Deep Learning based Enhanced Stock Market Trading and Patterns Oriented Improved Stock Selection

MSc Research Project
Data Analytics

Dipesh Khichada
Student ID: x22143271

School of Computing
National College of Ireland

Supervisor: Abid Yaqoob

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Dipesh Khichada
Student ID:	x22143271
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Abid Yaqoob
Submission Due Date:	14/12/2023
Project Title:	Deep Learning based Enhanced Stock Market Trading and Patterns Oriented Improved Stock Selection
Word Count:	7618
Page Count:	24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	11th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

(Meeting with Dipesh Laxmidas Khichada-20231214_064729-Meeting Recording.mp4)(video)

Deep Learning based Enhanced Stock Market Trading and Patterns Oriented Improved Stock Selection

Dipesh Khichada
x22143271

Abstract

This research employs sophisticated deep learning methodologies utilizing TensorFlow 2 and Keras to improve stock market trading and identify patterns. The main objective is to create strong machine learning models that can accurately detect intricate patterns and make predictions in the volatile stock market. We utilize technical indicators and analyze historical data to forecast future market trends. This research utilizes sophisticated LSTM models and explores various activation functions to determine the most precise and accurate model behavior. In addition, our approach incorporates real time data to enhance accuracy. We assess the performance of these models by employing metrics such as RMSE, MAE, and R2, showcasing the effectiveness of LSTM models, specifically when utilizing the TanH activation function, in predicting short-term stock market trends. This research enhances comprehension of stock market dynamics, providing valuable perspectives for investors and emphasizing the substantial capacity of deep learning infinancial evaluation.

Keywords: Deep Learning, Long Short-Term Memory, Price Prediction, Pattern Detection, Technical Indicators, Trading, Optimizing strategies.

1 Introduction

The stock market is a complex and volatile financial environment that poses a significant challenge for investors seeking accurate predictions of stock prices and returns. Machine learning, with its ability to analyze vast amounts of data and identify patterns, has emerged as a valuable tool for stock market forecasting. However, achieving high accuracy in predicting stock prices remains elusive due to the uncertainties and non-stationary nature of financial markets (Moghar, 2020). Stock market forecasting is not only crucial for investors aiming to optimize returns but also serves as a key indicator of economic trends, especially during periods of recession. Financial data's noisy and non-stationary nature has made accurate stock price prediction a formidable task for researchers and speculators alike (Moghar, 2020). In the pursuit of precision, feature selection becomes a vital component, with researchers exploring various indicators and data sources to enhance accuracy (Nabi pour, 2020). To assist investors in navigating the stock market, various strategies have been developed over the years. One widely used approach is technical analysis, also known as Chartist Analysis. This method involves analysing historical stock prices to calculate indicators, which are then plotted on charts alongside stock prices. Investors examine these charts to identify patterns that may offer insights into future market movements, helping inform their decisions (C. D. K. and J. Dahlquist, 2010).

Technical Analysis employs a range of indicators, each offering different information about the market. For instance, moving averages can reveal market trends, while indicators like momentum and relative strength provide insights into whether a particular stockis potentially overbought or oversold (C. D. K. and J. Dahlquist, Te,2010). In simpler terms, these tools help investors make sense of stock price movements and make more informed decisions.

Despite numerous attempts, the accuracy of existing models hovers with an average RMSE of 17, highlighting the need for improvement in data processing, feature selection, and overall prediction methodologies (Joiner, 2022). To address this, the current research proposes a novel approach, integrating live data fetched from APIs for real-time prediction. This departure from traditional data sources aims to enhance accuracy by incorporating live data which is till today's date market information. Developing an effective stock price prediction model is crucial for gaining insights into market situations. Numerous studies have explored the correlation between various variables and stock price behavior, but achieving consistent accuracy remains a challenge. One potential factor contributing to suboptimal outcomes is the process of selecting variables. This research focuses on utilizing a computational framework to predict stock index prices, employingthe (LSTM) model an improved neural network architecture designed for time series data. Our approach involves selecting features carefully from fundamental, macroeconomic, and technical data to construct the model. Subsequently, we normalize the collected data using the min-max normalization technique. The LSTM model's input sequence is then created with a specific time step.

Key hyperparameters, including the number of neurons, epochs, learning rate, batch. size, and time step, are incorporated into the model. To address overfitting issues, regularization techniques are employed. Once the hyperparameters are fine-tuned, the input data is fed into the LSTM model to predict the closing price of the stock market index. The model's quality is evaluated using RMSE, MAPE, and MAE

Our contribution lies in creating a sophisticated model that maintains a well-balanced set of variables without adding unnecessary complexity to the model architecture. This approach aims to capture the multi-dimensional behavior of the stock market effectively.

1.1 Research Problem/Question.

Q. How do different activation functions impact the accuracy and efficiency of LSTM based models in forecasting stock market trends? A. To answer the above question wehave explored a few key areas which include.

1. Comparison of Activation Functions: Investigate how different activation functions (such as ReLU, Swish, Tanh, etc.) affect how well LSTM networks predict the stock market.

2. