61] identifies the prevalent use of the Accuracy, Precision and Recall metrics in the literature. Our survey of the literature finds that Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are the most commonly used regression metrics.

Recent literature [23] [48] has supported the suitability of absolute error metrics. MAPE and MAE are generically expressed metrics that have superior interpretability, motivating its use. Note that RMSE is sensitive to large prediction errors by design, which can be undesirable in certain instances. Financial forecasting models also supplement standard measures with profit metrics [47] to validate performance. Sezer et al. [61] and Jiang [33] find that Cumulative Return, Excess Return and Sharpe Ratio [62] metrics are overwhelmingly selected. These metrics are also commonly used in the literature for DNN-based portfolio management [70] [76].

Table 1: Comparison of GNN Architectures

Reference	Architecture	Type	Evaluation Metrics	Applications	Strengths	Weaknesses
[79]	Graph WaveNet	Spatial-Temporal	MAPE, MAE, RMSE	Traffic	Learns graph structure automatically Accepts prior knowledge	Reduced accuracy without prior graph structure Unproven generality
[63]	Graph WaveNet++	Spatial-Temporal	MAPE, MAE, RMSE	Traffic	Improved accuracy	Reduced computational efficiency
[78]	MTGNN	Spatial-Temporal	MAPE, MAE, RMSE	Traffic, Solar, Electricity, FOREX	Learns graph structure automatically Accepts prior knowledge Supports uni- and multivariate series	Outperformed for multi-step-ahead prediction
[8]	StemGNN	Spectral-Temporal	MAPE, MAE, RMSE	Traffic, Solar, Electricity, ECG	Learns graph structure automatically Interpretable extracted graphs Supports uni- and multivariate series	Cannot accept prior knowledge

In their seminal paper on GNNs, Scarselli et al. [60] select Accuracy and Relative Error to measure the predictive performance of the models. In contrast to earlier work, state-of-the-art architectures, including GWN [79], MTGNN [78], Auto-STGCN [74] and StemGNN [8] use MAPE, RMSE and MAE metrics in unison. Multiple evaluation metrics support a robust evaluation of the model against benchmarks and other state-of-the-art models and can mitigate the impact of sensitive metrics on the experimental results.

7 CONCLUSIONS

This paper has reviewed several areas that are both implicitly and explicitly related to our research objectives. The EMH [18] can be described as the impetus for portfolio management but not as its theoretical underpinning. Rather, we find the emergence of counter-arguments have proved the existence of market inefficiencies [58] [51]. It is these structural and information inefficiencies in financial markets that provide economic opportunities for individuals to generate a return on investment through portfolio management. We identified the development of contrasting investing strategies with distinct methodologies. Furthermore, the literature has illustrated that comparatively higher returns are earned using value investing. This finding supports an underlying value investing strategy for future intelligent systems.

We discussed time series prediction tasks and reviewed non-linear deep learning architectures for general forecasting. We found that deep learning models have achieved a high degree of accuracy for prediction tasks in several domains [20], although the literature does not exemplify a singular model class that outperforms other techniques. Existing research has focused on multiple approaches, including hybrid architectures which have achieved state-of-the-art [77]. Considering deep learning architectures in the context of portfolio management, we categorised techniques by sub-problem. We found that RNNs, CNNs and hybrid models are prevalent techniques for price prediction problems; this finding is unsurprising given that RNN and LSTM models are designed to learn temporal dependencies. The reviewed studies indicate that RL-based frameworks [34] [35] are the predominant technique for portfolio selection, which we note is consistent with its formulation as an optimisation problem. We identified solutions [25] [82] to share evaluation, although the research did not comprehensively evaluation the models nor demonstrate generality. We also reviewed literature [70] [76] that proposed a unified framework for portfolio management but uses classical optimisation methods for portfolio selection rather than DNNs. The contributions are notable for evaluating shares based on their predicted returns. This is a promising approach for future research to investigate.

GNN models are a successful class of techniques for modelling data with explicit or latent non-temporal dependencies. Our review identified and compared three state-of-the-art architectures: GWN [79], MTGNN [78] and StemGNN [8]. The studies comprehensively documented the experimental methodology and results. We remark that the authors are justified in their performance claims based on the presented evidence. However, their performance is untested on stock market time series, which presents a significant opportunity. We surmise that existing research has focused on explicit relational structured data that illustrate the advantages of GNN models more clearly. We assessed the applicability of the identified GNN architectures through a comparison of their features and evaluation methodologies. We posit that they are applicable to the price prediction problem, and by extension, share evaluation [49]. Also, we found that current architectures are evaluated using identical metric sets, which develops a standard for future research to be appropriately evaluated.

