LSTM Prediction Capability on The South African JSE Top 40 of Historical and Live Data

Mohsen Elhag

February 2024

Supervisor: Dr Godfrey Ndlovu

Abstract

This study evaluates the efficacy of Long Short-Term Memory (LSTM) models in stock price forecasting using data from the South African FTSE/JSE Top 40 index, a domain yet to be extensively explored, particularly in real-time data analysis. Addressing the gap in existing research, this study assesses LSTM model predictive capability in the South African stock market on historical data and its adaptability to the dynamic, real-time stock market environment over the period from January 2001 to January 2024.

Various LSTM models were trained with different configurations, and the results show that a single-layer LSTM model performs better than its multilayer counterpart in processing historical data, in terms of the mean absolute error (MAE), the root mean square error (RMSE), Mean Absolute Percentage Error (MAPE) and the R-squared. However, when applied to real-time data, the accuracy of the single-layer model diminishes, underscoring the challenges posed by the dynamic and unpredictable nature of live stock market conditions.

The findings contribute to the field of financial forecasting by demonstrating the strengths and limitations of the LSTM model in the context of the South African stock market. While showcasing significant potential in historical data analysis, performing on par with previous studies, the study underscores the need for further development of the model for real-time forecasting. Future research directions include extending the testing period, integrating diverse data sets, and exploring a combination of LSTM with other forecasting methodologies.

Declaration

I, Mohsen Elhag, hereby declare that this thesis is a result of my own work and has been performed and written by myself and is being submitted as a part of the degree of Master of Philosophy in Financial Technology at the University of Cape Town. It has not been previously submitted for any degree or examination.

Signed: Mohsen Elhag Date: 29 February 2024

Contents

1. Introduction and background

The ability to predict live data and future stock prices stands as one of the most valuable applications of machine learning in the realm of stock market analysis. Existing research evidence demonstrates the efficacy of machine learning in quantitative finance, with a focus on predicting stock market prices, using algorithms that uncover patterns directly from data, like Chung and Shin, (2018), Nabipour et al. (2020), and Liu et.al. (2022). These algorithms not only show promising results in forecasting market trends, but they also play a role in broader financial operations, including asset management, portfolio construction, and investment strategies. Among these algorithms, the Long Short-Term Memory (LSTM) algorithm has proven to be effective in predicting historical stock market prices. However, its performance in predicting live market data remains relatively unexplored.

Although the prediction of stock market prices is a strenuous task, it is important for various decision makers, ranging from individual investors to large organizations that engage in the stock market, to yield profits, hedge funds and/or avoid plausible losses. The two main approaches used to achieve this endeavor, are fundamental analysis, which relies on information about the stock market and the market it operates in, and technical analysis, which relies on the previous stock price information (Nti et al. 2020).

Lately, Machine learning has emerged and garnered attention in the realm of financial forecasting and time series prediction. Machine learning is an area of Artificial Intelligence (AI) which uses a set of algorithms that allow computers to learn from past data. Machine learning algorithms can incorporate both technical and fundamental analysis. It has three main sub-areas namely, supervised, unsupervised and reinforcement learning. Ayodele (2010) defines unsupervised learning as clustering inputs together to form a pattern. Reinforcement learning is the closest form to how humans learn, where every action has an influence in the environment and the environment provides feedback that guides the learning algorithm (Charpentier and Remlinger, 2021). Finally, supervised learning is where the algorithm is mapped to approximate known inputs (Ayodele, 2010).

LSTM is a supervised learning algorithm, it is a subclass of recurrent neural networks and is suitable for understanding complex patterns in sequences of data and for predicting time series. The core part of an LSTM model is the memory block, which is made up of memory cells and gates. The parts of an LSTM model work collectively to help the network capture long-term dependencies in data, achieved by controlling the flow of information through the gates and updating the memory cell's state. LSTMs can selectively retain relevant information while discarding less essential information. Its ability to manage information flow over long sequences allows LSTM models to excel in tasks like time series prediction (Hua et al. 2019).

Although, the efficient market hypothesis (Fama, 1970), suggests that market prices follow a random walk, suggesting that at any given point in time, the market reflects all information available, and it is not possible to consistently outperform the market. Previous studies in the machine learning space, suggest that it is possible to predict market performance. For instance, the study of Kimoto et.al. (1990), Grudnitski and Osburn, (1993), Patel et.al. (2015) and Ghosh, Neufeld and Sahoo (2022). Specifically, the application of Long Short-Term Memory (LSTM) machine learning algorithm has gained significant attention in predicting the stock market, due to its remarkable performance in various market scenarios. Recently, the LSTM model has demonstrated its superiority over other algorithms when it comes to analyzing the stock market historical data and making accurate predictions. For example, a study by Banik et al. (2019), in the Indian stock market where they assess the effectiveness of LSTM in comparison to few other models. They find that the LSTM model had the best performance in terms of the Root Mean Square Error (RSME) and the Mean Absolute Error (MAE), compared to other models examined.

It is important to take into consideration the explanatory variables used in building the model. In a more recent study Peng eat al. (2021), the LSTM model was further enhanced by incorporating a substantial array of 124 technical indicators into the training dataset. The aim of the study was to establish the most relevant indicators used for stock forecasting. While this inclusion had the potential to boost the model's accuracy, it is imperative to take into consideration the "factor zoo." A concept introduced by Cochrane (2011) which highlights the challenge of determining which subset of indicators genuinely contributes to improving predictions accuracy due to the overwhelming number of indicators available currently. This highlights the complexity involved in designing and fine-tuning models that effectively utilize an abundance of data.

However, a study conducted by Gosh et al. (2019) examined the predictive capabilities of both Random Fores (RF) and LSTM models, focusing on the S&P 500 index. Their findings suggest that LSTM outperformed RF, in terms of the daily returns. Furthermore, researchers have explored the influence of varying feature sets in building the model and found that utilizing three specific features yielded the most optimal results when paired with LSTM model.

This paper seeks to assess the viability of the Long Short-Term Memory (LSTM) model in predicting historical and real-time daily closing prices of the FTSE/JSE Top 40 Index. Despite the model being assessed in different markets the model has not been applied in the South African stock market while taking into consideration performance in a real-world scenario. Although, the assumption may be that the model may yield the same results using live data as that of historical data, it is important to consider the high volatility of live data, which may cause the results to vary.

In this study, the LSTM model is incorporated with the core of technical analysis by focusing on predicting the next day's closing prices solely based on historical daily closing prices, utilizing the fundamental premise under technical analysis that price is a primary indicator of future market behavior. The study focuses on the South African stock market, particularly the JSE Top 40 Index. To determine whether machine learning can predict stock prices using historical and live data, and whether the findings can be applied to live data and how accurate such predictions will be. Most research tends to concentrate on developed markets like the United States and Europe, leaving a significant gap in understanding the dynamics of developing and emerging markets. Analyzing these less-studied markets is essential, as they present unique investment opportunities and economic insight. Developed markets offer stability and predictable growth, emerging markets on the other hand are known for their high growth potential and rapid economic changes, which in turn may lead to higher risk exposure, more expected growth, as well as more diversification opportunity to investors. Further, South Africa is the home to the largest stock exchange in Africa, the Johannesburg Stock Exchange (JSE). The JSE is perfect for bridging this gap as it is fairly sizable, liquid and accessible. Conducting studies in such a market can offer a gateway to the African market as a whole and bridging the understanding of the stock market predictability, specifically for emerging markets and more specifically for Africa.

The LSTM model leverages this concept of technical analysis but advances it further through its ability to process and learn from large volume of sequential price data. This approach is a marked improvement over traditional technical analysis models, such as Autoregressive Integrated Moving Average (ARIMA), which may not be as adept at handling the complexity and volume of the financial market data. LSTM's sophisticated neural network architecture is uniquely capable of discerning intricate patterns and trends within historical price data, thereby offering a more nuanced and potentially more accurate prediction of stock market trends. In doing so, the LSTM model represents an evolved form of technical analysis, harnessing the predictive power of machine learning to refine and elevate the accuracy of stock market forecasting.

The subsequent sections of this paper will be organized as follows: starting with Literature review in section 2, followed by the methodology in section 3, then the model building and results and discussion in section 4, and finally a brief conclusion and suggestion for future research in section 5.

