Advanced Machine Learning in Quantitative Finance Using Graph Neural Networks

Mvuleni Kekana, Mbuyu Sumbwanyambe, and Tlotlollo Hlalele *

Department of Electrical Engineering, College of Science, Engineering and Technology,
University of South Africa, Johannesburg, South Africa
Email: 55988024@mylife.unisa.ac.za (M.K.); sumbwm@unisa.ac.za (M.S.); hlalets@unisa.ac.za (T.H.)
*Corresponding author

Abstract—**Given the complexity of financial markets, predicting future prices is a major challenge at present. This paper proposes computational intelligence for stock price forecasting and conducts a preliminary investigation into graph-based neural networks for predicting stock market movement. Predicting stock prices remains a challenging endeavour due to the complex interplay of diverse factors. Traditional machine learning methods often struggle to capture these intricate relationships, as they typically analyse data points in isolation. This research paper aims to investigate the effectiveness of a graph-based neural network for stock price forecasting. Our experiment was carried out using stock data from the Johannesburg Stock Exchange (JSE) sourced from Yahoo finance. The time series data of the closing and opening prices of a Top 40 financial instrument namely the Standard Bank Group (JSE-SBK) instrument. The graph network architecture consists of Convolutional Neural Network (CNN) layers followed by Long Short-Term Memory (LSTM) layers and final dense layers. The graph-based network utilizes the Adaptive Moment Estimation algorithm for optimization during model training. The model performance was validated using a separate test set. This step achieved a resultant model with a prediction variance score of 0.913, which indicates an extremely high level of accuracy in predicting the stock price future behavior. This implies that our model captures over 91% of the variability in the data, which is a strong indication of its reliability.**

Keywords—**machine learning, neural network, quantitative finance**

I. INTRODUCTION

In 1994, South African enterprises operated differently in the economic environment then the year 2023. The transformation was driven by relentless micro and macro-economic reforms, marking a shift in capitalist economies. This period witnessed a transition from traditional industries to dynamic sectors like information and digital technology. As financial growth is the central aim for any enterprise, is bolstered by equity partners who invest in shares traded on platforms like National Association of Securities Dealers Automated Quotation (NASDAQ) and Johannesburg Stock Exchange (JSE). The price of shares, influenced by market dynamics, equity investors scrutinize the ebb and flow of market forces to decipher share prices, ultimately shaping their investment choices [1].

To navigate market entry and exit positions as a participant or investor, practitioners turn to financial analysts for guidance. They rely on mathematical and statistical tools, but traditional techniques are favoured for selecting equity instruments rather than speculative methods. Recently, machine learning has emerged as a powerful tool for data-driven forecasting in finance, poised to revolutionize various aspects of businesses [2]. This prompts an investigation into applying machine learning for financial stock price prediction, aiming to mitigate capital risks and address real-world forecasting challenges. Thus, the need to conduct an investigative study on the application of machine learning for price forecasting of financial stock price [3, 4]. Ultimately, we seek to address how computational intelligence may avert capital risk and loss and the solving price forecasting problems found in the real world. The main contribution of this research is the development of a graph network architecture of convolutional neural network and the adaptive moment estimation algorithm for optimization of model during training.

The rest of the paper is organized as follows: Section II gives the literature review, Section III gives a theoretical background, Section IV details the results and some discussions and finally, Section V gives the conclusion.

II. LITERATURE REVIEW

The recent trends in AI in particularly forecasting has been attributed by the development of advanced hardware and computing technologies that is able to facilitate or deploy AI in a number of different areas. One such area is in the field of quantitative finance [5]. Applications vary in nature. In particular those that are found in domain of quantitative finance.

Fang [6] contrasts statistical analysis: Autoregressive Moving Average (ARMA) Model, Autoregressive Integrated Moving Average (ARIMA) Model, Generalized Autoregressive Conditional Heteroskedasticity (GARCH), Markov Model with machine learning techniques: Recurrent Neural Network (RNN), Fuzzy Neural Network

Manuscript received November 23, 2023; revised April 23, 2024; accepted May 16, 2024; published September 13, 2024.

doi: 10.12720/jait.15.9.1025-1034

(FNN), Long Short Term Memory (LSTM), Support Vector Machine (SVM), Vector Machine-Long Term Short Memory (VM-LSTM), Autoregressive Moving Average-Support Vector Machine (ARMA-SVM) for stock price prediction using Support Vector Regressor (SVR). It evaluates model performance with Mean Squared Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE), highlighting impressive results for deep networks. The research focuses on an improved Support Vector Regression (SVR) model that is based on parameter selection and algorithm optimization. The result is that the predictive performance of this modified SVR model is improved R^2 and RMSE metric. Resulting in good stock price forecasting accuracy. As the SVM which is a widely known machine learning algorithm and statistical modelling technique, suffers from poor fitting, weighted sum of the inputs and inaccurate results. As presented in the limitations of SVM algorithm in [7]. However, there are other. Consequently, financial time series analysis is crucial for managing market risks and making optimal investment decisions. However, it's a complex task due to the presence of diverse data streams and lead-lag effects.

In Ref. [8], a graph neural network is proposed for price forecasting, a Multi-modality Graph Neural Network (MAGNN). This modal neural network leverages a heterogeneous graph structure, incorporating various data sources as nodes and their relationships in a financial knowledge graph as edges. To ensure interpretability, they employ a two-phase attention mechanism for joint optimization. This method empowers analysts with interpretable insights for making well-informed investment decisions. The modal neural network learns from various multimodal inputs for financial time series prediction. This computational structure captures lead-lag movements, utilizing informative market data like prices, volume, movement, and raw news text. Basically, there is a construction of a heterogeneous graph by events, news, relations and market data. The modality inputs are feed to the attention mechanism. First phase graph attention mechanism is designed to propagate and aggregate information from source nodes to the target. The second phase source attention selectively aggregate the information from multi-modal sources for the target node representation. The learned features are fed into a feed-forward and classification network for target forecasting. Experimental results show that this complex model is able to extract predications and rich features from the different data sources.

A similar approach using a multi-graph convolutional network and recurrent neural network, highlighting improved predictions of stock movements using diverse data sources was studied [9]. The architecture of this computing model consists of an autoencoder, which encodes textual news into fixed vector length for each stock. A graph convolutional network, which takes the encoded daily news vectors and the graphs as input, acquiring a graph structure with multi-node embedding for the respective stock. Which then the node embedding are passed through an RNN to capture the temporal patterns in the news. Which then the LSTM layer conducts the node classification and stock movement. The results of this complex model demonstrate that this RNN is valid for stock index prediction problem, and its performance is powerful than the vanilla LSTM model based on the RNN. This research holds relevance, but selecting an appropriate modelling tool and delving into recurrent neural networks may require thoroughly investigation. Graph Neural Networks (GNNs) are gaining traction in quantitative finance due to their adeptness at modelling intricate graph structures as presented in [10, 11]. In Ref. [12], a ground-breaking GNN-based framework is introduced for predicting stock and bond defaults using historical stock price co-movement patterns. This study showcases the efficacy of GNNs in analysing and predicting from dynamic graphs. The computational setup involves a graph neural network for computing nodes and a long-short term memory network for recalling memory in datasets. Extensive experiments across three stock markets affirm the model's substantial enhancements in forecasting stock movements and return rates. Consequently, there are other elementary computing algorithms that also be introduced in the exercise of forecasting. Such as LSTMs, a specialized type of RNNs, excel in time series analysis by effectively retaining long-term memory in data. Unlike traditional RNNs, LSTMs mitigate the vanishing gradient problem.