In conclusion, the literature has evidenced that opportunities for profiting from financial markets are consistently present. Our review has demonstrated that deep learning approaches are state-of-the-art for portfolio management sub-problems but have been primarily developed in isolation rather than in an end-to-end framework. We have identified gaps in the state-of-the-art; there are few unified DNN-based solutions for automated share evaluation and portfolio selection and an absence of research investigating GNN-based solutions. Possible future directions for research should investigate the applicability and stability of GNN models to non-structured data in the financial domain. This research should focus on stock markets and develop solutions for price prediction and share evaluation. Also, further study is required to explore intelligent DNN-based systems for portfolio management. This research also should consider the inclusion of GNNs in intelligent systems.

REFERENCES

[1] Jeffrey S Abarbanell and Brian J Bushee. 1997. Fundamental analysis, future earnings, and stock prices. *Journal of Accounting Research* 35, 1 (1997), 1–24.

[2] H Kent Baker and John R Nofsinger. 2010. *Behavioral finance: investors, corporations, and markets*. Vol. 6. John Wiley & Sons.

[3] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. 2013. *Machine Learning Strategies for Time Series Forecasting*. Vol. 138.

[4] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control.* John Wiley & Sons.

[5] Leo Breiman. 2001. Random forests. *Machine Learning* 45, 1 (2001), 5–32.

[6] Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems* 30, 1-7 (1998), 107–117.

[7] David P Brown and Robert H Jennings. 1989. On technical analysis. *The Review of Financial Studies* 2, 4 (1989), 527–551.

[8] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Conguri Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. 2021. Spectral Temporal Graph Neural Network for Multivariate Time-series Forecasting. (2021). arXiv:arXiv:2103.07719

[9] Kai Chen, Yi Zhou, and Fangyan Dai. 2015. A LSTM-based method for stock returns prediction: A case study of China stock market. In *2015 IEEE International Conference on Big Data (Big Data)*. IEEE, 2823–2824.

[10] Lin Chen, Zhilin Qiao, Minggang Wang, Chao Wang, Ruijin Du, and Harry Eugene Stanley. 2018. Which Artificial Intelligence Algorithm Better Predicts the Chinese Stock Market? *IEEE Access* 6 (2018), 48625–48633.

[11] Weiqi Chen, Ling Chen, Yu Xie, Wei Cao, Yusong Gao, and Xiaojie Feng. 2020. Multi-Range Attentive Bicomponent Graph Convolutional Network for Traffic Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3529–3536.

[12] Haibin Cheng, Pang-Ning Tan, Jing Gao, and Jerry Scripps. 2006. Multistep-ahead time series prediction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 765–774.

[13] Rajashree Dash, P.K. Dash, and Ranjeeta Bisoi. 2014. A self adaptive differential harmony search based optimized extreme learning machine for financial time series prediction. *Swarm and Evolutionary Computation* 19 (2014), 25–42.

[14] Werner FM De Bondt and Richard Thaler. 1985. Does the stock market overreact? *The Journal of Finance* 40, 3 (1985), 793–805.

[15] Rachel Drake. 2020. *A semantic Bayesian network for automated share evaluation on the JSE*. Master's thesis. University of Cape Town.

[16] Robert D Edwards, John Magee, and WH Charles Bassetti. 2018. *Technical analysis of stock trends*. CRC press.

[17] Stefan Falkner, Aaron Klein, and Frank Hutter. 2018. BOHB: Robust and efficient hyperparameter optimization at scale. In *International Conference on Machine Learning*. PMLR, 1437–1446.

[18] Eugene F Fama. 1995. Random walks in stock market prices. *Financial Analysts Journal* 51, 1 (1995), 75–80.

[19] Eugene F Fama and Kenneth R French. 1992. The cross-section of expected stock returns. *The Journal of Finance* 47, 2 (1992), 427–465.

[20] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery* 33, 4 (2019), 917–963.

[21] Alberto Fernández and Sergio Gómez. 2007. Portfolio selection using neural networks. *Computers & Operations Research* 34, 4 (2007), 1177–1191.

[22] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and Robust Automated Machine Learning. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Vol. 2. MIT Press, 2755–2763.

[23] Germain Forestier, François Petitjean, Hoang Anh Dau, Geoffrey I Webb, and Eamonn Keogh. 2017. Generating Synthetic Time Series to Augment Sparse Datasets. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 865–870.

[24] Paolo Frasconi, Marco Gori, and Alessandro Sperduti. 1998. A general framework for adaptive processing of data structures. *IEEE Transactions on Neural Networks* 9, 5 (1998), 768–786.