2. Literature Review

The Efficient Market Hypothesis (EMH), introduced by Samuelson (1965) and Fama (1965), is a fundamental concept in understanding market dynamics. According to EMH, markets efficiently reflect all available information at any given moment, indicating that consistently predicting stock market prices is merely an impossible endeavor. Further, Jensen (1987) categorized EMH into three levels: strong, semi-strong, and weak in accordance with the availability of information. In the strong form, it is posited that prices incorporate all available information at a particular moment. The semi-strong form suggests that prices mirror all publicly available information at a specific time. On the other hand, the weak form assumes that prices solely reflect historical prices of the market at a given time (Jensen, 1987). This study focuses on predicting the FTSE/JSE Top 40 index using the Long Short-Term Memory (LSTM) model, under the assumption that the market is weak form inefficient.

In stock market forecasting, the traditional methodologies are broadly classified into fundamental analysis and technical analysis. Fundamental analysis heavily relies on analyzing data obtained from current and previous financial statements to identify firms' value in a specific period. Including factors such as the simultaneous shift in inventory levels, gross margins, sales and capital expenses, audit outcomes and workforce production yield in sales (Abarbanell and Bushee, 1998). On the other hand, technical analysis relies on the study of market trends and investor behaviors, primarily achieved through various market indicators (Park and Irwin, 2007). Technical analysis is based on the belief that historical price action can offer clues about future market movements. As such, a number of indicators have been used in this field, such as closing prices, volume and moving averages; among others, which fundamentally depend on price data. This reliance underscores the significance of price as the core element in technical analysis.

Before the introduction of deep learning techniques, statistical models like exponential smoothing, autoregressive integrated moving average (ARIMA), and state space models, were used for time series prediction. However, these models are designed to identify and exploit patterns in historical data, and thus use these to make future predictions. Exponential smoothing forecasts are generated by applying weighted averages to past data points, which helps to create smooth predictions. ARIMA models, on the other hand, incorporate autoregressive and moving average elements to manage dependencies and random variations within the data. State Space models provide a higher level of adaptability, allowing them to capture multiple sources of variation, such as trends, cycles and seasonal effects (Gardner, 2006). However, despite their relative simplicity, interpretability, and robustness, traditional methods may struggle in scenarios where data is highly noisy, nonlinear, or exhibits complex dynamics (Siami-Namini, Tavakoli, and Namin, 2018). This limitation can lead to less accurate forecasts in such challenging conditions. Despite ARIMA being one of the most used traditional methods for time series forecasting as mentioned by Fan et al. (2021), numerous studies have compared ARIMA with deep learning techniques, in particular LSTM. For instance, the study of Siami-Namini, Tavakoli, and Namin (2018) and Pirani (2022), both find that the LSTM outperformed ARIMA in terms of root mean square error by a sizable margin. Further Bagul et al. (2022), established that the LSTM model performed better than ARIMA in terms of loss functions.

2.1. Overview of LSTM and its Application in Financial Markets

The LSTM model was first introduced by Hochreiter and Schmidhuber (1997) as a solution to the information loss observed in regular Recurrent Neural Network (RNN) models. LSTM models excel in processing time-series data, making them highly suitable for complex financial market predictions. This excellence and suitability have been demonstrated in numerous studies. For instance, Nabipour et al. (2020) compared 11 different machine learning models amongst which, there are two deep learning models, namely, LSTM and RNN. In their research, LSTM and RNN models demonstrated superior performance in predicting stock prices in the Tehran Stock Exchange in terms of accuracy. Similarly, Fischer and Krauss (2018) showcased LSTM's ability compared to memory free classification methods, over a period extending from December 1989 until September 2015. In their study the LSTM model outperformed the memory free classification methods, and the authors highlighted LSTM ability to extract meaningful information from noisy time series data in the S&P 500. Furthermore, Hiransha et al. (2018) compared four deep learning models, namely: RNN, LSTM, convolutional neural network (CNN) and multilayer perceptron (MLP) on the National Stock Market (NSE) of India and the New York Stock Exchange (NYSE). The models were built using two stocks from the NSE and tested on both, the NSE and the NYSE, and the LSTM model performed well on average, in terms of mean absolute percentage error (MAPE). Furthermore, the model was tested in the Korea Stock Exchange in a period extending from January 2000 until December 2015, the Korean Stock Price Index was used, and the LSTM model performed well in terms of MSE MAE and MAPE (Chung and Shin, 2018), Indonesia Stock

Exchange during COVID-19, the LSTM model performed well in terms of RMSE and accuracy (Budiharto, 2021), and the S&P 500 between January 1990 to October 2015, where the LSTM model outperformed the memory-free models tested in terms of return (Fischer and Krauss, 2018). These studies highlight the potential of LSTM models in capturing complex patterns in financial data. Despite their promise, it is essential to delve into the nuanced challenges and limitations faced by LSTM in diverse financial contexts, especially when considering the specific case of the FTSE/JSE Top 40.

Studies have suggested that while LSTM exhibits superior performance in various financial exchanges. There is a concern involving the temporal generalization of LSTM. Questions arise about its ability to maintain predictive accuracy over extended periods, particularly when forecasting stock movements over long-term investment horizons (Budiharto, 2021; Fischer and Krauss, 2018). Model robustness is also a critical consideration, as LSTM models may exhibit sensitivity to changes in market conditions, economic events, or unforeseen shocks (Ji et al., 2021; Fischer and Krauss, 2018).

To mitigate concerns of overfitting and enhance generalization, employing robust cross-validation strategies becomes imperative. Techniques such as k-fold cross-validation can help assess the model's performance on different subsets of the data, providing valuable insights into its stability across various scenarios (Budiharto, 2021; Fischer and Krauss, 2018). Rigorous model calibration, including careful selection of hyperparameters, is essential to ensure optimal performance across different datasets (Ji et al., 2021; Hochreiter and Schmidhuber, 1997).

Additionally, conducting scenario analyses and stress testing can further enhance the model's robustness. This involves evaluating its performance under various market conditions, thereby identifying the boundaries of the model's predictive capabilities and its response to extreme or unexpected events (Budiharto, 2021; Fischer and Krauss, 2018).

2.2 Overview of The South African Market

Markets are classified into three different categories, based on its economic growth and maturity, these categories are developing markets which are characterized by low-income levels, limited industrialization, and often rely on agricultural (Deb, 2022). Emerging markets are in a transitional phase, moving from a developing to a developed class. They exhibit economic growth, increasing industrialization, and improving financial markets, but still face challenges such as political instability and lower per capita income compared to developed nations (Diamonte, Liew and Stevens, 1996; Li and Maskin, 2021). Finally, developed markets are highly industrialized with advanced technological infrastructure, high per capita income, and mature financial system. Further, developed economies have stable political environment and well established regulatory frameworks (Lin, 2011). The distinctions of these markets are crucial for investors seeking to diversify their portfolio and manage risk effectively (Abid and Rault, 2021).

South Africa is an emerging market within which the JSE, the leading stock market in the Southern Development Community (SADC) and in Africa as a whole (Hearn, Piesse and Strange, 2010). Although, research is limited when it comes to developing and emerging countries as opposed to developed countries as noted by the literature review of Kumbure, et.al. (2022) where the authors examined 138 journal articles in the subject of predicting stock prices using machine learning and the majority were conducted in developed economies, with the United States being at the top with 62 focused studies. Studying such markets is imperative, since developing and emerging markets are deemed to induce greater risk and reward, they appeal to international investors (Nur Ozkan-Gunay and Ozkan, 2007). Therefore, leading to a strong local financial sector empowering emerging markets to generate more domestic assets, thereby attracting foreign investment, and robust domestic capital market can benefit developing and emerging countries by diversifying fund sources and facilitating domestic currency financing (Boamah, 2022).

The FTSE/JSE Top 40 index is regarded as the primary index in the JSE, encompassing the top 40 companies by the investable market value. It holds immense significance for passive funds and, by the end of 2017, comprised 82% of the total capitalization of shares listed on the JSE (Kotze, 2017). Harmse (2017), compared the JSE Top 40 index with Multinational Entities (MNE), and noted that the capital structure of the JSE Top 40 is similar to that of MNE shares listed on the Fortune 500, excluding the banking industry (Harmse, 2017). Since there is similarity between the MNE and JSE, this in turn may indicate the potential for LSTM to offer valuable predictive insights for the JSE, as it has for other markets.

