Investigating the Application of Graph Neural Networks to Stock Market Prediction

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

ABSTRACT

Stock markets are dynamic systems that exhibit both temporal and non-temporal dependencies. The recent emergence of spatial-temporal graph neural networks allows for modelling multivariate time series data generated from systems with explicit or latent structure. This research addresses two weaknesses in the literature by investigating the applicability of spatial-temporal graph neural networks for Johannesburg Stock Exchange price prediction and the suitability of a correlation matrix to encode prior structural information for forecasting tasks. Our results demonstrate that the evaluated graph neural network models achieve suitable performance for forecasting tasks over variable prediction horizons. However, the results exemplify that inter-stock correlations do not accurately capture stock market dependencies, and further, a correlation matrix is an unsuitable encoding of prior information.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**.

KEYWORDS

price prediction, Johannesburg Stock Exchange, graph neural networks, correlation matrix

1 INTRODUCTION

Graph neural networks (GNN) are a class of deep neural network (DNN) models that process graphically-encoded multivariate data. Spatial-temporal GNNs are a variant that model multivariate data exhibiting both spatial and temporal dependencies. This data forms a spatial-temporal graph that consists of a set of nodes, each with dynamic node-level temporal inputs [27] and a predefined graph structure. The existing literature has demonstrated GNN applications to traffic network, energy and medical datasets, but there is a notable deficit of price prediction studies.

A stock market is a financial market that facilitates the exchange of ownership shares of listed financial securities [7]. Stock markets are dynamic systems that can exhibit linear and non-linear dependencies. Pair-wise price correlations between shares typify explicit linear dependencies. A correlation matrix provides a mechanism for encoding these multivariate correlations in a graphical structure. Furthermore, recent studies [22] [15] that evaluate DNNs have illustrated that price forecasting can function as a proxy for identifying suitable securities for investment, a process termed share evaluation.

The existing literature naively presumes that the graph reflects concrete dependencies [27]. Furthermore, the literature assumes prior knowledge of the graph structure and the proposed models require predefined structure, an unsuitable prerequisite for stock market prediction. In this paper, we address two principal weaknesses in the literature. Firstly, we evaluate the applicability of three state-of-the-art GNN architectures for Johannesburg Stock Exchange (JSE)-listed share price prediction. These architectures are applied to prediction problems in other domains, but without evidence, we cannot infer their stock market predictive performance. Secondly, we assess the suitability of a correlation matrix to capture market dependencies and encode structural information and test the hypothesis that multivariate correlations can improve share price prediction accuracy.

2 BACKGROUND AND RELATED WORK

2.1 Problem Formulation

A time series is a sequence of real-valued observations ordered in time. Formally, a univariate time series is a set of random variables $\{X_t, t \in T\}$, where $T = \{1, 2, ..., M\}$. A multivariate time series $\mathbf{X} \in \mathbb{R}^{N \times M}$ is defined as a set of N univariate series.

Mathematically, a graph is a pair $G = (V, E)$, where V denotes the set of nodes and E the set of edges $e = (v, u)$. A feature vector X_v is associated with each node $v \in V$. The neighborhood of a node is defined as $N(v) = \{v \in V \mid (v, u) \in E\}$. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ is a mathematical representation of a graph G, with $A_{ij} > 0$ for $(v_i, v_j) \in E$, $A_{ij} = 0$ for $(v_i, v_j) \notin E$ and N denotes the number of nodes.

In a stock market multivariate spatial-temporal graph, each node v is a stock, and the feature vector X_v corresponds to univariate share price time series input. The graph represents latent stock market structure.

Node degree is the number of connected node edges. *Betweenness centrality* is a measure of a node's occurrence frequency on the pair-wise geodesic. *Closeness centrality* is a measure of a node's average inverse distance to all other nodes. *Transitivity* is a measure of interconnected neighbourhood nodes.

Multivariate single-step forecasting is the task of predicting the value of a single future set of daily share prices (node values) in a spatial-temporal graph conditional on the historical observations. Let $\mathbf{z_t} \in \mathbb{R}^N$ denote a N-dimensional variable at time t. Given $\mathbf{X} = \{\mathbf{z_1}, \mathbf{z_2}, ..., \mathbf{z_t}\}$, the aim is to predict the single-step-ahead vector of node values $\mathbf{Y} = \{\mathbf{\hat{z}_{t+1}}\}$.

Multivariate multi-step forecasting is the task of predicting a sequence of daily share price values conditional on the historical observations. Given observed values $\mathbf{X} = \{\mathbf{z_1}, \mathbf{z_2}, ..., \mathbf{z_t}\}$, the aim is to predict the sequence $\mathbf{Y} = \{\mathbf{\hat{z}_{t+1}}, \mathbf{\hat{z}_{t+2}}, ..., \mathbf{\hat{z}_{t+H}}\}$, where H is the next H time-steps, termed the prediction horizon.

2.2 Deep Neural Network Price Prediction

Deep neural networks (DNNs) have superseded classical machine learning models as the state-of-the-art for recognition, detection and prediction problems in varying domains. DNNs are multi-layer computational processing models that effectively capture latent data features at multiple abstraction levels [12].