Mahadik *et al.* [13] introduces two models, primarily focusing on LSTM and ARIMA, for visualizing future stock trends. The study employs a dataset encompassing open, close, high, and low values, subject to pre-processing steps like sorting, feature scaling, auto-correlation checks, and division into training and testing sets. Achieving over 90% accuracy, the LSTM model proves highly effective, particularly with large datasets. ARIMA, while accurate, demands more processing time and works best with complete attribute values. Thus, LSTM suits our research, contingent on optimal epochs, with confirmed accuracy based on MSE and MAE metrics [14]. Additionally, a novel approach using sentiment analysis for stock price forecasting is discussed, but its implementation involves complex natural language processing algorithms, exceeding the scope of this study. For detailed case studies, refer to Gupta and Chen [15], Mehta *et al*. [16], Pathak *et al*. [17].

The underlying concept of stock investment strategy revolves around the notion that if an individual can anticipate the future behaviour of a stock at the next time step, they can take appropriate actions, such as buying, selling, or holding. This summarizes the various computing models that have been recently published in literature. The graph based computing models are becoming more popular. As the structure of a graph enables market analysis by depicting relationships among stock prices. Therefore, analysing the graph representation offers insights and valid performance in the exercise of stock prediction. Moreover, researchers introduce different complex frameworks based on the structure of a graph that include one or more of the following: LSTM, Convolutional Neural Network (CNN), autoencoder or

RNN. A consist aspect of these graph based neural networks structures is that there is an attention mechanism where node classification is facilitated by a vanilla based computing model (LSTM, CNN, or RNN). Which then prediction is conducted on the finding or rather outputs from the attention mechanism. It is clear evidence that graph based neural network structures can be applied to the exercise of stock price prediction.

III. Theoretical Background

Artificial Intelligence (AI), which encompasses Machine Learning (ML), is a subset of computer science dedicated to analysing and interpreting data patterns and structures. This enables machines to learn, reason, and make decisions independently of human intervention. In essence, machine learning involves providing a computer algorithm with extensive data for analysis, enabling it to generate recommendations and decisions based solely on input data. If errors are identified, the algorithm can integrate this feedback to enhance future decision-making processes. In the field of Machine Learning there are three methods as to how to program the models, they are Supervised machine learning, Unsupervised machine learning, reinforcement learning and recently the emergence of transfer learning [18, 19]. Machine learning projects are conducted using a systematic analytic approach discussed below.

A. Data Science Methodology

Machine learning projects follow a specific methodology in either finding insight to data or building a model. Consequently, this methodology is called Cross Industry Standard Process for Data Mining (CRISP-DM) as depicted in Fig. 1. This is a commonly used data science methodology used in the field of data science. This methodology core focus is finding insight to the data. In addition, it is an iterative neutral process that can be used to structure the machine learning project.

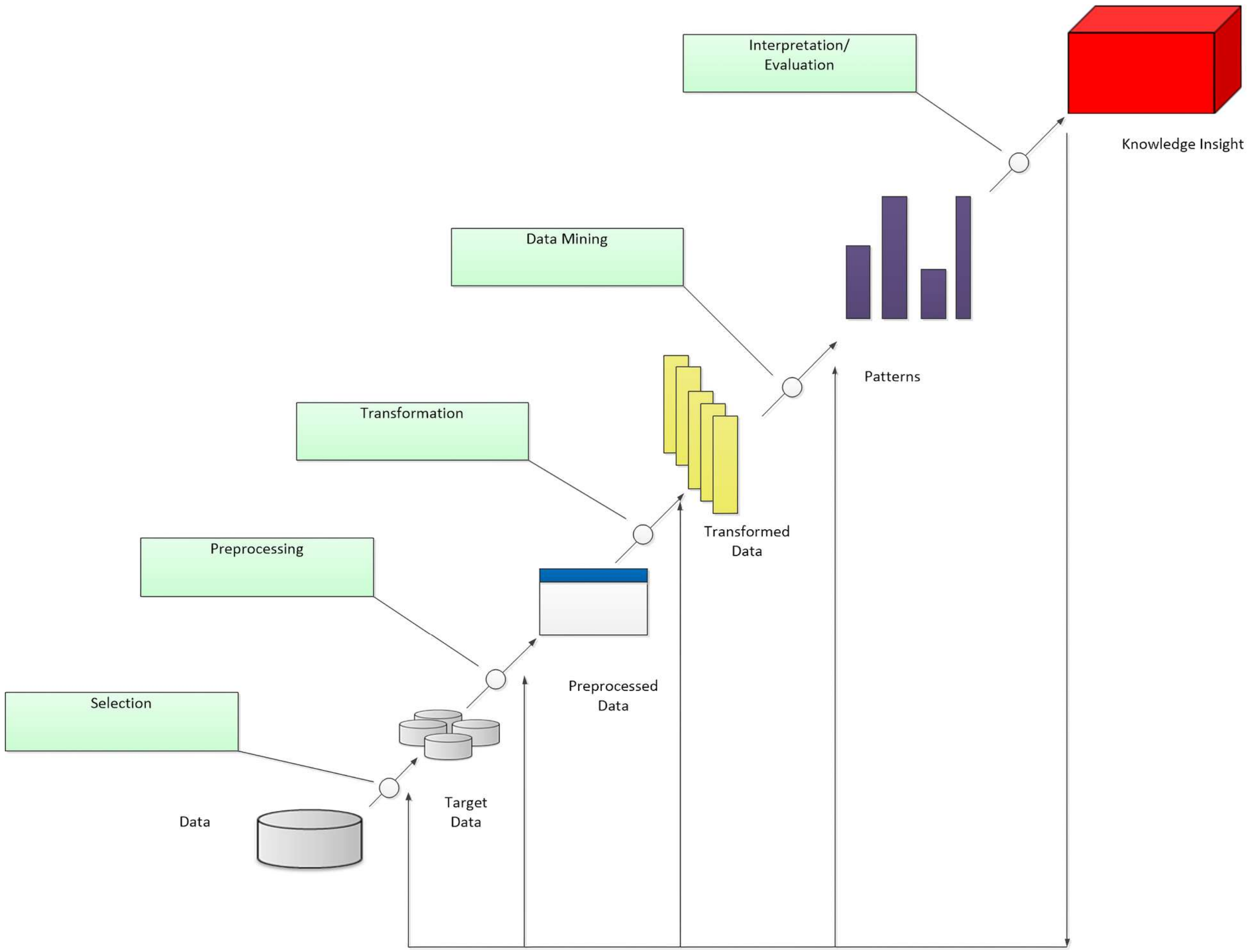

Fig. 1. Cross Industry Standard Process for Data Mining (CRISP-DM).