3. Methodology

3.1. Research Design

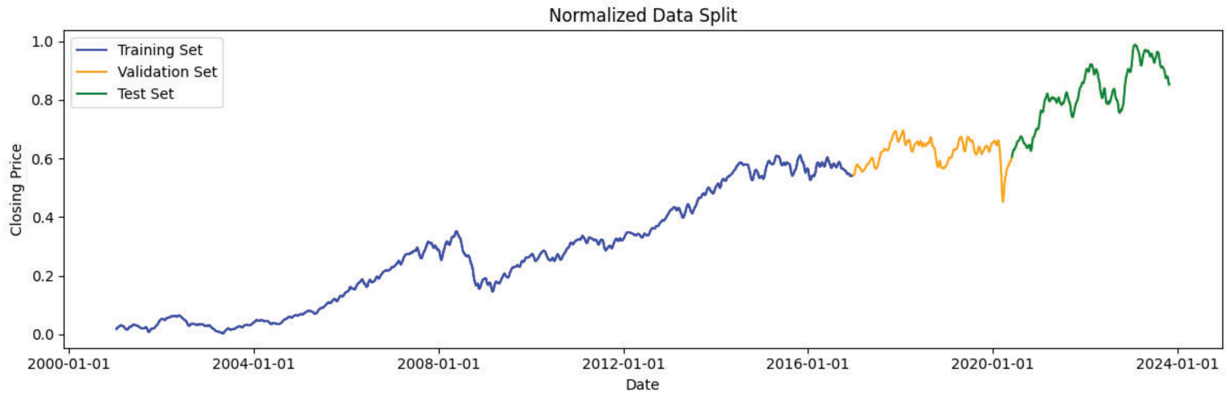
The aim of this research is to create a predictive analytics framework leveraging Long Short-Term Memory (LSTM) model, with the goal of forecasting live-stock price using the FTSE/JSE Top 40 index. This research aims to use the LSTM model to establish a predictive model suited for the FTSE/JSE top 40 index, as a tool for real-time decision support.

3.2. Data Acquisition and Pre-Processing

The data for this study is sourced from the Bloomberg Terminal, a comprehensive software system providing financial data developed by Bloomberg L.P. Specifically, the FTSE/JSE Top 40 daily index data is used to build the predictive model. The dataset consists of daily opening, closing, highest, and lowest prices, spanning from January 2001 to 20th November 2023, while the live prediction compromised a period extending from November 21st, 2023, until January 3rd, 2024. All stock prices are denoted in South African Rand (ZAR). Although the data include opening, highest, lowest, and closing prices, the model is built using closing prices solely, as it performed best using only closing prices.

The data pre-processing phase is vital for ensuring the quality and reliability of the dataset. In this phase the raw data is cleaned and transformed, by removing missing values, through dropping any columns containing a missing value because, having missing values in a data frame could adversely affect the performance of the LSTM model. The data is then standardized and normalized, ensuring consistency and comparability and help mitigate outliers across different features. To achieve this the min-max scaler is used, which shifts all the values to be between 0 and 1. The data is then made stationary through seasonal decomposition, a technique that isolates and removes seasonal patterns from the time series data. By extracting the seasonality, the time series is detrended and deseasonalized, resulting in a stationary dataset with a constant mean and variance over time, making the dataset suitable for further analysis and model building. Following, the data is then split into a training, validation, and testing data sets, composed of 70% of the data being for training, and 15% for validation and testing each as shown in Figure 1 below showing the data split between training, validation and testing. This rigorous pre-processing lays the foundation for accurate model training and analysis.

Figure 1: This figure shows the how the data is split for training, validation, and testing.



3.3. Long-Short Term Memory (LSTM)

LSTM is a class of recurrent neural network (RNN) tailored to capture long-term dependencies and patterns in time-series data. Its unique architecture, featuring memory cells and multiplicative units, enables it to effectively model the complex dynamics of financial markets. The LSTM architecture includes memory blocks with self-connected memory cells and three multiplicative gates: input, output, and forget gates. The forget gate (f_t) decides which information to discard, while the input gate (i_t) and a candidate layer $(\widetilde{C_t})$ work together to update the cell state (C_t). The output gate (o_t), then make use of the updated cell state to produce the hidden state (H_t) at time t. This interaction of gates and states allows for a continuous process of reading, writing, and resetting the memory, enabling the LSTM to capture intricate relationships over prolonged periods (Hochreiter and Schmidhuber, 1997).

The process starts with the forget gate, which serves as the initial filter, to determine the relevance of the information from the preceding cell state (C_{t-1}). The forget gate computes the importance of each peace of information based on the concatenation to the previous hidden state (H_{t-1}) and the current input (X_t). Utilizing the weight matrix (W_f), and based on the bias term (b_f), the forget gate output is then passed through a sigmoid function (σ), producing a value ranging from 0 to 1. This value indicates the proportion of information to be discarded or reserved, denoted in equation 1 below.

$$f_t = \sigma\left(W_f \cdot [H_{t-1}, X_t] + b_f\right) \tag{1}$$

Once the forget gate determines the relevance of information, the LSTM proceeds to the input gate i_t. This gate controls which new information $\widetilde{C_t}$ ought to be appended to the current state C_t. It evaluates the significance of the incoming data, by examining the concatenation of the preceding hidden state with the current input. Similar to the forget gate, the input gate employs a weight matrix (W_i), a bias term b_i and a sigmoid function σ, to generate values ranging from 0 and 1, representing the amount of information to incorporate into the cell state, shown in equation 2 below.

$$i_t = \sigma(W_i \cdot [H_{t-1}, X_t] + b_i) \tag{2}$$

Following, the LSTM model computes new candidate values for the cell state C_t through the candidate layer $\widetilde{C_t}$. This layer evaluates the merging of the preceding hidden state with the current input using a weight matrix W_C, adds a bias term (b_C), then passes the results through the hyperbolic tangent function (tanh). The hyperbolic tangent function produces values ranging from -1 and 1, representing the candidate values for the new cell state, demonstrated in equation 3.

$$\widetilde{C_t} = \tanh(W_C \cdot [H_{t-1}, X_t] + b_C) \tag{3}$$

With the forget gate decision, input gate evaluation, and new candidate evaluation values determined, the model proceeds to update the cell state (C_t). Which involves blending information retained from the preceding cell state ($f_t * C_{t-1}$) with the relevant new information ($i_t * \widetilde{C_t}$). Overlooked by the forget and input gates, allowing the LSTM model to selectively reserve or disregard information over time, as in equation 4.

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{4}$$

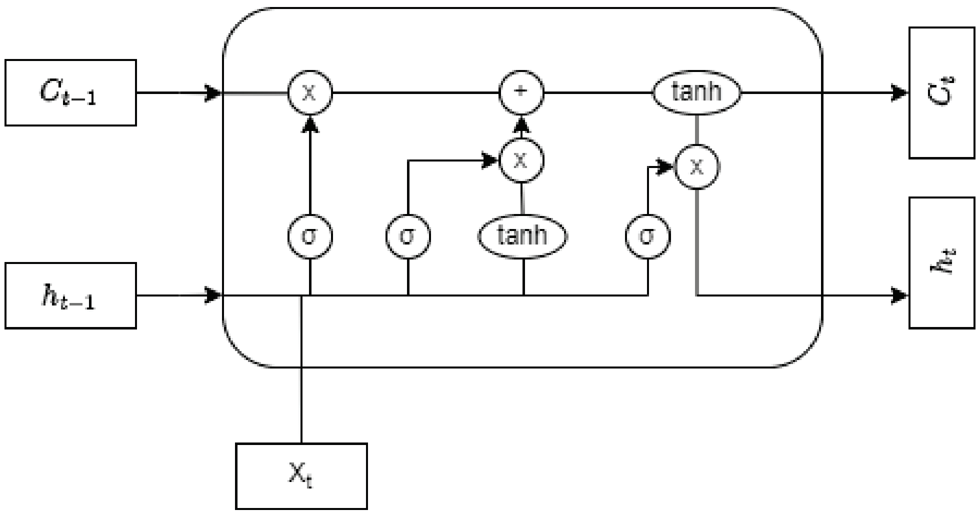
Finally, the model decides which data derived from the updated cell state (C_t) ought to be exposed through the output gate (o_t) and the resulting hidden state (H_t). The output gate evaluates the combination of the preceding hidden state with the present input using a weight matrix W_o, adds a bias term b_o, then passes the results through a sigmoid function (σ). The sigmoid function output establishes the extent to which the revised cell state ought to be propagated to the hidden state (H_t). The hidden state computation is done by multiplying the output gate with the hyperbolic tangent function, determining the final output, illustrated in equations 5 and 6 below.

$$o_t = \sigma(W_o \cdot [H_{t-1}, X_t] + b_o) \tag{5}$$

$$H_t = O_t * \tanh(C_t) \quad 6$$

This design of memory cells in LSTM models shown in Figure 2 below, facilitates its ability to learn and adapt to diverse and intricate patterns inherent in stock price movements. The process is summarized in Figure 5 below, illustrating how the LSTM model is applied in this study to predict the FTSE/JSE Top 40 index closing prices.