Sezer et al. [19] find that the majority of recent research investigating financial time series forecasting has focused on recurrent neural network (RNN) or Long short-term memory (LSTM) models, with consistent applications to different financial markets. The literature has also illustrated the increased prevalence of hybrid models that combine RNNs with an LSTM or RNNs combined with convolutional neural networks (CNN). Tealab [23] finds that hybrid models produced more accurate predictions compared to standalone models, although the author does not conduct an exhaustive review.

In an applied portfolio management context, Ta et al. [22] and Wang et al. [25] apply an LSTM to identify investable financial securities based on the forecasted price, using a composite of price and fundamental data. Building on this methodology, Ma et al. [15] diverge by testing the predictive performance of DNNs using only historical raw price data.

Bontempi et al. [1] remark on the importance of the prediction horizon length in forecasting tasks. The existing literature has evidenced that the investigated models are applied to both single- [5] and multi-step [11] [4] forecasting tasks. However, multi-step forecasting tasks involve additional complexity due to error accumulation. Error accumulation is the propagation of past error into future predictions [6], thereby decreasing predictive accuracy.

2.3 Graph Neural Networks

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. [18] to introduce the graph neural network (GNN) model class that extends existing supervised DNN methods to process graphically-encoded non-Euclidean data. GNNs assume that the node state is dependant on its neighbourhood state [26].

Historically, RNNs [9] [21] and Markov chains [24] were applied to graph-based problems, notably exploited by Brin and Page [2] for search engine creation. Zhou et al. [31] note 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, the authors state that the recent advances in deep learning have supported the development of GNN-based architectures. Wu et al. attribute the success of GNN models to their properties of permutation-invariance, local connectivity and compositionality [26].

GNNs 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 value is $Y_v = f(H_v)$ [28]. Graph classification tasks instead learn a representation vector H_G and a function g to predict the graph label Y_G [28]. The function g is independent of any node $v \in V$ and thus unconditional on the node-level properties of the graph.

2.4 Spatial-Temporal Graph Neural Networks

RNN-based approaches [14] are vulnerable to the vanishing gradient problem [10] and ineffective on long-range temporal sequences. Diffusion Convolutional Recurrent Neural Network [14] is a hybrid architecture that extracts spatial information and passes the output to an RNN to learn temporal dependencies [31]. CNN models can be adapted to learn temporal dependencies using one-dimensional convolutions but require an exceptionally deep network for learning to be effective [27].

Spatial-temporal GNNs are explicitly designed to model spatial and temporal dependencies in a system. Spatial and temporal components extract underlying patterns from the data in the corresponding domains. The spatial relations are reflected by the graph structure, with dynamic node-level inputs exhibiting one-dimensional temporal dependencies [27]. Spatial-temporal graph convolutional networks [30] [20] are CNN-based architectures frequently applied to spatial-temporal graph modelling tasks. However, Wu et al. state that previously introduced spatial-temporal GNNs are not suitable for modelling multivariate time series due to two distinct factors: unavailability of prior information and the sub-optimality of the predefined graph [26].

In non-idealised systems, a dependency can exist if an edge is absent from the graph, nor does an edge always imply an inter-node dependency. However, previous contributions evaluating spatial-temporal GNNs incorrectly assume that the graph is a perfect representation of dependencies [27]. Prior knowledge of the spatial dependencies is further assumed, and the models rely on a predefined fixed graph structure for training. This approach is unsuitable for stock market systems with complex dependencies and latent structure.

Graph WaveNet (GWN) [27], MTGNN [26] and Spectral-Temporal Graph Neural Network (StemGNN) [3] are architectures that overcome the aforementioned limitations. A comparative analysis is summarised in table 1. GWN, MTGNN and StemGNN have achieved the state-of-the-art or parity for single-step and multi-step forecasting task performance on the evaluated datasets.

Whilst the MTGNN and StemGNN domain applications prove generalisability, neither architecture is tested on stock market data. GWN is evaluated only on traffic network datasets, and therefore, we cannot infer generalisability. An exchange-rate time series is used to evaluate MTGNN's performance, although the foreign exchange and stock markets exhibit heterogeneous dynamics. MTGNN and StemGNN outperform GWN on traffic network datasets, but MTGNN is notably absent from the StemGNN performance evaluation.

A graph convolution [8] component is uniformly integrated to extract spatial structure. However, in contrast to GWN and MTGNN, StemGNN extracts dependencies in the spectral domain. The reported experimental results [3] suggest that this does not affect its applicability to stock market prediction. GWN and MTGNN both utilise a temporal convolution component to learn temporal dependencies within the time series.

GWN, MTGNN and StemGNN all can adaptively learn the graph structure without the provision of prior information. StemGNN and MTGNN outperform spatial-temporal GNNs that are initialised with fixed graph structures. GWN similarly outperforms the evaluated models. However, GWN's predictive performance decreases without prior structural information. A notable weakness of StemGNN is that in its explicit focus on general modelling cases, it fails to accommodate available prior information, whilst GWN and MTGNN are flexible and accept adjacency matrix initialisations.

2.5 Graph Structure Encoding

GNN applications can be categorised into structural and non-structural applications [31]. Structural applications involve data that exhibits an explicit relational structure. Non-structural applications involve data that has implicit or hidden relationships. Stock market data is an example of non-structured data in which modelling does not expose the relational schema, although the data can be transformed to formulate a structural problem. Zhou et al. [31] review illustrates that the existing literature seldom explores GNN applications [29] [16] to stock market prediction.