B. Neural Networks

Artificial neural networks are derived from their biological counterparts [20, 21]. The cell body of the neuron, where most of the neural computation takes place. Neural activity passes from one neuron to another in terms of electrical triggers via synaptic interaction. Consequently, we can simplify this as the weighted inputs and bias are summed and processed with a transfer function or called activation function as shown in Fig. 2. After being processed, the information is passed via outputs.

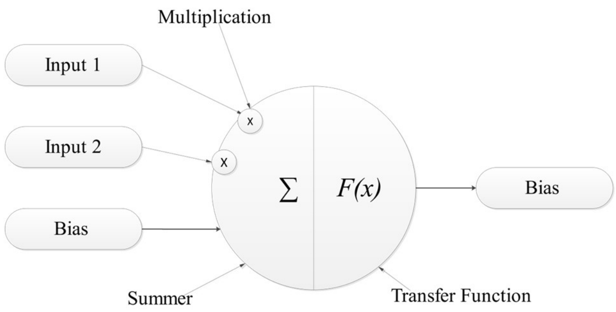

Fig. 2. Biological neuron model with a transfer function.

The choice of activation function in a neural network is crucial as it directly influences the network's ability to solve specific problems effectively [22]. Among the myriad of functions found in literature. Two functions stand out the most, which is the logistic Sigmoid and hyperbolic tangent transfer function, which is commonly utilised in neural networks due to their simplicity in finding derivatives, which is essential for training. Typically, these functions are employed within the hidden layers of the network.

Sigmoid activation function is represented as follows:

$$a(n) = \frac{1}{1+e^{-n}} \tag{1}$$

Hyperbolic activation function is defined as:

$$a(n) = \frac{e^{n}+e^{-n}}{e^{-n}+e^{-n}} \tag{2}$$

In both equations, n represents the slope parameter. Which permits the adjustment of the slope of the function. By varying the parameter n, we obtain functions of different slopes of n for the Sigmoid and Hyperbolic functions. The multi-layered network in Fig. 3 consists of input layer and consist of more hidden layers or rather hidden neurons, the stated activation function are applied. The input layer receives signals from source nodes, forming the input vector, which serves as the input signals for the neurons in the subsequent layers, such as the first hidden layer. The output signals of each layer then become the inputs for the following layer, continuing until the output layer is reached [23, 24].

This structure permits the network to process and transform data from its input through successive layers of computation nodes, thus a production of outputs that corresponds to the problem being solved. The process of training such a network, often using the back-propagation algorithm, involves adjusting the weights of connections between neurons to minimize errors and improve performance.

Thus, the choice and application of activation functions, such as the logistic Sigmoid and hyperbolic tangent

functions, play a crucial role in determining the effectiveness of neural networks in solving specific problems by facilitating the transformation and processing of input data throughout the network's layers [25]. Below we discuss some of the more advanced neural network architectures.

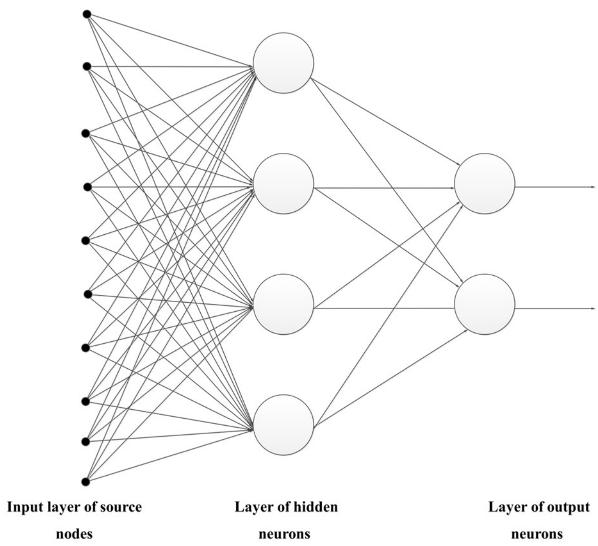

Fig. 3. Multi-layered network.

1) *Deep neural networks*

a) *Convolutional neural networks*

Various neural network architectures cater to different classification tasks. One such category, known as Convolutional Neural Networks (CNNs) which is shown in Fig. 4, is designed specifically for processing grid-like data, particularly images. CNNs have demonstrated notable effectiveness in computer vision applications. Unlike traditional neural networks that receive input as a vector of pixel values, CNNs take input in the form of a matrix, preserving the spatial relationships within images (comprising width, height, and depth). While CNNs share similarities with the artificial neural networks discussed earlier, they exhibit localized connectivity, where each neuron is linked to a subset of neurons in preceding layers. Additionally, CNNs employ weight sharing, a concept inspired by the functioning of the mammalian visual cortex [26–28].

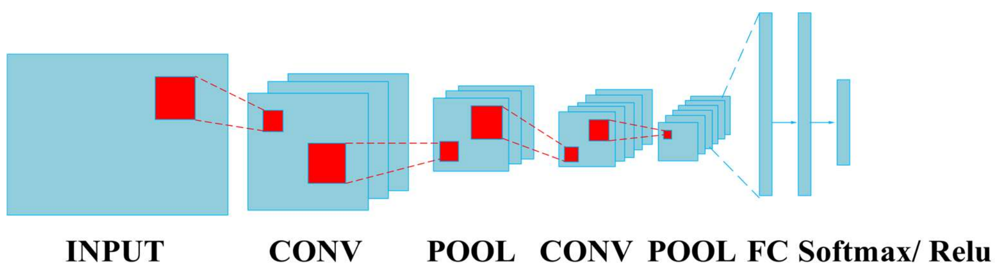

Fig. 4. General convolutional neural network.

A typical CNN comprises several key layers: data input, convolution, excitation, pooling, and fully connected layers, culminating in the output of results. The specific architecture of a CNN is determined by the design, arrangement, and quantity of these layers, tailored to achieve distinct classification objectives. the final output classification function is obtained by SoftMax function.

$$a(n) = \frac{e^{n_i}}{\sum_{j-1}^{K} e^{n_j}} \tag{3}$$

where, the SoftMax function variables are the input vector (n), the exponential function applied to each element of the input vector (e^n), and the sum of all the elements of the exponential function of the input vector ($\sum_{j-1}^{K} e^{n_j}$). The SoftMax function uses these variables to convert a vector of real numbers into a probability distribution.

b) Long short term memory networks

Long Short-Term Memory (LSTM) network is a variant of a recurrent neural network to overcome the problem of vanishing or exploding gradients in a traditional RNN. This property of the LSTM architecture is basically the ability to deal with lags of different durations of historical data [29]. The equation of an LSTM with a forget gate is given as follows:

$$\begin{aligned} f_t &= \sigma\,(x_t U^f + h_{t-1} W^f) \\ o_t &= \sigma\,(x_t U^o + h_{t-1} W^o) \\ C_t &= \sigma(f_t \times C_{t-1} + i_t \times \bar{C}_t) \\ h_t &= \tanh(C_t) \times o_t \end{aligned} \tag{4}$$

where, there are several components that make up this memory cell., the $x_t \in \mathbb{R}^a$ is the input vector the LSTM, $f_t \in \mathbb{R}^h$ is the forget gate's activation vector, $i_t \in \mathbb{R}^h$ is the gate's activation vector, $o_t \in \mathbb{R}^a$ is the output gate's activation vector , $h_t \in \mathbb{R}^a$ denotes the hidden vector also known as output vector of the LSTM, lastly, the $W \in \mathbb{R}^{h \times d}, U \in \mathbb{R}^{h \times h} and\ b \in \mathbb{R}^h$ are the weight matrices and bias vector parameters which need be learned. When the algorithm is being trained. Thus, this model network comprises cells that function as memory blocks and incorporates gate functions: input gate, forget gate, and output gate. The input gate determines which states should be updated, the forget gate determines which historical data to discard from the cell state, and the output gate decides which portion of the cell state should be outputted. Essentially, these gate functions instruct the LSTM to input, discard, or output information from a cell as needed [30–32]. Fig. 5 illustrates a LSTM cell.

Building the concepts of LSTM networks, represents a significant in advancement in neural network architecture particularly in handling data with long historical durations. However, for the purpose of our research. We introduce a neural network architecture that is able to computer complex relational structures. Called graph neural networks. These type of networks extend the capability to process and extract meaningful insights.

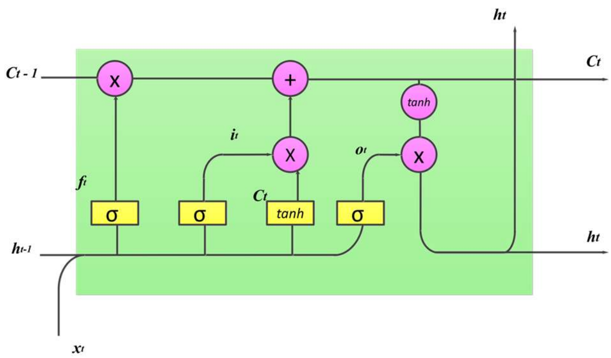

Fig. 5. Long term short memory cell.

2) Graph neural networks

Graph is a general method for describing and modelling complex systems, graphs are used to characterize interactions between objects of interest, to model simple and complex networks, or in general to represent real-world problems. Because they are based on a rigorous but straightforward formalism, graph based neural networks is becoming more common over time, transcending numerous traditional techniques which are commonly called in literature graph neural networks as shown in Fig. 6. Those are neural networks that constructed for processing graph-structured data sources [33–35].

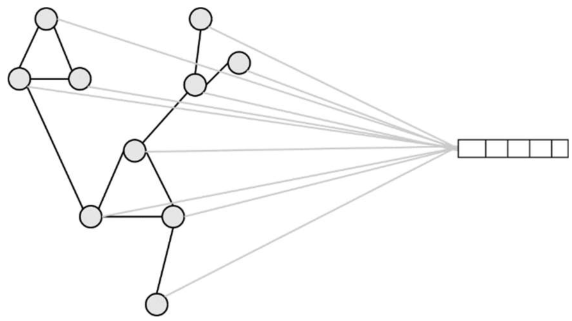

Fig. 6. Graph neural network.

A node is naturally defined by its features and related nodes in the graph. The target of GNN is to learn a state embedding $h_v \in \mathbb{R}^s$, which encodes the information of the neighbourhood, for each node. The state embedding h_v is used to produce an output $\boldsymbol{o}_v$, such as the distribution of the predicted node label.

In order to update the node state according to the input neighbourhood, there is a parametric function f, called *local* transition *function*, shared among all nodes. In order to produce the output of the node, there is a parametric function g, called local output function. Then, h_v and o_v are defined as follows [36]:

$$\begin{aligned} h_v &= f\,(x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]}) \\ o_v &= g\,(h_v, x_v) \end{aligned} \tag{1}$$

where x denotes the input feature and h denotes the hidden state. $co.v$ is the set of edges connected to node v and $ne[v]$ is set of neighbours of node v.

IV. Result and Discussion

A. Dataset Creation

All the data sets within the scope of this investigative study are time sequential. In its simplest form, a time series data set can be split up into input data, x, and output data, d. The output is what a ML model is supposed to output for the corresponding input.

For instance, the input data can be in the form $[x_0, \dots, x_{i-1}, x_i, x_{i+1}, \dots x_N]$where $i \in [0, N]$, x_i is an array of input values (of a constant length $n+1$) at time i, and $N+1$ is the number of measurements in the time sequence (Fig. 7). This matches to output data in the form $[d]$, where $[d_0, \dots, d_{i-1}, d_i, d_{i+1}, \dots d_N]$ is an array of output values (of a constant length $m+1$) at time i (as shown in Fig. 7). Sequential data can be processed in time windows of size smaller than or equal to the time sequence.

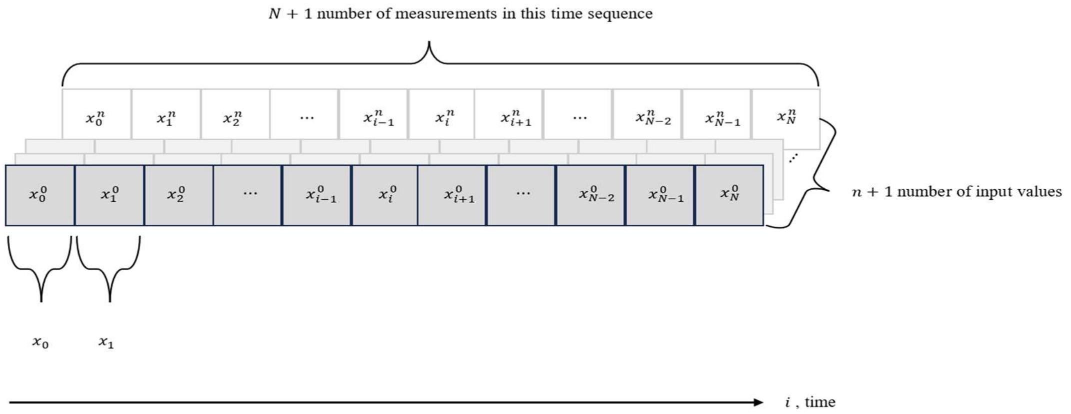

Fig. 7. Time series data of an array of inputs.

The data utilized in this study was sourced from Yahoo Finance, while the financial instrument that is going to be the case study data, namely Standard bank Group of South Africa. Which is a one of the top 40 large firms listed on the JSE by performance and market valuation. Yahoo Finance provides comprehensive financial data and insights across various markets and industries.

B. Dataset Experiment Setup

In the initial stages of the project, we used a local desktop setup for data pre-processing and graph creation. Our experiments were conducted on a computer environment featuring an 11th Gen Intel(R) Core(TM) i5-1145G7 @ 2.60GHz 2.61 GHz processor with 16.0 GB RAM. However, this setup lacked dedicated GPU hardware, limiting its capabilities. This system was primarily used for constructing the source code. To complement our work, we also leveraged Google Colaboratory (Colab), which comes equipped with popular deep learning libraries like TensorFlow, Pandas, Keras, and PyTorch. However, Colab has its limitations, including a 12-hour session time restriction and a maximum file upload size of 100 MB [37–40]. We train our model using an Adaptive Moment Estimation optimizer, applied to the outputs of the model. The evaluation metrics that are selected include Mean Absolute Error (MAE) and Mean Squared Error (MSE) Missing values are excluded both from training and testing.

The graph network architecture consists of Convolutional Neural Network (CNN) layers followed by Long Short-Term Memory (LSTM) layers and final dense layers [41–43]. The CNN layers in the model are designed to automatically learn hierarchical representations of the input sequences, capturing relevant patterns and features that are useful for subsequent sequence modelling by the LSTM layers. The LSTM layers play a crucial role in capturing and understanding the temporal relationships within the input sequences, enabling the model to make accurate predictions of future stock prices. The dense layers serve as the interface between the extracted features and the final output, enabling the network to learn complex mappings from input data to target predictions [44–46]. The graph network utilizes the Adaptive Moment Estimation algorithm for optimization during model training. The Adam optimizer plays a crucial role by effectively optimizing model parameters and accelerating convergence.

C. Experiment Results

Below is the preliminary work of the building the machine learning model. For the sake of the experimentation, historical data of Standard Bank (JSE-SBK) is utilized [47–49].

	Date	Open	High	Low	Close	Volume	OpenInv
0	2000-01-04	2555.0	2645.0	2500.0	2500.0	1154938	0
1	2000-01-05	2350.0	2525.0	2350.0	2510.0	2034137	0
2	2000-01-06	2500.0	2500.0	2450.0	2470.0	562136	0
3	2000-01-07	2460.0	2595.0	2460.0	2520.0	773065	0
4	2000-01-10	2600.0	2900.0	2600.0	2800.0	5396191	0

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5157 entries, 0 to 5156
Data columns (total 7 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Date     5157 non-null   object
 1   Open     5157 non-null   float64
 2   High     5157 non-null   float64
 3   Low      5157 non-null   float64
 4   Close    5157 non-null   float64
 5   Volume   5157 non-null   int64
 6   OpenInv  5157 non-null   int64
dtypes: float64(4), int64(2), object(1)
memory usage: 282.1+ KB
```

Fig. 8. Loaded data into environment.

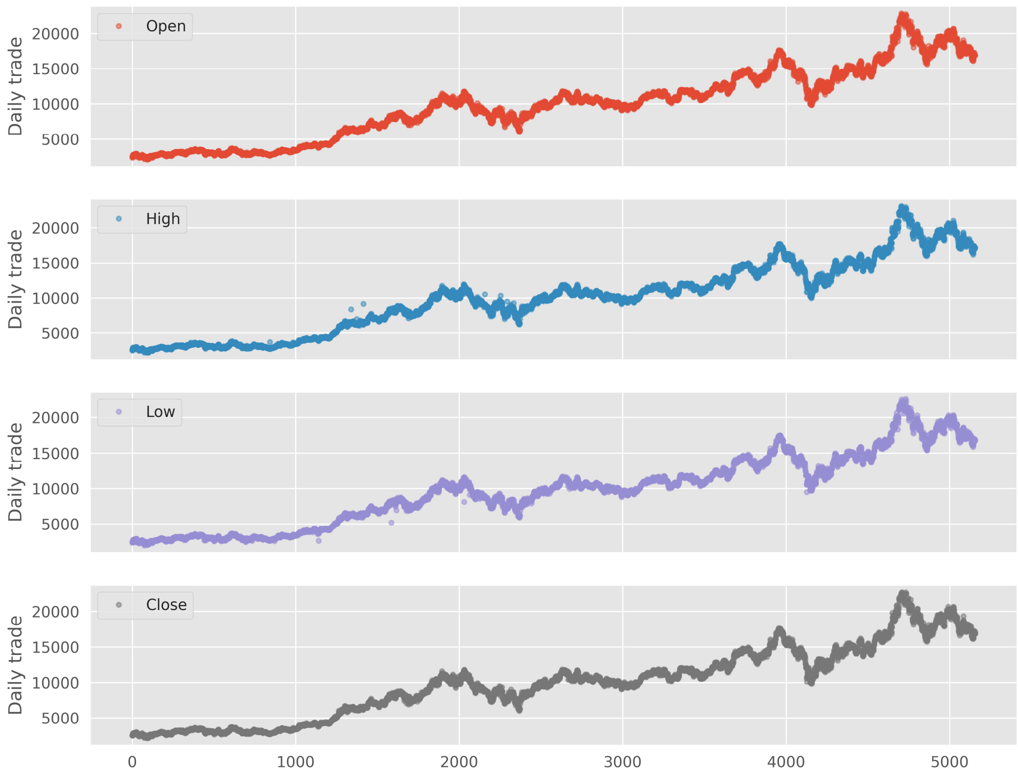

Fig. 9. Visualized the closing and opening prices.

The dataset was loaded, reading its contents, and also observation of the characteristics of datasets to see if there is a requirement of cleaning or addition of features. Using Panda functions. Fig. 8 is where we pre-process the data or in simply terms. We prepare the data set for the project. This is done by loading it into our environment. Consequently, we present the features of the dataset which are the variables for our neural network. And off course we also convert the dataset into a multidimensional array and visualise the dataset, as seen in Fig. 9.

We utilise the pre-built neural network structures found in Keras and Tensorflow [50]. For CNN, the layers are created. In every layer, TimeDistributed function is added to track the features for every temporal slice of data with respect to time. In between, MaxPooling layers are added. After that, it's passed to Bi-LSTM and CNN layers. As presented before, this is our graph based forecast model. Which is based on the following framework presented in [51]. Which from the experiments we acquire the following.

In Fig. 10, for node representations, this indicates how much data has been modelled to each node. Thus, for a certain amount of times series data that will be passed by the nodes of the machine learning model.

In Fig. 11, the Average loss versus MSE graph output from Tensorflow board is presented. This indicates the number of MSE errors that occurred numerically for each iterations during the training of the dataset. As with time series forecasting, the errors must be reduced to find the best predictions of the state or the dependent variables in this the closing and opening price of the stock.

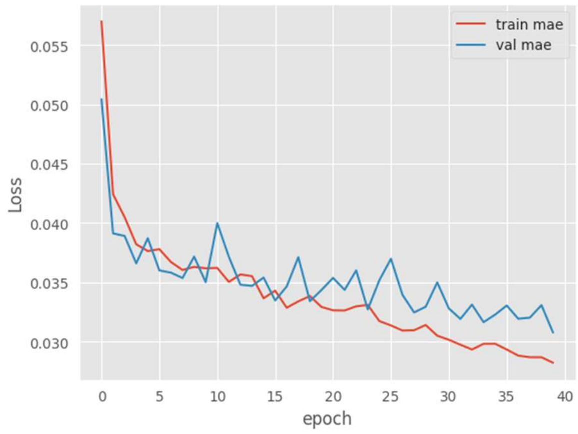

Fig. 10. Node representation.

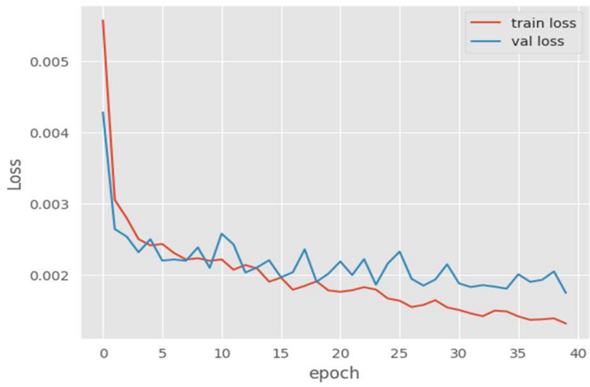

Fig. 11. Average loss and MSE per epoch.

The graph of Fig. 12 is an indication of the training loss when the neural network was going back and fourth in the data, the is an exponential decay as the number of epochs increase in number as seen by the red curve.

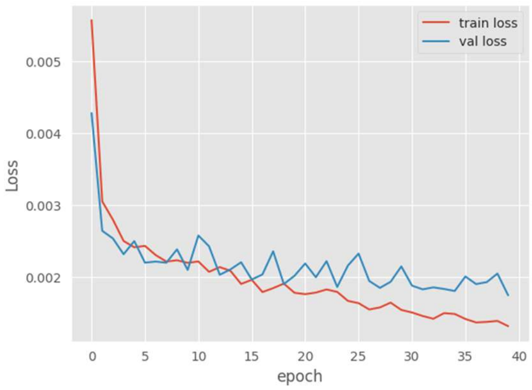

Fig. 12. Training and validation loss versus epochs.

Fig. 13 is the predication of the volatility of the stock based on the predicated future trend. An indication of where the stock is non-volatile regardless the macro-economic factors. However, we must recall that industry methods require back testing strategies to quantify the results for real life applications. In addition, Fig. 14 is the prediction of the graph-based LSTM-CNN model in, is that of the near the actual price, and Which indicates the satisfactory results of our simulation. However, the LSTM cannot entirely, produce 99% accurate results as there are other computational factors to account for.

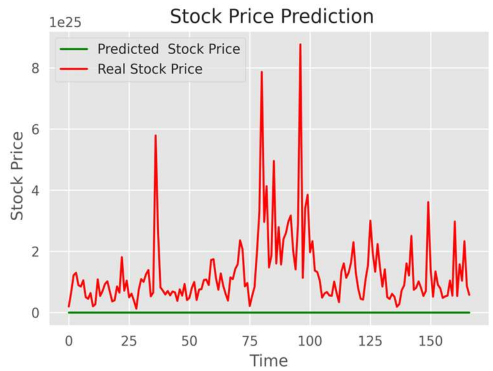

Fig. 13. Predication of the volatility

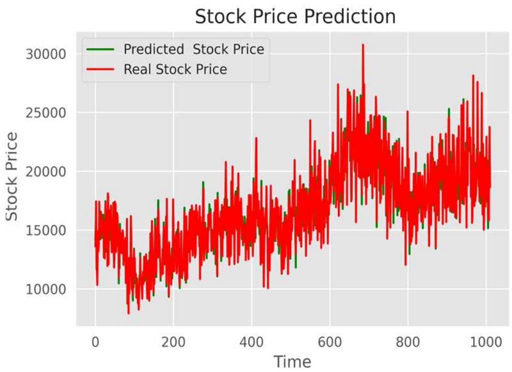

Fig. 14. Exponential averaging prediction of movement on yearly chart.

The graph based CNN + LSTM model was complied, we can see different behaviour of the stock price over time, during the 2012s the stock rises and thereafter drastically falls during the Covid pandemic in South Africa. To ensure the validity of the model and its ability to generalize, we validated it using a separate test set. This set was distinct from the data used during training, allowing for an unbiased assessment of the model's performance on new, unseen data. However, the division of data into training and test sets was approximately 85% for training and 15% for testing. This split deviates significantly from the recommended 70%/30% ratio, which could potentially lead to overfitting of the model to the dataset. Thus, according to statistical metrics. The model was predicted with a variance score of 0.913, which indicates an extremely high level of accuracy in predicting the stock price future behaviour. This implies that our model captures over 91% of the variability in the data, which is a strong indication of its reliability. Moreover, the R^2 score, also at 0.913, reinforces the model's efficiency. This score, often referred to as the coefficient of determination, signifies the proportion of the response variable's variance that is predictable from the independent variables. A value of 0.913 indicates an impressively strong relationship between the predicted and actual values. Furthermore, the maximum error of 0.154 suggests that our model's predictions are very close to the actual values. This low margin of error indicates that our LSTM-CNN model is consistently accurate in its forecasting, which is a critical characteristic for any predictive model. Overall, these results demonstrate that our graph-based LSTM-CNN forecast model is not only powerful but also highly reliable. It has the potential to provide valuable insights for a range of applications, from finance to climate modelling, where accurate forecasting is paramount.

V. Conclusion

Anticipating the future has long been an aspiration for both economist and individuals, given the potential advantages it holds. Those engaged in stock market analysis will find value in accurately predicting price movements. With the advent of artificial intelligence, researchers can now access forecasts of unprecedented precision. This capability is poised to improve further as technology and algorithms continue to advance. The findings of this study underscore the efficacy of this novel approach.