Figure 2: The structure of LSTM memory cell.



Source: (Olah, 2015)

3.4. Model Optimization

Optimizing a model for optimal performance is a crucial step in achieving the best possible results. Model tuning involves experimenting with various hyperparameters to determine which configuration yields superior results. In line with the approach by Bhandari et al. (2022), this study is adopting a similar model tuning methodology. In their study, Bhandari et al. (2022), conducted comprehensive experiments involving key hyperparameters such as the number of neurons, choice of optimizers, learning rates, and batch sizes. Similarly, the model was tuned using Adagrad, Adam and Nadam optimizers, learning rates of 0.1, 0,01 and 0.001, batch sizes of 4, 8 and 16 as well as different single-layer and multiple layers with different number of neurons. This approach ensures that the model is well-calibrated to deliver accurate and effective results.

3.5. Model Performance Evaluation

The performance of the model is evaluated using the Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE), which are commonly employed metrics for assessing the accuracy of predictive models. RMSE and MAE provide a quantitative measure of the model's predictive precision, with lower values indicating better performance explained in equation 7 and 8 below respectively. MAPE offers a relative measure of prediction accuracy, as detailed in question 9. Additionally, R-squared is used to measure how much of the variance is explained by the model given in equation 10 below. This robust combination of tools and methodologies ensures a comprehensive and effective exploration of LSTM's capabilities in predicting live-stock prices within the FTSE/JSE Top 40 index.

$$RSME = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y_i})^2} \tag{7}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \widehat{Y_i}| \tag{8}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i - \widehat{Y_i}|}{Y_i} \tag{9}$$

Where:
$i = variable$
$n = number\ of\ data\ points$
$Y_i = observed\ values$
$\widehat{Y_i} = estimated\ observation$

$$R^2 = 1 - \frac{RSS}{TSS} \tag{10}$$

Where:
$R^2 = coefficient\ of\ determination$
$RSS = sum\ of\ square\ residuals$
$TSS = total\ sum\ of\ squares$

4. Results and Discussion

4.1. Model Development and training

The development phase of the LSTM model was approached with a focus on predictive accuracy and computational feasibility. The finalized architecture featured a single LSTM layer to capture complex temporal dependencies within the index data. Training was conducted using a split of historical data, with validation sets employed to monitor and mitigate overfitting. A total of 189 models were tested with different configurations. The model's parameters were fine-tuned through iterative training rounds to optimize performance metrics, preparing the LSTM for the eventual challenge of live data prediction. The best model came out to be using Adam optimizer, with a batch size of 16, a single layer of 100 neurons, and a learning rate of 0.01, as shown in the Table 1: Top 10 performing models based on Mean Absolute Error (MAE). below. This aligns with Bhandari et al (2022) best model's parameters with regards to the number of layers and the optimizer, but differs in terms of the learning rate, the batch size, and the number of neurons, where in their research they had the best results with 10 neurons, a batch size of 8 and a learning rate of 0.001.

Table 1: Top 10 performing models based on Mean Absolute Error (MAE).

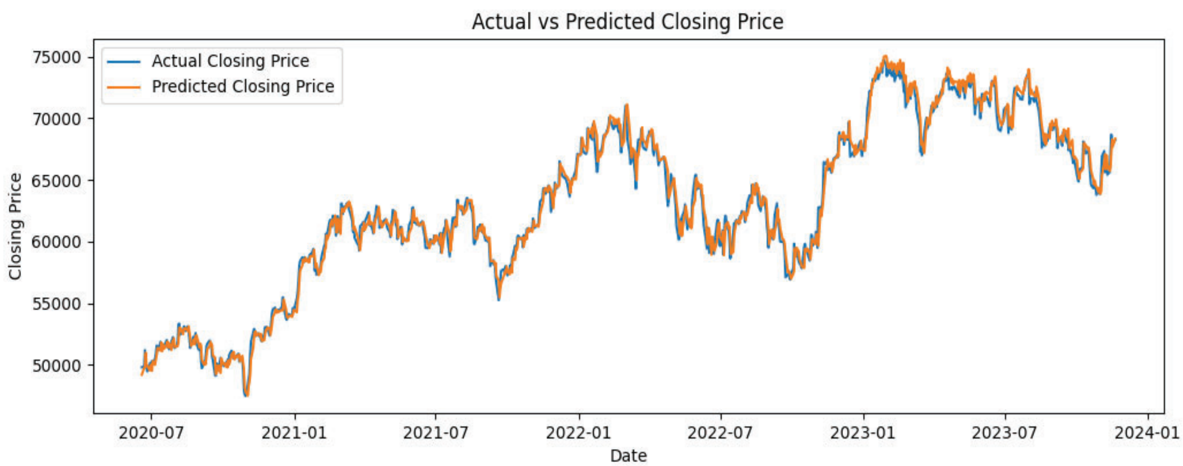
Optimizer	Learning Rate	Batch Size	Neuron Configuration	MAE	RMSE	R-squared
Adam	0.01	16	100	321.98	475.02	0.9993
Adam	0.01	8	10	332.29	489.29	0.9993
Adam	0.01	8	30	333.91	481.33	0.9993
Nadam	0.1	8	10	334.98	491.85	0.9993
Adam	0.001	4	100	346.28	490.25	0.9993
Adam	0.001	4	(10, 5)	349.14	508.07	0.9992
Adam	0.001	16	10	353.88	522.98	0.9992
Adam	0.01	16	10	357.42	508.77	0.9992
Adam	0.1	16	10	359.05	506.52	0.9992
Adam	0.001	4	30	363.31	522.55	0.9992

4.2. Performance on Historical Data

Evaluation metrics demonstrated the LSTM model's adeptness in interpreting historical trends of the JSE Top 40 index. The model displayed a promising Mean Absolute Error (MAE), indicating high accuracy in capturing the index's movements. These results set a benchmark for expected performance, providing a comparison point for live data predictions. In particular, the best MAE score recorded was 584.50. Indicating that on average the model has a mean error of R584.50 different from the actual value. The RMSE score on the other hand was 475.02, and this is the value that the model predictions deviated from the actual index by on average. Since the RMSE penalizes large errors, the difference between the MSE and the RMSE points to the presence of some large errors in the model's prediction.

Furthermore, the model attained a Mean Absolute Percentage Error (MAPE) of 0.93%, signifying high accuracy and surpassing the performance reported by Bharandi et al. (2022). Additionally, the model has also outperformed the results achieved by Hiransha et al. (2018) in terms of MAPE where in their study the authors achieved a MAPE ranging between 6.37% and 8.13% with regards to National Stock Exchange (NSE) and an average of 7.495% in terms of New York Stock Exchange (NYSE) stocks. Furthermore, the model had similar results with Chung and shin (2018) where the authors attained a MAPE of 0.91% on the Korean Stock Price Index (KSPI). The performance of the LSTM model is visualized in Figure 3 below.

Figure 3: This figure shows the LSTM model performance, comparing between the actual and the predicted prices of the FTSE/JSE Top 40.



4.3. Forecasting Capability

When subjected to live data, the LSTM model was evaluated for its real-time prediction capabilities. The model uses the preceding 3 days values to predict the next day's value. A dedicated data pipeline was established to simulate live data feed, and the model's predictions were recorded in real-time. Despite the challenges of latency and dynamic data flow, the model maintained a commendable level of accuracy, affirming its potential utility in a live trading environment. With forecasting the model performed relatively poor compared to historical data. The model yielded a mean absolute error (MAE) of 772.24, a root mean squared error (RMSE) of 1060.79 and an R squared score of -0.19. These scores indicate the model incapability of accurately predicting the FTSE/JSE Top 40 closing prices using the parameters used in this paper. To evaluate forecasting capability the model was run to forecast one value at a time and then appending the actual value to the data and predicting the next value as it would be in a real-world scenario.