LSTM Relational Graph Convolutional Network (LSTM-RGCN) [13] is a GNN architecture that exploits inter-stock linear correlations to capture market dependencies and encode them in a graphical structure. The RCGN component extracts spatial dependencies, whilst an LSTM functions as a news text encoder and dynamic information propagation mechanism between RGCN layers. However, LSTM-RGCN notably does not incorporate a temporal component, with the authors formulating the problem as dual-class movement prediction instead of price prediction.

A correlation matrix is calculated using historical price data under the assumption of share inter-dependency, thus positing that the matrix is representative of the latent market structure. Li et al. motivate their choice by noting that historical market data encompasses prior market movements [13]. The correlation matrix functions as a proxy for prior structural information and initialises the graph adjacency matrix. The authors constrain their evaluation of LSTM-RGCN to Chinese stock market data. LSTM-RGCN outperforms the evaluated models but produces a relatively poor accuracy rate.

3 METHODOLOGY

3.1 Models

In this research, we evaluate GWN [27], MTGNN [26] and StemGNN [3] for stock market price prediction. Each model is composed using the PyTorch machine learning library, based on the authors' implementations.

3.1.1 Graph WaveNet. GWN is a hybrid architecture for spatial-temporal graph modelling. The GWN architecture consists of temporal convolution (Gated TCN) and graph convolution (GCN) modules. The GCN module contains a self-adaptive adjacency matrix that requires no prior information and is learnt using stochastic gradient descent. The Gated TCN module consists of one-dimensional convolutions that capture long-range temporal sequences [17] whilst avoiding the vanishing gradient problem [10]. The GCN module extracts node-level dependencies using neighbourhood feature aggregation [28]. The model outputs the predicted

Table 1: Comparison of selected graph neural network architectures

Architecture	Type	Temporal Component.	Spatial Component	Datasets (MAPE%)	Strengths	Weaknesses
Graph WaveNet [27]	Spatial-Temporal	Gated Temporal (Gated TCN) Convolution	Graph Convolution (GCN)	Traffic (8.23%)	Learns graph structure adaptively Accepts prior information	Sub-optimal adaptive structure Unproven generality
MTGNN [26]	Spatial-Temporal	Temporal Convolution	Graph Convolution (GCN)	Traffic (5.18%) Electricity (-) Solar (-) Forex (-)	Learns graph structure adaptively Accepts prior information Proven generality	Outperformed on multi-step tasks
StemGNN[3]	Spectral-Temporal	Spectral-Sequential Cell (Spe-Seq Cell)	Spectral Graph Convolution (Spectral GCN)	Traffic (6.46%) Electricity (14.77%) Solar (11.55%) ECG (10.58%)	Learns graph structure adaptively Interpretable adaptive graph Proven generality	Cannot accept prior information

sequence over the entire prediction horizon H instead of iteratively generating H conditioned predictions. Following the authors' [27] adjacency matrix configuration results, we evaluate a double transition matrix plus adaptive adjacency matrix using the structural information initialisation against the adaptive-only adjacency matrix.

3.1.2 MTGNN. MTGNN is a hybrid architecture designed with an explicit focus on multivariate time series forecasting. MTGNN can accommodate unavailable prior information through an adaptive adjacency matrix, although the model does not update the structure during training. MTGNN consists of a distinct graph learning layer that extracts the adjacency matrix. The GCN module utilises a neighbourhood aggregation strategy [28] to learn node-level spatial dependencies and the TCN module extracts temporal dependencies. MTGNN avoids the vanishing gradient problem [10] by including residual and skip intra- and inter-layer connections. The model is trained using a curriculum learning strategy that splits the input into subgroups. Curriculum learning locates optimal local minima by training the algorithm on a single-step forecasting task first and subsequently increases the prediction horizon at each iteration [26]. We evaluate a predefined static graph structure against the adaptive-only adjacency matrix.

3.1.3 StemGNN. StemGNN is a hybrid architecture for multivariate time series forecasting that captures inter-series correlations and temporal dependencies in the spectral domain. The spectral GCN component analogously extracts dependencies in the spectral rather than spatial domain. Spectral-Temporal GNN models are trained on the spectral representation of the graph [31] using graph signal processing techniques. 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 [31]. StemGNN consists of a latent correlation layer to automatically learn correlations between time series without a predefined graph structure to generate an adjacency matrix. For our evaluation, StemGNN is not initialised with prior information by design.

3.2 Experimental Design

3.2.1 Data. The performances of the GNN models are compared on daily close price data for FTSE/JSE Top 40 Index constituent shares from 18 May 2009 to 20 July 2021. The Top 40 Index contains the 40 largest JSE-listed companies by market capitalisation. Market capitalisation is the current value of all outstanding shares. Companies listed after 2012 are excluded such that sufficient training data is available. The final dataset consists of 30 nodes (stocks) and 3146 samples. For GWN and MTGNN, the data is further pre-processed following Wu et al. [27] and Wu et al. [26] to generate a four-dimensional dataset. The dataset is augmented by aggregating samples into windows of a specified size to construct synthetic features. The data is standardised using Z-score normalisation that removes the mean and re-scales to unit variance. The dataset is split in chronological order to preserve temporal dependencies with 60% for training, 20% for validation and 20% for testing (2019 - 2021).