In this paper, we present a novel model for spatial-temporal machine learning modelling based on [43, 50]. "Spatial" refers to the relationships or dependencies between different data points or entities, while "temporal" refers to the changes in data over time. In the context of this paper, the model captures the relationships between different stocks (spatial) and how their prices change over time (temporal). The model uses a combination of Long Short-Term Memory (LSTM) units and Convolutional Neural Networks (CNNs) to capture these dependencies. Our model captures spatial-temporal dependencies efficiently and effectively by combining LSTMs with CNNs. Even though this model showed comparative results similar to those presented in [52]. The combination of a neural network framework incorporating convolution and long-short-term memory units has been confirmed to outperform traditional statistical methods, as well as conventional CNN and LSTM approaches, in predictive tasks as mentioned in [52]. However, feature engineering should be considered in designing better deep learning models as it improves the quality of the model. In this work, complex modelling of deep learning model was learnt, and more about graph machine learning.

Given that the primary objective of this paper is to forecast stock market fluctuations and demonstrate its effective application in forecasting, the subsequent phase will involve deploying the proposed algorithm to signal specific market movements. Temporal Graph Neural Networks (TGNNs) are a type of graph neural network that incorporate temporal information into the graph structure. In our future research, we plan to explore how TGNNs can be used to capture temporal dependencies in stock price data more effectively. For instance, we might investigate different ways of incorporating time into the graph structure, or different types of temporal attention mechanisms. In future research, consideration will be given to developing a pure graph model utilizing temporal graph neural networks, or spatial-temporal graph networks. Additionally, existing models like FinGAT or GraphWave network may also be explored [53]. Furthermore, there will be an endeavour to construct an expert system for investment purposes. Where, a back testing strategy system will also be developed to improve its practicality in quantitative hedge funds.

Conflict of Interest

The authors declare no conflict of interest

Author Contributions

The main contribution of this research is on the ability to apply machine learning for forecasting stock prices in the JSE. In the realm of financial markets, a multitude of traders leverage technical analysis to anticipate stock price fluctuations. Understanding the impact of the market volatility on a JSE Top 40 listed instrument, introduces a novel approach in this field.

Previously, forecasting JSE listed instruments was not feasible. However, as a tool for measuring market fear, its predictability could significantly contribute to the field. In this work Mvuleni Kekana contributed to the conduction of the investigative study underpinning computational intelligence and wrote the first draft, Mbuyu Sumbwanyambe, conceptualised the idea and Tlotlollo Hlalele reviewed and wrote the final draft of the paper. All authors had approved the final version

Acknowledgment

I am deeply grateful to my supervisor, Prof. Mbuyu Sumbwanyambe, for his support, guidance, and exceptional supervision. He was always available to read and give feedback on my progress, and he spent considerable time on MS Teams calls during the investigative process. His attention to detail has helped me to develop a critical eye and improve my writing and work. His assistance and advice made it possible for me to complete and submit this dissertation. Thanks to my colleagues from the honours in electrical and electronic engineering study group. And I also give thanks to Dr. Hlalele for his guidance on compiling and preparing the journal paper for publication submission.

I would like to take this opportunity to express my sincere thanks and heart-felt gratitude to Dr. Mashabela, Dr. Kubheka for their guidelines on conducting postgraduate research activities. I am also grateful to the researchers who have published their work, as their work has been a valuable source of information and education. Finally, I am grateful to the Department of Electrical Engineering at the University of South Africa for allowing me to pursue and complete my honour's degree. And the Postgraduate studies office for granting me with the necessary funds to further my studies.

References

[1] R. A. Schwartz, J. A. Bryne, and L. Wheatley, *The Economic Function of a Stock*, London, UK: Springer International Publishing, 2015, p. 6.

[2] M. Jaeger and S. Krugal. (2020). Five towards robust portfolios: Hierarchical risk parity: persistent diversification uncovered by graph theory and machine learning. Munich RE. [Online]. Available: https://www.munichre.com/content/dam/munichre/contentlounge/website-pieces/documents/FIVE_VROBUST_Index__Research-EN.pdf

[3] J. C. Hull, *Machine Learning in Business: Introduction to the World of Data Science*, Toronto, CND: John. C. Hull, 2019.

[4] X. Xie, *A Systematic Approach to Active Stock Market Investing*, Clementi Loop, SPR: John Wiley and Sons (Asia) Pte. Ltd, 2011, pp. 11–33.

[5] S. Hupfer. (2020). Capitalizing on the promise of artificial intelligence. *Perspectives on AI Adoption from Around the World, Deloitte Insights*. [Online]. Available: https://www2.deloitte.com/us/en/insights/focus/cognitive-technologies/global-perspectivesai-adoption. html

[6] Y. Fang, "Stock price forecasting based on improved support vector regression," in *Proc. 7th Int. Conf. on Information Science and Control Engineering (ICISCE)*, 2020, pp. 1351–1354.

[7] G. Bathla, "Stock price using LSTM and SVR," in *Proc. 6th Int. Conf. on Parallel, Distributed and Grid Computing (PDGC)*, 2020, pp. 211–214.

[8] D. Cheng, F. Yang, S. Xiang, and J. Liu, "Financial time series forecasting with multi-modality graph neural network," *Journal of Pattern Recognition*, vol. 121, 108218, 2022.

[9] Q. Chen and C. Y. Robert, "Graph-based learning for stock movement prediction with textual and relational data," arXiv preprint, arXiv: 2107.10941, 2021.

[10] M. Wang, A. L. M. Vilela, L. Tian, H. Xu, and R. Du, "A new time series prediction method based on complex network theory," in *Proc. 2017 IEEE International Conference on Big Data*, Boston, MA, USA, 2017, pp. 4170–4175.

[11] F. Xia, K. Sun, S. Yu, A. Azizm L. Wan, S. Pan, and H. Liu, "Graph learning: A survey," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 2, pp. 109–127, April 2021.

[12] H. Tian, X. Zheng, K. Zhao, M. W. Liu, and D. D. Zeng, "Inductive representation learning on dynamic stock co-movement graphs for stock predictions," *INFORMS Journal on Computing*, vol. 34, no. 4, pp. 1940–1957, 2022.

[13] A. Mahadik, D. Vaghela, and A. M. Haisgawali, "Price prediction using LSTM and ARIMA," in *Proc. Int. Conf. on Electronics and Sustainable Communication Systems (ICESC)*, 2021, pp. 1594–1601.

[14] D. Liu, A. Chen, and J. Wu, "Research on stock price prediction method based on deep learning," in *Proc. 2nd Int. Conf. on*

Information Technology and Computer Applications (ITCA), 2020, pp. 69–72.
[15] R. Gupta and M. Chen, "Sentiment analysis for stock price prediction," in *Proc. IEEE Conf. on Multimedia Information Processing and Retrieval (MIPR)*, 2020, pp. 213–218.
[16] Y. Mehta, A. Malhar, and R. Shankarmani, "Stock price prediction using machine learning and sentiment analysis," in *Proc. 2nd Int. Conf. for Emerging Technology (INCET)*, 2021, pp. 1–4.
[17] A. Pathak and S. Pathak, 'Study of machine learning algorithms for stock market prediction," *Int. Journal of Engineering Research and Technology (IJERT)*, vol. 9, no. 6, pp. 295–300, 2020.
[18] M. Obthong, N. Tantisantiwong, W. Jeamwattanachi, and G. Wills, "A survey on machine learning for stock price prediction: Algorithms and techniques," in *Proc. 2nd International Conference on Finance, Economics, Management and IT Business*, Prague, Czech Republic, 2020, pp. 63–71.
[19] N. Rouf, M. B. Malik, T. Arif, S. Sharma, S. Singh, S. Aich, and H. Kim, "Stock market prediction using machine learning techniques: A decade survey on methodologies, recent developments, and future directions," *Electronics*, vol. 10, no. 21, 2021.
[20] S. Lin, J. Alves, F. Bugiotti, and F. Magoules, "A comparison study of graph neural network and support vector machine," in *Proc. 21st International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, Chizhou, China, 2022, pp. 46–49.
[21] S. Haykin, *Neural Networks and Learning Machines,* 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2012, pp. 14–30.
[22] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. de Jesus, *Neural Network Design,* 2nd ed. Stillwater, USA: M. T. Hagan, H. B. Demuth, 2014, pp. 30–60.
[23] D. Graupe, *Principles of Artificial Neural Networks*, 3rd ed. Hackensack, USA: World Scientific Publishing Co. Pte. Ltd., 2013, pp. 28–34.
[24] L. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*, New York, USA: Prentice-Hall, 1994, pp. 294–296.
[25] G. James, D. Witten, T. Hastie, and R. Tibshirani, *Introduction to Statistical Learning*, New York, USA: Springer+Business Media LLC, 2021, pp. 403–430.
[26] G. Bonaccorso, *Machine Learning Algorithm*, Birmingham, UK: Packt Publishing, 2020.
[27] L. D. Marchi and L. Mitchell, *Hands on Neural Networks*, Birmingham, UK: Packt Publishing, 2019, pp. 112–130.
[28] C. Stamile, A. Marzullo, and E. Deusebio, *Graph Machine Learning: Take Graph Data to The Next Level by Applying Machine Learning Techniques and Algorithms*, Birmingham, UK, Packt Publishing, 2021.
[29] A. Negro, *Graph Powered Machine Learning*, Shelter Island, USA: Manning Publications Co., 2021.
[30] M. Labonne, *Hands-On Graph Neural Networks Using Python*, Birmingham, UK, Packt Publishing, 2023.
[31] S. Raschaka and V. Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and Tensorflow 2*, Birmingham, UK: Packt Publishing, 2019.
[32] A. Geron, *Hands on Machine Learning with Scikit-Learn, Keras, TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, Sebsatopol, USA: O'Reilly Media, 2021.
[33] W. Di, A. Bhardwaj, and J. Wei, *Deep Learning Essentials*, Birmingham, UK: Packt Publishing, 2020
[34] A. Grigorev, *Machine Learning Bookcamp: Build a Portfolio of Real Life Projects*, Shelter Island, USA: Manning Publications, 2021.
[35] D. Grattarola and C. Alippi, "Graph neural networks in TensorFlow and Keras with Spektral [application notes]," *IEEE Computational Intelligence Magazine*, vol. 16, no. 1, pp. 99–106, 2021.
[36] P. Kim, *MATLAB Deep Learning*, New York, USA: Springer Science + Business Media, 2017
[37] The Johannesburg Stock Exchange. (1941). Committee for the Information of the Public. [Online]. Available: http://financialmarketsjournal.co.za/oldsite/14thedition/printedarticles/jse.htm
[38] Standard bank. Basic Investment course. [Online]. Available: https://securities.standardbank.co.za/ost/
[39] J. H. Cochrane and T. J. Moskowitz, *The Fama Portfolio: Selected Papers of Eugene F. Fama*, Chicago, USA: The University of Chicago Press, 2017, pp. 62–100.
[40] Z. Wu, S. Oan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modelling," arXiv preprint, arXiv:1906.00121, 2019.
[41] Y. Hsu, Y. Tsai, and C. Li, "FinGAT: Financial graph attention networks for recommending top-k profitable stocks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 469–481, 2023.
[42] R. Kim, C. So, M. Jeong, S. Lee, J. Kim, and J. Kang, "HATS: A hierarchical graph attention network for stock movement prediction," arXiv preprint, arXiv:1908.07999, 2019.
[43] R. Sawhney, S. Agarwal, A. Wadhwa, T. Derr, and R. R. Shah, "Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach," in *Proc. the AAAI Conf. on Artificial Intelligence*, 2021, pp. 497–504.
[44] X. Li, J. Wang, J. Tan, S. Ji, and H. Jia, "A graph neural network-based stock forecasting method utilizing multi-source heterogeneous data fusion," *Journal of Mulitmedia Tools and Application*, vol. 81, no. 30, pp. 43753–43775, 2022.
[45] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T. Chua, "Temporal relational ranking for stock prediction", *ACM Transactions on Information Systems*, vol. 37, no. 2, pp. 1–30, 2019.
[46] Z. Zheng, J. Shao, J. Zhu, and H. T. Shen, "Relational temporal graph convolutional networks for ranking-based stock prediction," in *Proc. 2023 IEEE 39th International Conference on Data Engineering (ICDE)*, Anaheim, CA, USA, 2023, pp. 123–136.
[47] K. Huang, X. Li, F. Liu, X. Yang, and W. Yu, "ML-GAT: A multilevel graph attention model for stock prediction," *IEEE Access*, vol. 10, no. 10, pp. 86408–86422, 2022.
[48] K. Pillay. (2021). Investigating the application of graph neural networks to stock market prediction. [Online]. Available: https://projects.cs.uct.ac.za/honsproj/cgi-bin/view/2021/dhansay_pillay.zip/assets/res/CSC4002W_COGSPMS_Paper_PLLKIA010.pdf
[49] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, J. Bai, and Q. Zhang, "Spectral temporal graph neural network for multivariate time-series forecasting," in *Proc. Int. Conf. on Neural Information Processing Systems (NeurIPS)*, 2020, vol. 33, pp. 17766–17778.
[50] J. M.-T. Wu, Z. Li, N. Herencsar, B. Vo, and J. C.-W. Lin, "A graph based CNN LSTM stock price prediction algorithm with leading indicators," *Multimedia Systems*, vol. 29, no. 3, pp. 1751–1770, 2021.
[51] A. C. Faul, *A Concise Introduction to Machine Learning*, Boca Raton, USA: CRC Press, 2020.
[52] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Boston, USA: MIT Press, 2016.
[53] M. Peixeiro, *Time Series Forecasting in Python*, Shelter Island, USA: Manning Publications Co, 2022.