[25] Xingyu Fu, Jin-Hong Du, Yifeng Guo, MingWen Liu, Tao Dong, and Xiuwen Duan. 2018. A machine learning framework for stock selection. (2018). arXiv:arXiv:1806.01743

[26] John Cristian Borges Gamboa. 2017. Deep learning for time-series analysis. (2017). arXiv:arXiv preprint arXiv:1701.01887

[27] Larry R Gorman, Steven G Sapra, and Robert A Weigand. 2010. The cross-sectional dispersion of stock returns, alpha, and the information ratio. *The Journal of Investing* 19, 3 (2010), 113–127.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.

[29] M Hiransha, E Ab Gopalakrishnan, Vijay Krishna Menon, and KP Soman. 2018. NSE Stock Market Prediction Using Deep-Learning Models. *Procedia Computer Science* 132 (2018), 1351–1362.

[30] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. 2018. Relation Networks for Object Detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 3588–3597.

[31] Gang Huang, Xiaohua Zhou, and Qingyang Song. 2020. Deep reinforcement learning for portfolio management. (2020). arXiv:arXiv:2012.13773

[32] Michael C Jensen. 1978. Some anomalous evidence regarding market efficiency. *Journal of financial economics* 6, 2/3 (1978), 95–101.

[33] Weiwei Jiang. 2020. Applications of deep learning in stock market prediction: recent progress. (2020). arXiv:arXiv:2003.01859

[34] Zhengyao Jiang and Jinjun Liang. 2017. Cryptocurrency portfolio management with deep reinforcement learning. In *2017 Intelligent Systems Conference (IntelliSys)*. IEEE, 905–913.

[35] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. 2017. A deep reinforcement learning framework for the financial portfolio management problem. (2017). arXiv:arXiv:1706.10059

[36] Con Keating and William F Shadwick. 2002. An introduction to omega. *AIMA Newsletter* (2002).

[37] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. 2001. An online algorithm for segmenting time series. In *Proceedings 2001 IEEE International Conference on Data Mining*. IEEE, 289–296.

[38] Kouame Hermann Kouassi and Deshendran Moodley. 2020. An Analysis of Deep Neural Networks for Predicting Trends in Time Series Data. In *Artificial Intelligence Research*. Springer International Publishing, 119–140.

[39] Kouame Hermann Kouassi and Deshendran Moodley. 2020. Automatic deep learning for trend prediction in time series data. (2020). arXiv:arXiv:2009.08510

[40] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems* 25 (2012), 1097–1105.

[41] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep Learning. *Nature* 521, 7553 (2015), 436–444.

[42] Diego León, Arbey Aragón, Javier Sandoval, Germán Hernández, Andrés Arévalo, and Jaime Niño. 2017. Clustering algorithms for risk-adjusted portfolio construction. *Procedia Computer Science* 108 (2017), 1334–1343.

[43] Baruch Lev and Anup Srivastava. 2019. Explaining the recent failure of value investing. *NYU Stern School of Business* (2019).

[44] Wei Li, Ruihan Bao, Keiko Harimoto, Deli Chen, Jingjing Xu, and Qi Su. 2020. Modeling the Stock Relation with Graph Network for Overnight Stock Movement Prediction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. International Joint Conferences on Artificial Intelligence Organization, 4541–4547.

[45] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations (ICLR '18)*.

[46] Tao Lin, Tian Guo, and Karl Aberer. 2017. Hybrid neural networks for learning the trend in time series. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. International Joint Conferences on Artificial Intelligence Organization, 2273–2279.

[47] Wen Long, Zhichen Lu, and Lingxiao Cui. 2019. Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems* 164 (2019), 163–173.

[48] Benjamin Lucas, Ahmed Shifaz, Charlotte Pelletier, Lachlan O'Neill, Nayyar Zaidi, Bart Goethals, François Petitjean, and Geoffrey I Webb. 2019. Proximity forest: an effective and scalable distance-based classifier for time series. *Data Mining and Knowledge Discovery* 33, 3 (2019), 607–635.

[49] Yilin Ma, Ruizhu Han, and Weizhong Wang. 2021. Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications* 165 (2021), 113973.

[50] Burton G Malkiel. 2003. The efficient market hypothesis and its critics. *Journal of Economic Perspectives* 17, 1 (2003), 59–82.

[51] Burton G Malkiel. 2011. The Efficient-Market Hypothesis and the Financial Crisis. (2011).

[52] Harry Markowitz. 1952. Portfolio Selection. *The Journal of Finance* 7, 1 (1952), 77–91.