As a proxy for prior information, a statistical correlation matrix (Figure 1) is calculated using the historical daily close prices of the 30 companies in the training dataset partition and initialises the adjacency matrix of GWN and MTGNN pre-training. The static correlation matrix captures potential inter-share dependencies in the market and encodes this information in a graphical structure that is mathematically equivalent to the adjacency matrix.

3.2.2 Baseline. Graph WaveNet, MTGNN and StemGNN are baselined against a 100-layer LSTM model. The baseline LSTM model outputs a forecast over the prediction horizon for a single node, in contrast to the GNN models that produce multi-node predictions for the entire horizon in a single run. For single-step forecasting, a last-value or naive model is selected in addition to the LSTM model to produce baseline performance results. For any time series $X = \{x_1, x_2, x_3, ...x_t\}$, a naive model outputs the last sequential observation x_t as the forecasted series value x_{t+1}. The baseline models are implemented using the PyTorch machine learning library.

3.2.3 Setup. Each experiment is conducted on an Apple MacBook Pro with an Intel(R) Core(TM) i5-8257U CPU @ 1.4 GHz. Following Cao et al. [3], the models are trained using

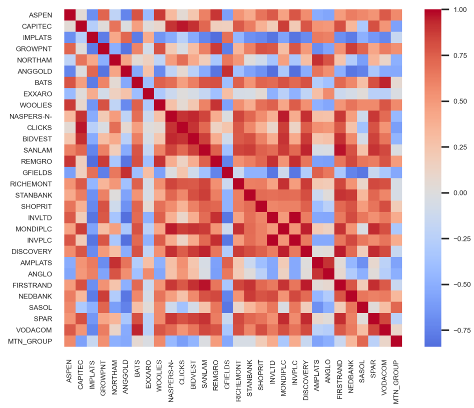

Figure 1: FTSE/JSE Top 40 Index correlation matrix

RMSProp optimiser and Mean Squared Error loss function for 50 epochs. The initial learning rate is set to 0.001 with a decay and dropout rate of 0.05. All other hyperparameters are configured as reported by the authors [27] [26] [3] for accurate comparative analysis.

The following error metrics are selected to evaluate out-of-sample performance: Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). For multi-step forecasting, these metrics are averaged over H steps corresponding to the prediction horizon and all nodes. Each experiment is run 5 times and the denormalised mean metric value is reported.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \tag{1}$$

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right|} \tag{3}$$

4 RESULTS

4.1 Single-Step Forecasting

Table 2 compares the performance of GWN, MTGNN and StemGNN and the baseline LSTM and Naive models on a single-step forecasting task for 30-, 60- and 120-day window sizes. Regular trading days for the Johannesburg Stock Exchange are Monday through Friday, and thus the selected window sizes correspond to 6, 12 and 24 trading weeks.

The experimental results illustrate that MTGNN outperforms both GWN and StemGNN by a substantial margin across the MAPE, MAE and RMSE metrics for all tested window sizes. In addition, the performance variation between MTGNN and GWN and StemGNN increases as the window size is increased, as MTGNN does not demonstrate similar levels of performance degradation. MTGNN achieves a MAPE score of 10.62% for the 120-day window, whilst the successive model StemGNN only achieves a MAPE score of 23.25%. Further, the results demonstrate that predictive performance of the GNN and baseline models decreases as the window size is increased. Whilst GWN outperforms StemGNN for the 30-day window, the results indicate a sharp decline in GWN's performance as the window size is doubled from 30 to 60.

4.2 Multi-Step Forecasting

Tables 3, 4 and 5 compare the performance of GWN, MTGNN and StemGNN and baseline LSTM on a multi-step forecasting task for 20-, 40- and 60-day window sizes and 5-, 10- and 20-day close price prediction horizons. Comparing MAPE scores, GWN significantly outperforms MTGNN, StemGNN and the LSTM for the tested hyperparameters. However, the results for 60-day window size forecasting deviate and demonstrate that StemGNN achieves marginally lower MAPE scores. Whilst the LSTM obtains consistently lower RMSE scores, this can be attributed to the independently-generated model predictions, which potentially reduces error propagation and hence the RMSE of predictions.

MTGNN is the unexpected laggard for the evaluated multi-step forecasting tasks, demonstrating inferior performance relative to the baseline. However, recall that the LSTM is incapable of multi-node forecasting. The results indicate that a 40-day window considerably improves MTGNN's performance, although MAPE scores for all tested horizons are $> 100\%$.

GWN achieves its best MAPE performance scores for a 20-day window and horizon hyperparameter combination. This is followed closely by the model's predictive performance for a 40-day window and 5-day horizon. With the exception of the aforementioned result, the results illustrate a decline in GNN and baseline model predictive performance as the horizon length is increased, measured across all metrics. Furthermore, the results do not demonstrate a clear relation between window size and prediction error. GWN's performance improves for 5-day and 10-day horizons as the window size is increased from 20 to 40, but the MAPE score significantly decreases once the window is incremented by an additional 20 days.

StemGNN demonstrates inferior performance on the shortest window size, producing superior MAPE scores on a 40-day window. An outlier is its performance for a 40-day window and 20-day horizon, where it outperforms GWN and MTGNN as measured by MAPE. Furthermore, StemGNN's performance is relatively stable as the prediction horizon

Table 2: Single-step forecasting performance comparison of GNNs and baseline models

	30-Day Window			60-Day Window			120-Day Window		
Model	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE
Last-Value	198.33	13252.53	16380.79	201.12	13235.56	16371.01	218.19	13274.57	16458.70
LSTM	49.84	3588.49	4877.43	192.90	12505.11	15455.85	197.15	11847.00	14561.59
Graph WaveNet	10.44	3072.72	8496.79	24.35	5432.03	15325.71	32.90	7603.66	20206.61
MTGNN	**6.76**	**1478.92**	**3511.12**	**9.06**	**1953.99**	**4871.51**	**10.62**	**2472.13**	**6857.16**
StemGNN	18.01	8415.90	21715.64	23.62	9329.78	21636.29	23.25	9249.32	21583.35

increases for both 40- and 60-day windows across all metrics for the tested horizon lengths. This is evidenced by the standard deviation of the MAPE scores, which is 1.22 and 1.31 for the corresponding window sizes.

4.3 Correlation Matrix Multi-Step Forecasting

Tables 6 and 7 compare the performance of GWN and MTGNN on a multi-step forecasting task for 20- and 40- window sizes, 5- and 10-day horizons with the inclusion of the statistical correlation matrix to initialise the GWN and MTGNN adjacency matrices.

The results indicate that the provision of structural information proxied by a multivariate correlation matrix does not improve predictive performance, as evidenced by the MAPE scores. For GWN, the forward-backward-adaptive adjacency matrix configuration obtains the highest MAPE score for a 20-day window only. However, the improvement in MAPE is marginal for the 5- and 10-day horizons and the recorded MAE and RMSE metrics increase as compared to the adaptive-only model. Doubling the window size significantly degrades the performance of the GWN model initialised with the correlation matrix, illustrated by the consistent increase across MAPE, MAE and RMSE. This can be attributed to the temporal divergence between the extended input sequence length and the static multivariate correlations.

The provision of prior information results in a minimal improvement in MTGNN's predictive performance as measured by the selected evaluation metrics for a 20-day window. However, performance similarly degrades as the window size is doubled for the tested horizons. An outlier is MTGNN's performance for a 40-day window and 5-day horizon, for which it obtains a marginally lower MAPE score.

4.4 Multivariate Correlation Analysis

GWN (Figure 2) and MTGNN (Figure 3) extract a graph structure that is highly dissimilar to that represented by the correlation matrix. Comparing the adaptive adjacency matrices, GWN extracts a sparse graphical structure, in contrast to the dense graph learnt by MTGNN. There are no identifiable commonalities between the two matrices, nor is one matrix a more fine-grained representation of the extracted dependencies. Interpreting the adjacency matrices under the hypothesis of multivariate correlations representing graph structure, MTGNN extracts a perfect positive correlation between the majority of pairs. GWN extracts a minimal subset of strongly positive correlations, but these diverge from the pair-wise correlations learnt by the MTGNN model.

The graph representation of each adaptive adjacency matrix is given in Figure 7 and Figure 8 and further highlights the disparity in graph density. Node colour is a gradient scale of node degree, whilst node size illustrates betweenness centrality. Figure 8 illustrates that there are multiple nodes with a relatively high degree which is indicative of salient stocks. For GWN, there are fewer highly connected stocks as expected, with the analysis indicating that Anglo American Platinum, Capitec, and AngloGold Ashanti are the dominant stocks. The MTGNN graph (Figure 8) shows that several nodes frequently occur on the pair-wise geodesic and thus represent highly influential stocks. This finding is consistent with the dense graph structure. On the other hand, Figure 7 illustrates far fewer influential nodes in the GWN graph, with Anglo American Platinum, Capitec, and AngloGold Ashanti providing maximal influence on other graph nodes through strategic placement.

Tables 8 and 9 contain metrics for a simple and hierarchical correlation network constructed from the static correlation matrix. Unsurprisingly, the density, betweenness centrality, closeness centrality, and transitivity metrics are directly proportional to the correlations for each stock, represented by edges. Betweenness centrality peaks for $c = 2$, and then gradually decline as density increases, for both the simple and hierarchical network. Transitivity increases substantially for $c \geq 2$. The metric value is consistent with the number of identified communities, which stabilises at two for $c \geq 2$ in both constructed networks. This aligns with the results of bi-clustering (Figure 4) performed on the correlation matrix, which finds two compact clusters of highly correlated stocks.

Table 3: 20-day window multi-step forecasting performance comparison of GNNs and baseline LSTM model

	5-Day Horizon			10-Day Horizon			20-Day Horizon		
Model	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE
LSTM	46.85	3345.75	**4653.75**	54.97	3912.20	**5408.92**	68.54	4782.45	**6596.19**
Graph WaveNet	**17.87**	**2554.10**	6430.45	**17.27**	**2716.00**	6819.69	**12.17**	**3147.66**	8852.29
MTGNN	216.97	29217.75	54875.26	258.05	31484.04	54808.86	232.43	32251.45	63249.00
StemGNN	24.96	10327.52	24464.46	26.64	10907.52	25375.75	30.92	11992.33	27384.19

Table 4: 40-day window multi-step forecasting performance comparison of GNNs and baseline LSTM model

	5-Day Horizon			10-Day Horizon			20-Day Horizon		
Model	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE
LSTM	63.23	4276.09	**5707.56**	72.85	4813.56	**6402.42**	78.71	5538.65	**7173.87**
Graph WaveNet	**12.44**	**3960.92**	11642.37	**15.35**	**4370.69**	12882.04	25.69	**5022.74**	13472.62
MTGNN	122.63	19905.99	42729.60	152.71	23566.66	49893.09	182.12	29028.59	62179.24
StemGNN	19.78	8640.64	21504.16	19.94	8576.14	21305.53	**21.97**	9287.20	22693.34

Table 5: 60-day window multi-step forecasting performance comparison of GNNs and baseline LSTM model

	5-Day Horizon			10-Day Horizon			20-Day Horizon		
Model	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE
LSTM	57.93	**4054.60**	**5139.31**	68.95	**4808.36**	**6052.57**	80.53	**5666.38**	**7039.22**
Graph WaveNet	25.82	5535.79	15284.90	26.23	5792.11	16287.67	28.82	5998.07	16441.86
MTGNN	269.24	32627.72	59139.12	324.88	34788.54	54687.03	269.80	30684.99	51937.85
StemGNN	**23.10**	9198.82	21732.27	**25.55**	10797.06	26811.51	**23.51**	9604.94	22622.16

Table 6: 20-day window multi-step forecasting performance comparison of GWN and MTGNN with predefined adjacency matrix

		5-Day Horizon			10-Day Horizon		
Model	Adjacency Matrix	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE
Graph WaveNet	Adaptive-only	17.87	**2554.10**	**6430.45**	17.27	**2716.00**	**6819.69**
Graph WaveNet	Forward-backward-adaptive	**15.80**	3237.76	9103.10	**16.59**	3348.49	9104.82
MTGNN	Adaptive-only	216.97	29217.75	54875.26	258.05	31484.04	54808.86
MTGNN	Predefined	214.05	28885.62	54690.14	237.09	30302.03	55173.77

Table 7: 40-day window multi-step forecasting performance comparison of GWN and MTGNN with predefined adjacency matrix

		5-Day Horizon			10-Day Horizon		
Model	Adjacency Matrix	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE
Graph WaveNet	Adaptive-only	**12.44**	**3960.92**	**11642.37**	**15.35**	**4370.69**	**12882.04**
Graph WaveNet	Forward-backward-adaptive	22.25	5150.09	14683.01	20.43	5089.13	14254.43
MTGNN	Adaptive-only	122.63	19905.99	42729.60	152.71	23566.66	49893.09
MTGNN	Predefined	118.53	20303.73	44546.03	192.05	26726.98	52491.44

Figure 5 illustrates that Discovery, Remgro and AngloGold Ashanti are both the most influential and salient stocks in the network. There are several nodes on the edge of the network that are not influential nor important. The network illustrates that only a fractional subset of the total stocks are considered salient, with a larger subset providing a range of influence in the network. The hierarchical network (Figure 6) contains additional influential nodes as a consequence of the denser structure. The network further illustrates that Mondi is the dominant stock, followed by a subset of six stocks that hold diminished, although roughly equal importance.

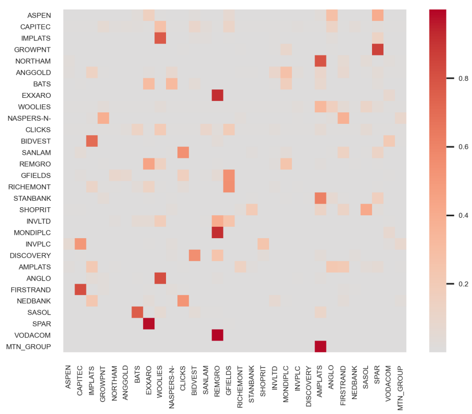

Figure 2: GWN 50-epoch adaptive adjacency matrix

Figure 3: MTGNN 50-epoch adaptive adjacency matrix

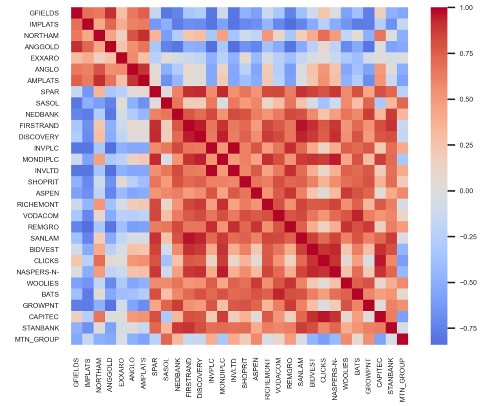

Figure 4: Bi-clustered correlation matrix

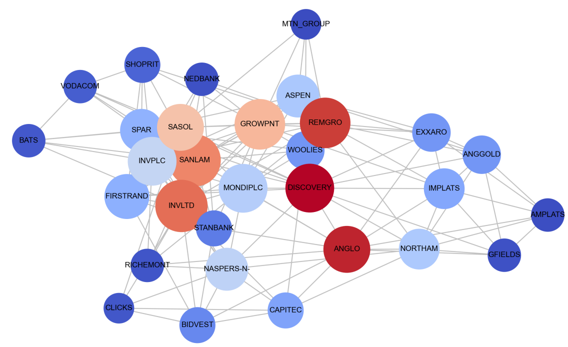

Figure 5: FTSE/JSE Top 40 Index correlation network ($c = 5$)

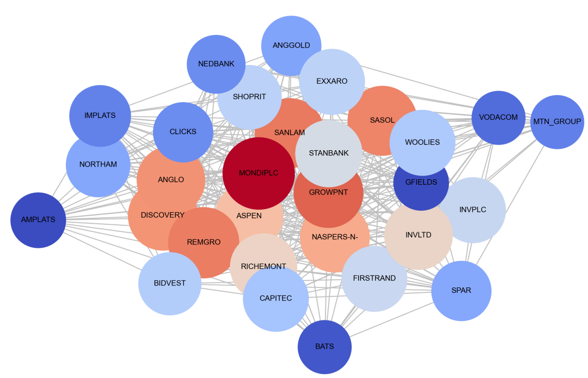

Figure 6: FTSE/JSE Top 40 Index two-level hierarchical correlation network ($c = 5$)

5 DISCUSSION AND CONCLUSIONS

Recently introduced spatial-temporal GNNs are unsuitable for modelling stock market data with complex dependencies due to unavailable prior structural information. Graph WaveNet [27], MTGNN [26] and StemGNN are GNN architectures that solve the requirement of a predefined graph structure by adaptively learning spatial dependencies during training. These architectures have been applied to forecasting problems in several distinct domains and demonstrate reasonable predictive accuracy with reported MAPE scores

Table 8: FTSE/JSE Top 40 Index correlation network metrics

Correlations (c)	Nodes	Edges	Density	Betweenness Centrality	Closeness Centrality	Transitivity	Communities
1	30	30	0.07	0.07	0.16	0.06	3
2	30	56	0.13	0.13	0.36	0.25	2
3	30	85	0.20	0.10	0.47	0.29	2
4	30	113	0.26	0.10	0.52	0.39	2
5	30	139	0.32	0.09	0.57	0.45	2

Table 9: FTSE/JSE Top 40 Index two-level hierarchical correlation network metrics

Correlations (c)	Nodes	Edges	Density	Betweenness Centrality	Closeness Centrality	Transitivity	Communities
1	30	55	0.13	0.05	0.25	0.38	3
2	30	136	0.31	0.10	0.55	0.46	2
3	30	221	0.51	0.08	0.67	0.58	2
4	30	289	0.66	0.08	0.75	0.7	2
5	30	355	0.82	0.07	0.84	0.83	2

of $\leq$ 10% for traffic network datasets. However, stock markets are an application not evaluated in these studies. LSTM-RGCN is a GNN architecture that utilises a correlation matrix to represent spatial dependencies but does not contain a temporal component. LSTM-RGCN only performs price movement prediction and reports an accuracy of 58.71% on Chinese stock market data.

To address these weaknesses in the literature, this research evaluated the application of GWN, MTGNN and StemGNN to single-step and multi-step forecasting tasks on the JSE and tested the effect of prior structural information proxied by a correlation matrix on share price prediction accuracy.

For single-step forecasting, MTGNN outperformed GWN and StemGNN, achieving a significant increase in relative predictive accuracy across varying windows. We suggest that MTGNN's performance is attributable to the curriculum learning strategy. Recall that curriculum learning locates optimal local minima by initially training the algorithm on a single-step forecasting task. An optimal initialisation point may allow MTGNN to update its parameters more effectively as the prediction horizon remains constant.

Whilst the authors [26] fail to report MAPE scores for single-step forecasting, our results are comparable with MTGNN's performance on traffic network multi-step forecasting. This consistency is notable given that traffic networks exhibit explicit spatial relations. However, MTGNN's multi-step forecasting results differ substantially from those presented in the original contribution. Our results suggest that MTGNN cannot accurately forecast a sequence of daily close prices but sufficiently models the data if the problem is constrained to forecasting a single set of future values.

For multi-step forecasting tasks, the results illustrated that GWN achieved the highest predictive accuracy over the tested window sizes and horizon lengths. We posit that GWN's performance on this task is a consequence of the artificially set receptive field size, which is equated to the window size hyperparameter [27] by design. Overall, a 40-day window produces the highest predictive performance for all evaluated GNN models. This window size corresponds to the previous two-months daily close prices and indicates that this range is the optimal input sequence length.

The results of the single- and multi-step forecasting experiments illustrate that a GNN model can achieve suitable predictive accuracy on FTSE/JSE Top 40 Index daily close price data. For both tasks, a GNN model achieves a considerable performance improvement over the baseline LSTM. These results provide further evidence in support of modelling both temporal and latent spatial dependencies in stock market data.

StemGNN obtains MAPE scores that are approximately 10% lower than those reported on electricity, ECG and solar datasets. However, these domains exhibit explicit structure and recall that StemGNN's latent correlation layer adaptively learns inter-series correlations to extract the graph structure without prior information.