4.4. Comparison Between Historical and Live Data Prediction

Contrasting the model's performance on historical versus live data unveiled insights into its adaptability and reliability. While the model exhibited proficiency in interpreting historical trends of the JSE Top 40 index, its performance with live data was notably different. The model's ability to accurately predict live stock price data was less effective compared to its performance with historical data, aligning with the study of Deora et.al. (2019). This discrepancy can be attributed to the inherent unpredictability and volatility of the live stock market conditions, which pose a significant challenge to the model's predictive capabilities. The results indicate that while LSTM models are adept at learning and identifying patterns within historical data, their application in a real-time trading environment is more complex. This complexity is due to the rapidly changing market conditions that the live data encompasses. Therefore, while the model's performance with historical data sets a high benchmark, its effectiveness in real-time scenario, as observed during the testing phase, is limited.

The real-time data stream introduced unique challenges, including the need for instantaneous data processing and model response. Techniques to handle these included implementing efficient data handling algorithms and optimizing the model's architecture for speed without compromising predictive integrity.

In the assessment of the LSTM model's robustness and stability, the model was evaluated over a 27-day period of live data prediction. This relatively short time frame revealed significant variability in its performance. The model displayed a mixed ability to adapt to market fluctuations, as evidenced by its inconsistent accuracy. While it showed promise on certain days, on others, the results were considerably weaker. Shown in Table 4: Difference between actual and predicted prices, using live below.

4.5. Limitations in Predicting Live Data

In exploring the LSTM model's application to live data prediction, several notable limitations were identified, that are essential for understanding its current performance and guiding future enhancements.

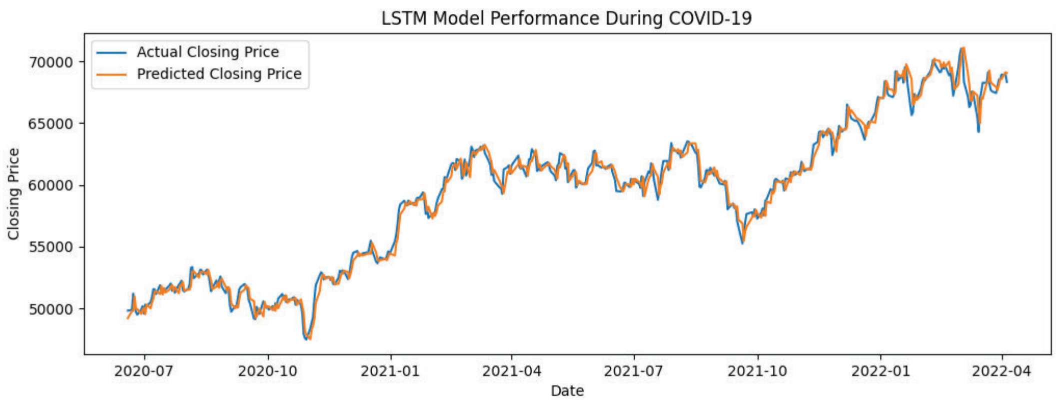
Despite the Mean Absolute Percentage Error (MAPE) being a low 1.19%, indicating a high accuracy, a primary limitation observed in the model's performance is its susceptibility to noise within the data stream. This sensitivity to noise was particularly apparent in the Root Mean Squared Error (RMSE) score, which stood at a high 1060.79. The disparity between the low MAPE and the high RMSE reveals that while the model has very high accurate prediction, it exhibits significant inaccuracies as well. Given that the RMSE, known for penalizing larger errors more significantly, the score suggests that the model may be prone to substantial deviations in its predictions when confronted with irregular or noisy data. This characteristic poses a challenge in the fast-paced and often volatile environment of live financial markets, where data can be subject to rapid and unexpected changes.

Understanding these limitations is essential for accurately contextualizing the model's current capabilities. While the LSTM model has shown potential in handling and predicting live data, its tendency to be influenced by noise and the resulting large errors indicate areas that require further refinement. This understanding helps set realistic expectations of the model's performance in a live setting and underscores the importance of continuous improvement.

4.6. Economic Events and Model Responsiveness

The testing phase of our model coincided with a significant part of the COVID-19 era, which spanned from March 5, 2020, to April 5, 2022. However, our specific evaluation period began on June 19, 2020, and continued through this era and beyond. In this challenging context, the model demonstrated remarkable adaptability, as evidenced by its performance metrics: a Mean Absolute Error (MAE) of 522.36 and a Root Mean Squared Error (RMSE) of 684.23, alongside a notable R-squared value of 0.98. The model with a Mean Absolute Percentage Error (MAPE) of 0.88% performed on par with that developed by Budiharto (2021), where the author tested LSTM model in the Indonesian stock market during COVID. These statistics reflect the model's ability to effectively adapt to the turbulent market conditions brought on by the pandemic. The high R-squared value indicates a strong correlation between the model's predictions and the actual market movements during this period. For a clearer view of the model's journey through these unpredictable times, Figure 4 below offers an insightful visualization, showcasing the model's predictive ability during COVID-19.

Figure 4: This figure shows the performance of LSTM model during the COVID-19 era in South Africa.



4.7. Limitations in Predicting Live Data

In exploring the LSTM model's application to live data prediction, several notable limitations were identified, that are essential for understanding its current performance and guiding future enhancements.

Despite the Mean Absolute Percentage Error (MAPE) being a low 1.19%, indicating a high accuracy, a primary limitation observed in the model's performance is its susceptibility to noise within the data stream. This sensitivity to noise was particularly apparent in the Root Mean Squared Error (RMSE) score, which stood at a high 1060.79. The disparity between the low MAPE and the high RMSE reveals that while the model has remarkably high accurate prediction, it exhibits significant inaccuracies as well. Given that the RMSE, known for penalizing larger errors more significantly, suggesting that the model may be prone to substantial deviations in its predictions when confronted with irregular or noisy data. This characteristic poses a challenge in the fast-paced and often volatile environment of live financial markets, where data can be subject to rapid and unexpected changes.

Understanding these limitations is essential for accurately contextualizing the model's current capabilities. While the LSTM model has shown potential in handling and predicting live data, its tendency to be influenced by noise and the resulting large errors indicate areas that require further refinement. This understanding helps set realistic expectations of the model's performance in a live setting and underscores the importance of continuous improvement.

4.8. Implications for Traders and Investors

The study's findings on the LSTM model's performance in predicting the JSE Top 40 index offer mixed implications for traders and investors. While the model exhibited potential, its inconsistent reliability in the current state suggests caution in its application. The LSTM model, in its present form, may best serve as a supplementary tool in the arsenal of traders and investors. Its use could provide additional insights when making trading decisions, but it should not be the sole basis for such decisions due to its varying accuracy.

It is crucial for traders and investors to integrate the LSTM model into a broader, more comprehensive investment strategy. Relying solely on this model, given its current limitations, might not yield optimal outcomes. Instead, it should be used in conjunction with other analytical tools and market indicators to enhance decision-making processes.

4.9. Future Enhancements and Research Directions

The exploration of the LSTM model's application in predicting the JSE Top 40 index has opened avenues for future enhancements and research directions. Given the model's varying performance and sensitivity to data noise, a critical focus will be on refining its data processing capabilities, potentially through advanced filtering techniques or integrating more robust architectures. Emphasis on incorporating broader economic indicators and sentiment analysis could enrich the model's input, thereby enhancing its predictive accuracy. Furthermore, exploring the synergies between LSTM and other advanced neural network architectures, or even delving into the realm of reinforcement learning, could pave the way for a more dynamic and self-optimizing model. Such developments would not only improve the model's performance in real-time scenarios but also broaden its applicability in complex financial market analytics. This ongoing research and development are essential in evolving the LSTM model into a more reliable tool for financial forecasting, offering a more comprehensive understanding of market dynamics.

5. Conclusion and Future Work

This study embarked on evaluating the effectiveness of Long Short-Term Memory (LSTM) models in predicting the performance of the South African FTSE/JSE Top 40 index, covering both historical and real-time data. The aim was to determine whether these advanced machine learning models could serve as reliable tools in the complex and dynamic realm of stock market forecasting.

The research highlighted that LSTM models are proficient at analyzing historical stock market data. The model demonstrated a notable ability to identify and learn from patterns within past market trends, which is an asset for understanding historical market dynamics. This suggests that LSTMs could be a useful tool for retrospective analysis in stock market research, providing insights into past market behaviors.

However, the study revealed that LSTMs face significant challenges when applied to predicting live, real-time stock market data. The unpredictability and rapid fluctuations inherent in live stock market data posed a hurdle, leading to less accurate predictions compared to historical data analysis. This indicates that while LSTMs have potential in stock market forecasting, their current application in live data prediction is limited.

The research also shows the need for further improvements and adaptations in LSTM models for stock market forecasting. Future research could focus on integrating more diverse and comprehensive data sets, experimenting with different model architectures, and possibly combining LSTM with other machine learning techniques to enhance accuracy and reliability in predicting the FTSE/JSE Top 40 index, especially in live market conditions.

In conclusion, the findings of this study provide a nuanced understanding of the capabilities and limitations of LSTM models in stock market prediction. While they show promise in historical data analysis, their application in real-time forecasting requires cautious interpretation and further development. For traders, investors, and analysts, LSTMs can be a valuable part of a broader set of tools for market analysis, but they should not be relied upon exclusively, particularly for making real-time trading decisions. The path forward includes continued research and model refinement within the South African stock market and prolonging the time horizon for live testing, aiming for more robust and reliable LSTM applications in stock market forecasting.

References

[1] Abarbanell, J.S. and Bushee, B.J., 1998. Abnormal returns to a fundamental analysis strategy. *Accounting Review*, 73(1), pp.19-45.

[2] Abid, A. and Rault, C., 2021. On the exchange rates volatility and economic policy uncertainty nexus: A panel VAR approach for emerging markets. *Journal of Quantitative Economics*, *19*, pp.403-425.

[3] Ayodele, T.O., 2010. Machine learning overview. *New Advances in Machine Learning*, *2*(9-19), pp.13-16.

[4] Bagul, J., Warkhade, P., Gangwal, T. and Mangaonkar, N., 2022, December. Arima vs lstm algorithm–a comparative study based on stock market prediction. In *2022 5th International Conference on Advances in Science and Technology (ICAST),* pp.49-53. IEEE.

[5] Banik, S., Sharma, N., Mangla, M., Mohanty, S.N. and Shitharth, S., 2022. LSTM based decision support system for swing trading in stock market. *Knowledge-Based Systems*, *239*, p.107994.

[6] Bhandari, H.N., Rimal, B., Pokhrel, N.R., Rimal, R., Dahal, K.R. and Khatri, R.K., 2022. Predicting stock market index using LSTM. *Machine Learning with Applications*, *9*, p.100320.

[7] Boamah, N.A., 2022. Segmentation, business environment and global informational efficiency of emerging financial markets. *The Quarterly Review of Economics and Finance*, *84*, pp.52-60.

[8] Budiharto, W., 2021. Data science approach to stock prices forecasting in Indonesia during Covid-19 using Long Short-Term Memory (LSTM). *Journal of Big Data*, *8*(1), pp.1-9.

[9] Charpentier, A., Elie, R. and Remlinger, C., 2021. Reinforcement learning in economics and finance. *Computational Economics*, *62*(1) pp.1-38.

[10] Chung, H. and Shin, K.S., 2018. Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability*, *10*(10), p.3765.

[11] Cochrane, J.H., 2011. Presidential address: Discount rates. *The Journal of Finance*, *66*(4), pp.1047-1108.

[12] Deb, K., 2022. Trade Structure and Economic Growth in Emerging and Developing Asia: A Dynamic Panel Analysis. *Journal of Quantitative Economics*, *20*(3), pp.601-628.

[13] Deora, O.S., Jha, P., Patil, S.S., Patil, T.B. and Joshi, S.D., 2019. Monitoring and Training Stock Prediction System for Historical & Live Dataset using Lstm & Cnn.

[14] Diamonte, R.L., Liew, J.M. and Stevens, R.L., 1996. Political risk in emerging and developed markets. *Financial Analysts Journal*, *52*(3), pp.71-76.

[15] Fama, E.F., 1965. The behavior of stock-market prices. *The Journal of Business*, *38*(1), pp.34-105.

[16] Fan, D., Sun, H., Yao, J., Zhang, K., Yan, X. and Sun, Z., 2021. Well production forecasting based on ARIMA-LSTM model considering manual operations. *Energy*, *220*, p.119708.

[17] Fischer, T. and Krauss, C., 2018. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), pp.654-669.

[18] Gardner Jr, E.S., 2006. Exponential smoothing: The state of the art—Part II. *International Journal of Forecasting*, *22*(4), pp.637-666.

[19] Ghosh, P., Neufeld, A. and Sahoo, J.K., 2022. Forecasting directional movements of stock prices for intraday trading using LSTM and random forests. *Finance Research Letters*, *46*, p.102280.

[20] Grudnitski, G. and Osburn, L., 1993. Forecasting S&P and gold futures prices: An application of neural networks. *Journal of Futures Markets*, *13*(6), pp.631-643.

[21] Harmse, L.H., 2017. A comparison of the capital structures of the top 40 multinational entities and the top 40 JSE-listed entities. *International Journal of Social Sciences and Humanity Studies*, *9*(1), pp.121-135.

[22] Hearn, B., Piesse, J. and Strange, R., 2010. Market liquidity and stock size premia in emerging financial markets: The implications for foreign investment. *International Business Review*, *19*(5), pp.489-501.

[23] Hiransha, M., Gopalakrishnan, E.A., Menon, V.K. and Soman, K.P., 2018. NSE stock market prediction using deep-learning models. *Procedia Computer Science*, *132*, pp.1351-1362.

[24] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural Computation*, *9*(8), pp.1735-1780.

[25] Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z. and Zhang, H., 2019. Deep learning with long short-term memory for time series prediction. *IEEE Communications Magazine*, *57*(6), pp.114-119.

[26] Jensen, M.C., 1978. Some anomalous evidence regarding market efficiency. *Journal of Financial Economics*, *6*(2/3), pp.95-101.

[27] Ji, Y., Liew, A.W.C. and Yang, L., 2021. A novel improved particle swarm optimization with long-short term memory hybrid model for stock indices forecast. *Ieee Access*, *9*, pp.23660-23671.
[28] Kimoto, T., Asakawa, K., Yoda, M. and Takeoka, M., 1990, June. Stock market prediction system with modular neural networks. In *1990 IJCNN International Joint Conference on Neural Networks, 1,* pp. 1-6. IEEE.
[29] Kotze, A., 2017. FTSE/JSE Top 40 index long-term returns. *Available at SSRN 2978093*.
[30] Li, D.D. and Maskin, E.S., 2021. Government and economics: An emerging field of study. *Journal of Government and Economics*, *1*, p.100005.
[31] Lin, J.Y., 2011. New structural economics: A framework for rethinking development. *The World Bank Research Observer*, *26*(2), pp.193-221.
[32] Liu, Q., Tao, Z., Tse, Y. and Wang, C., 2022. Stock market prediction with deep learning: The case of China. *Finance Research Letters*, *46*, p.102209.
[33] Nabipour, M., Nayyeri, P., Jabani, H., Shahab, S. and Mosavi, A., 2020. Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. *IEEE Access*, *8*, pp.150199-150212.
[34] Nti, I.K., Adekoya, A.F. and Weyori, B.A., 2020. A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, *53*(4), pp.3007-3057.
[35] Nur Ozkan-Gunay, E. and Ozkan, M., 2007. Prediction of bank failures in emerging financial markets: an ANN approach. *The Journal of Risk Finance*, *8*(5), pp.465-480.
[36] Olah, C., 2015. Understanding lstm networks. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs [2023, December 24]
[37] Park, C.H. and Irwin, S.H., 2007. What do we know about the profitability of technical analysis?. *Journal of Economic Surveys*, *21*(4), pp.786-826.
[38] Patel, J., Shah, S., Thakkar, P. and Kotecha, K., 2015. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), pp.259-268.
[39] Peng, Y., Albuquerque, P.H.M., Kimura, H. and Saavedra, C.A.P.B., 2021. Feature selection and deep neural networks for stock price direction forecasting using technical analysis indicators. *Machine Learning with Applications*, *5*, p.100060.
[40] Pirani, M., Thakkar, P., Jivrani, P., Bohara, M.H. and Garg, D., 2022, April. A comparative analysis of ARIMA, GRU, LSTM and BiLSTM on financial time series forecasting. In *2022 IEEE*

International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), pp. 1-6. IEEE.

[41] Samuelson, P.A., 1965. Proof that properly anticipated prices fluctuate randomly. *Industrial Management Review,* 6(2), pp. 41-50.

[42] Siami-Namini, S., Tavakoli, N. and Namin, A.S., 2018, December. A comparison of ARIMA and LSTM in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA),* pp.1394-1401. Ieee.

Tables and Figures

Table 2: This table shows the results of the optimization for a single layer of neurons.

Optimizer	Learning Rate	Batch Size	Neuron Configuration	MAE	RMSE	R-squared
Adagrad	0.1	4	10	389.0521	569.1421	0.999083
Adagrad	0.1	4	30	495.0977	647.1386	0.998815
Adagrad	0.1	4	50	467.179	650.9353	0.998801
Adagrad	0.1	4	100	381.2076	563.5212	0.999101
Adagrad	0.1	8	10	404.348	583.422	0.999037
Adagrad	0.1	8	30	448.9855	618.1937	0.998918
Adagrad	0.1	8	50	372.5925	560.2635	0.999112
Adagrad	0.1	8	100	377.1445	560.8238	0.99911
Adagrad	0.1	16	10	376.318	556.063	0.999125
Adagrad	0.1	16	30	442.2149	619.2818	0.998914
Adagrad	0.1	16	50	390.2376	566.5264	0.999092
Adagrad	0.1	16	100	467.8808	639.7291	0.998842
Adagrad	0.01	4	10	507.7052	661.7429	0.998761
Adagrad	0.01	4	30	439.4259	607.0945	0.998957
Adagrad	0.01	4	50	404.0313	576.4933	0.999059
Adagrad	0.01	4	100	376.5748	559.3365	0.999114
Adagrad	0.01	8	10	1020.615	1191.766	0.99598
Adagrad	0.01	8	30	396.3029	569.1244	0.999083
Adagrad	0.01	8	50	380.8807	567.2741	0.999089
Adagrad	0.01	8	100	379.3731	564.9952	0.999096
Adagrad	0.01	16	10	16505.77	18798.89	-0.00029
Adagrad	0.01	16	30	391.4346	572.1748	0.999073
Adagrad	0.01	16	50	441.2582	601.9732	0.998974
Adagrad	0.01	16	100	389.9607	566.3491	0.999092
Adagrad	0.001	4	10	5798.927	6665.011	0.874263
Adagrad	0.001	4	30	4885.314	5638.962	0.909996
Adagrad	0.001	4	50	2397.863	2813.175	0.9776
Adagrad	0.001	4	100	1432.26	1714.019	0.991684
Adagrad	0.001	8	10	11374.42	13385.46	0.492858
Adagrad	0.001	8	30	4826.07	5562.167	0.912431
Adagrad	0.001	8	50	4949.182	5684.901	0.908524
Adagrad	0.001	8	100	6906.475	7907.052	0.823033
Adagrad	0.001	16	10	6599.035	7557.763	0.838323
Adagrad	0.001	16	30	6326.152	7235.856	0.851802
Adagrad	0.001	16	50	10028.72	11783.11	0.607009
Adagrad	0.001	16	100	7509.186	8578.314	0.791711
Adam	0.1	4	10	560.5107	756.4321	0.99838
Adam	0.1	4	30	1304.459	1428.774	0.994222

Adam	0.1	4	50	700.7371	880.9502	0.997803
Adam	0.1	4	100	425.16	617.9965	0.998919
Adam	0.1	8	10	436.4482	572.9318	0.999071
Adam	0.1	8	30	412.8116	589.0113	0.999018
Adam	0.1	8	50	16418.96	18895.9	-0.01064
Adam	0.1	8	100	550.4678	677.7726	0.9987
Adam	0.1	16	10	359.0583	506.5209	0.999274
Adam	0.1	16	30	1142.611	1247.729	0.995593
Adam	0.1	16	50	557.3145	667.9269	0.998737
Adam	0.1	16	100	732.4334	846.0243	0.997974
Adam	0.01	4	10	678.6905	815.12	0.998119
Adam	0.01	4	30	550.6485	654.6597	0.998787
Adam	0.01	4	50	479.3897	580.2658	0.999047
Adam	0.01	4	100	428.9362	538.7727	0.999178
Adam	0.01	8	10	332.2975	489.2964	0.999322
Adam	0.01	8	30	333.9155	481.3399	0.999344
Adam	0.01	8	50	407.8171	532.9162	0.999196
Adam	0.01	8	100	386.3339	531.1557	0.999201
Adam	0.01	16	10	357.4255	508.7718	0.999267
Adam	0.01	16	30	489.0977	625.0924	0.998894
Adam	0.01	16	50	396.5478	576.6693	0.999059
Adam	0.01	16	100	**321.9808**	**475.0221**	0.999361
Adam	0.001	4	10	387.5534	529.5232	0.999206
Adam	0.001	4	30	363.31	522.557	0.999227
Adam	0.001	4	50	486.9871	619.1047	0.998915
Adam	0.001	4	100	346.2868	490.2578	0.99932
Adam	0.001	8	10	531.2054	672.9186	0.998718
Adam	0.001	8	30	423.7578	592.875	0.999005
Adam	0.001	8	50	439.4081	578.129	0.999054
Adam	0.001	8	100	400.8066	529.2682	0.999207
Adam	0.001	16	10	353.8885	522.9837	0.999226
Adam	0.001	16	30	377.9742	515.5193	0.999248
Adam	0.001	16	50	418.7582	581.9401	0.999041
Adam	0.001	16	100	446.2557	601.7648	0.998975
Nadam	0.1	4	10	607.3402	851.673	0.997947
Nadam	0.1	4	30	1056.32	1221.103	0.995779
Nadam	0.1	4	50	587.8671	843.2318	0.997987
Nadam	0.1	4	100	16391.6	19035.87	-0.02567
Nadam	0.1	8	10	334.9882	491.8576	0.999315
Nadam	0.1	8	30	719.4569	835.2566	0.998025
Nadam	0.1	8	50	459.1048	593.9471	0.999001
Nadam	0.1	8	100	440.2417	567.739	0.999088
Nadam	0.1	16	10	1135.304	1220.866	0.995781

Nadam	0.1	16	30	540.7176	705.2982	0.998592
Nadam	0.1	16	50	1077.112	1326.856	0.995017
Nadam	0.1	16	100	1136.217	1331.001	0.994986
Nadam	0.01	4	10	555.9191	691.4936	0.998647
Nadam	0.01	4	30	647.0438	813.5033	0.998127
Nadam	0.01	4	50	555.4046	662.1554	0.998759
Nadam	0.01	4	100	1227.101	1335.777	0.99495
Nadam	0.01	8	10	471.8932	614.7597	0.99893
Nadam	0.01	8	30	399.8197	528.3635	0.99921
Nadam	0.01	8	50	624.9427	819.1494	0.998101
Nadam	0.01	8	100	689.2055	812.6547	0.998131
Nadam	0.01	16	10	572.0545	741.9919	0.998442
Nadam	0.01	16	30	579.0726	757.0622	0.998378
Nadam	0.01	16	50	378.0316	550.199	0.999143
Nadam	0.01	16	100	953.8146	1158.983	0.996198
Nadam	0.001	4	10	407.5366	602.7313	0.998972
Nadam	0.001	4	30	392.9228	566.3426	0.999092
Nadam	0.001	4	50	394.0607	586.9683	0.999025
Nadam	0.001	4	100	460.0217	636.4941	0.998853
Nadam	0.001	8	10	405.3199	590.9151	0.999012
Nadam	0.001	8	30	389.5459	578.6913	0.999052
Nadam	0.001	8	50	530.0048	735.4512	0.998469
Nadam	0.001	8	100	477.6407	675.6418	0.998708
Nadam	0.001	16	10	390.5752	566.7506	0.999091
Nadam	0.001	16	30	399.044	593.2802	0.999004
Nadam	0.001	16	50	400.8857	590.03	0.999015
Nadam	0.001	16	100	408.654	608.739	0.998951

Table 3: This table shows the results for different configurations using multiple layers.