[53] Seyedali Mirjalili. 2019. Genetic algorithm. In *Evolutionary algorithms and neural networks*. Springer, 43–55.

[54] Tom M Mitchell. 1997. *Machine Learning*. McGraw-Hill.

[55] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. (2016). arXiv:arXiv:1609.03499

[56] Antonio Rafael Sabino Parmezan, Vinicius MA Souza, and Gustavo EAPA Batista. 2019. Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information sciences* 484 (2019), 302–337.

[57] Emanuel Parzen et al. 1961. An approach to time series analysis. *Annals of Mathematical Statistics* 32, 4 (1961), 951–989.

[58] Edgar Peters. 2003. Simple and complex market inefficiencies: integrating efficient markets, behavioral finance, and complexity. *The Journal of Behavioral Finance* 4, 4 (2003), 225–233.

[59] Yuji Roh, Geon Heo, and Steven Euijong Whang. 2021. A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective. *IEEE Transactions on Knowledge and Data Engineering* 33, 4 (2021), 1328–1347.

[60] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE Transactions on Neural Networks* 20, 1 (2008), 61–80.

[61] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing* 90 (2020), 106181.

[62] William F Sharpe. 1966. Mutual fund performance. *The Journal of Business* 39, 1 (1966), 119–138.

[63] Sam Shleifer, Clara McCreery, and Vamsi Chitters. 2019. Incrementally Improving Graph WaveNet Performance on Traffic Prediction. (2019). arXiv:arXiv:1912.07390

[64] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine* 30, 3 (2013), 83–98.

[65] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2020. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 914–921.

[66] Alessandro Sperduti and Antonina Starita. 1997. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks* 8, 3 (1997), 714–735.

[67] Jaap Spronk and Winfried Hallerbach. 1997. Financial modelling: Where to go? With an illustration for portfolio management. *European Journal of Operational Research* 99, 1 (1997), 113–125.

[68] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. [n.d.]. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[69] Csaba Szepesvári. 2010. Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 4, 1 (2010), 1–103.

[70] Van-Dai Ta, Chuan-Ming Liu, and Direselign Addis Tadesse. 2020. Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. *Applied Sciences* 10, 2 (2020), 437.

[71] Ahmed Tealab. 2018. Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal* 3, 2 (2018), 334–340.

[72] Ka Ho Tsang and Hoi Ying Wong. 2020. Deep-learning solution to portfolio selection with serially dependent returns. *SIAM Journal on Financial Mathematics* 11, 2 (2020), 593–619.

[73] Ah Chung Tsoi, Gianni Morini, Franco Scarselli, Markus Hagenbuchner, and Marco Maggini. 2003. Adaptive ranking of web pages. In *Proceedings of the 12th International Conference on World Wide Web*. 356–365.

[74] Chunnan Wang, Kaixin Zhang, Hongzhi Wang, and Bozhou Chen. 2020. Auto-STGCN: Autonomous Spatial-Temporal Graph Convolutional Network Search Based on Reinforcement Learning and Existing Research Results. (2020). arXiv:arXiv:2010.07474

[75] Haoran Wang and Xun Yu Zhou. 2020. Continuous-time mean–variance portfolio selection: A reinforcement learning framework. *Mathematical Finance* 30, 4 (2020), 1273–1308.

[76] Wuyu Wang, Weizi Li, Ning Zhang, and Kecheng Liu. 2020. Portfolio formation with preselection using deep learning from long-term financial data. *Expert Systems with Applications* 143 (2020), 113042.

[77] Zhiguang Wang, Weizhong Yan, and Tim Oates. 2017. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1578–1585.

[78] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 753–763.

[79] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 1907–1913.

[80] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. 2019. Spatial-Temporal Relation Networks for Multi-Object Tracking. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 3988–3998.

[81] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[82] Fengmei Yang, Zhiwen Chen, Jingjing Li, and Ling Tang. 2019. A novel hybrid stock selection method with stock prediction. *Applied Soft Computing* 80 (2019), 820–831.

[83] Jihoon Yang and Vasant Honavar. 1998. Feature subset selection using a genetic algorithm. In *Feature extraction, construction and selection*. Springer, 117–136.

[84] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7370–7377.

[85] Yunan Ye, Hengzhi Pei, Boxin Wang, Pin-Yu Chen, Yada Zhu, Ju Xiao, and Bo Li. 2020. Reinforcement-learning Based Portfolio Management with Augmented Asset Movement Prediction States. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1112–1119.

[86] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 3634–3640.

[87] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. Gman: A Graph Multi-Attention Network for Traffic Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1234–1241.

[88] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81.