Similarly, comparing the best GWN performance score to the metrics reported for traffic network forecasting, we find that the approximately 10% decrease in MAPE is realistic given the dynamic dependencies exhibited by stock markets and inherent complexity in its extraction. In practical applications, tasks are predominately formulated as multi-step forecasting. Therefore, our results demonstrate that GWN can successfully predict future daily close price values over short- and medium-term prediction horizons. Furthermore, they support the authors' proposed architectures and claims of generalisability.

Based on the results of the correlation matrix experiment, we found that this mechanism captures and encodes prior structural information sub-optimally. The *a priori* inclusion of multivariate correlations produced a negligible impact on model predictive accuracy. This result is evidenced by the disparity between the adaptive graph structures of GWN and MTGNN compared to the correlation matrix. The results are further consistent with those presented by Li et al. [13], which illustrated poor stock price movement predictive accuracy. Furthermore, the results show that both adaptive adjacency matrices do not extract linear correlations. We conclude that a static correlation matrix does not accurately capture dynamic latent dependencies or structure.

However, the multi-step forecasting results indicate that the dense adaptive MTGNN graph structure is similarly unrepresentative of the dynamic non-linear market dependencies. In contrast, GWN's share price forecasting performance suggests that the model can accurately extract spatial dependencies. Therefore, we can infer that fewer complex dependencies exist, given that the sparse graph is an approximate representation of market structure. We conclude that GWN can accommodate the unavailability of prior information sufficiently. Our results are consistent with Wu et al. [27] findings which demonstrate that an adaptive-only adjacency matrix achieves suitable performance, albeit with a reduction in accuracy.

In our analysis of the inter-stock correlations, we constructed correlation networks and identified two clusters of highly correlated stocks. This analysis also identified a minimal subset of influential and salient stocks in the market. Further analysis of the correlation networks can produce prior structural information encodings that may capture non-linear dependencies more effectively and thus improve predictive performance.

In summary, this research has addressed the aforementioned gaps in the literature and produced empirical results that assess the applicability of the selected GNN models to JSE-listed share price prediction and the suitability of a correlation matrix proxy for the graph structure. The MTGNN and GWN architectures produced the highest predictive accuracy on a single-step and multi-step forecasting task, demonstrating the suitability of applying GNN models in this domain. We disproved the hypothesised performance increase and determined that multivariate linear correlations do not capture complex stock market dependencies. Our novel analysis of share price correlations highlighted the potential in exploring richer transformations of the data to construct alternative structural information encodings more representative of the linear and non-linear dependencies. Furthermore, the extracted GWN adaptive graph is a state-of-the-art representation of the dependencies in the FTSE/JSE Top 40 Index. This work provides a foundation for future investigation of this model class and real-world applications of share price forecasting.

6 LIMITATIONS AND FUTURE WORK

This research has demonstrated that GNNs can be successfully applied to forecast JSE-listed share prices. The results of this work present several opportunities for further investigation. The selected models were evaluated on a single dataset. Obtaining empirical results on data from different stock exchanges allows for a broader assessment of model generalisability and identifying potential discrepancies in market dynamics. The methodology was restricted to an evaluation of three state-of-the-art models, which can be extended as novel GNN architectures are introduced in the literature. In addition, testing prior structural information sources other than a correlation matrix, or feature combinations thereof, presents a promising avenue for future study.

7 ACKNOWLEDGEMENTS

This research is wholly funded by the National Research Foundation (Grant Number MND200411512622). We acknowledge the foundation for their generous support.

REFERENCES

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

[2] 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.

[3] 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

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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)*.

[15] 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.

[16] Daiki Matsunaga, Toyotaro Suzumura, and Toshihiro Takahashi. 2019. Exploring Graph Neural Networks for Stock Market Predictions with Rolling Window Analysis. (2019). arXiv:arXiv:1909.10660

[17] 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

[18] 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.

[19] 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.

[20] 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.

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

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] Yiying Yang, Zhongyu Wei, Qin Chen, and Libo Wu. 2019. Using External Knowledge for Financial Event Prediction Based on Graph Neural Networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 2161–2164. https://doi.org/10.1145/3357384.3358156

[30] 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.

[31] 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.

A ADAPTIVE GRAPHS

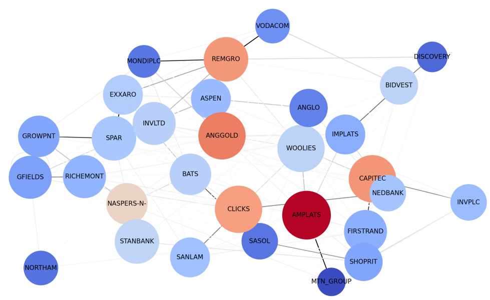

Figure 7: GWN 50-epoch adaptive graph

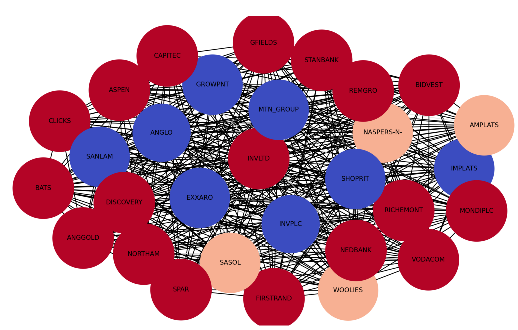

Figure 8: MTGNN 50-epoch adaptive graph