Optimizer	Learning Rate	Batch Size	Neuron Configuration	MAE	RMSE	R-squared
Adagrad	0.1	4	(10, 5)	475.5309	675.6312	0.998708
Adagrad	0.1	4	(20, 10)	436.5166	653.3995	0.998792
Adagrad	0.1	4	(50, 20)	457.2625	658.727	0.998772
Adagrad	0.1	8	(10, 5)	16528.85	18810.42	-0.00152
Adagrad	0.1	8	(20, 10)	410.2068	618.4169	0.998918
Adagrad	0.1	8	(50, 20)	433.6174	637.4697	0.99885
Adagrad	0.1	16	(10, 5)	642.9977	817.897	0.998107
Adagrad	0.1	16	(20, 10)	473.8081	667.7477	0.998738
Adagrad	0.1	16	(50, 20)	864.7031	1045.264	0.996907
Adagrad	0.01	4	(10, 5)	510.2048	688.1844	0.998659
Adagrad	0.01	4	(20, 10)	501.4696	680.6663	0.998689
Adagrad	0.01	4	(50, 20)	496.5488	676.8103	0.998703
Adagrad	0.01	8	(10, 5)	1656.208	1940.173	0.989345
Adagrad	0.01	8	(20, 10)	748.6235	918.1914	0.997614
Adagrad	0.01	8	(50, 20)	492.5016	678.7181	0.998696
Adagrad	0.01	16	(10, 5)	1065.022	1615.499	0.992613
Adagrad	0.01	16	(20, 10)	734.8795	883.4986	0.997791
Adagrad	0.01	16	(50, 20)	540.5005	712.5686	0.998563
Adagrad	0.001	4	(10, 5)	19786.11	24592.09	-0.7118
Adagrad	0.001	4	(20, 10)	9731.462	11128.21	0.649479
Adagrad	0.001	4	(50, 20)	10191.8	11632.02	0.617022
Adagrad	0.001	8	(10, 5)	15079.63	18373.28	0.044487
Adagrad	0.001	8	(20, 10)	12277.05	14011.5	0.44431
Adagrad	0.001	8	(50, 20)	13294.41	15459.22	0.323546
Adagrad	0.001	16	(10, 5)	13244.81	16008.96	0.27458
Adagrad	0.001	16	(20, 10)	18176	22465.23	-0.42851
Adagrad	0.001	16	(50, 20)	18207.45	22569.94	-0.44186
Adam	0.1	4	(10, 5)	3322.316	3405.251	0.967178
Adam	0.1	4	(20, 10)	16394.06	19303.77	-0.05474
Adam	0.1	4	(50, 20)	16648.39	18937.61	-0.01511
Adam	0.1	8	(10, 5)	16704.68	19013.99	-0.02332
Adam	0.1	8	(20, 10)	16903.34	19403.79	-0.0657
Adam	0.1	8	(50, 20)	870.2138	1011.122	0.997106
Adam	0.1	16	(10, 5)	654.2783	762.2752	0.998355
Adam	0.1	16	(20, 10)	16389.87	19051.77	-0.02739
Adam	0.1	16	(50, 20)	1380.089	1475.559	0.993837
Adam	0.01	4	(10, 5)	562.3163	670.6094	0.998727
Adam	0.01	4	(20, 10)	557.5731	666.441	0.998743
Adam	0.01	4	(50, 20)	676.0899	784.573	0.998258

Adam	0.01	8	(10, 5)	478.646	601.5698	0.998976
Adam	0.01	8	(20, 10)	782.1892	893.0053	0.997743
Adam	0.01	8	(50, 20)	514.1164	635.6179	0.998856
Adam	0.01	16	(10, 5)	369.9601	537.0515	0.999184
Adam	0.01	16	(20, 10)	537.3172	739.5315	0.998452
Adam	0.01	16	(50, 20)	572.4639	800.5877	0.998186
Adam	0.001	4	(10, 5)	**349.1438**	508.0707	0.999269
Adam	0.001	4	(20, 10)	464.5164	623.8052	0.998899
Adam	0.001	4	(50, 20)	386.8879	**506.5896**	0.999274
Adam	0.001	8	(10, 5)	517.4395	721.1957	0.998528
Adam	0.001	8	(20, 10)	701.5297	936.9139	0.997515
Adam	0.001	8	(50, 20)	422.1561	597.7368	0.998989
Adam	0.001	16	(10, 5)	445.5009	649.6394	0.998805
Adam	0.001	16	(20, 10)	420.8798	611.031	0.998943
Adam	0.001	16	(50, 20)	383.8728	552.0622	0.999137
Nadam	0.1	4	(10, 5)	699.5098	1055.949	0.996844
Nadam	0.1	4	(20, 10)	16391.39	19037.77	-0.02588
Nadam	0.1	4	(50, 20)	16753.27	19092.29	-0.03176
Nadam	0.1	8	(10, 5)	16426.96	18872.03	-0.00809
Nadam	0.1	8	(20, 10)	17270.67	20306.04	-0.16711
Nadam	0.1	8	(50, 20)	514.2261	698.3058	0.99862
Nadam	0.1	16	(10, 5)	474.4317	660.3259	0.998766
Nadam	0.1	16	(20, 10)	16665.9	18959.97	-0.01751
Nadam	0.1	16	(50, 20)	832.2318	956.5049	0.99741
Nadam	0.01	4	(10, 5)	509.1847	699.8159	0.998614
Nadam	0.01	4	(20, 10)	421.5006	542.498	0.999167
Nadam	0.01	4	(50, 20)	383.4185	519.1093	0.999237
Nadam	0.01	8	(10, 5)	372.047	530.0002	0.999205
Nadam	0.01	8	(20, 10)	375.7055	543.3756	0.999164
Nadam	0.01	8	(50, 20)	677.8564	832.8322	0.998037
Nadam	0.01	16	(10, 5)	663.8588	847.7803	0.997966
Nadam	0.01	16	(20, 10)	742.1761	955.3982	0.997416
Nadam	0.01	16	(50, 20)	954.2176	1251.772	0.995565
Nadam	0.001	4	(10, 5)	454.2809	658.7263	0.998772
Nadam	0.001	4	(20, 10)	454.5512	683.3953	0.998678
Nadam	0.001	4	(50, 20)	474.5549	685.251	0.998671
Nadam	0.001	8	(10, 5)	456.5307	682.5552	0.998681
Nadam	0.001	8	(20, 10)	449.8436	637.0944	0.998851
Nadam	0.001	8	(50, 20)	489.5487	715.2692	0.998552
Nadam	0.001	16	(10, 5)	466.3847	681.9122	0.998684
Nadam	0.001	16	(20, 10)	509.3542	695.4276	0.998631
Nadam	0.001	16	(50, 20)	444.7108	659.4987	0.998769

Table 4: Difference between actual and predicted prices, using live data.

Date	Predicted	Actual
12-01-23	68370.65	69800.58
12-04-23	69774.07	69908.27
12-05-23	69756.22	69134.61
12-06-23	69478.69	69287.64
12-07-23	69742.56	68720.96
12-08-23	69002.02	67667.22
12-11-23	68169.08	67796.69
21-11-23	68258.41	68204.11
22-11-23	68324.92	68689.75
23-11-23	68744.66	69340.5
24-11-23	69378.55	69662.79
27-11-23	69718.28	69366.89
28-11-23	69590.97	69574.16
29-11-23	69903.79	69303.49
30-11-23	69552.5	69647.14
12-12-23	69953.12	66496.06
13-12-23	67024.55	66563.75
14-12-23	67508.49	69154.73
18-12-23	69021.25	68201.23
19-12-23	67856.2	68708.56
20-12-23	69130.86	68718.28
21-12-23	68816.96	69195.95
22-12-23	69383.07	67966.89
27-12-23	68225.89	70273.03
28-12-23	70587.73	70076.25
29-12-23	69767.75	70494.8
01-02-24	70813.01	69337.93
01-03-24	69664.04	68412.11

Figure 5: The structure of LSTM model building.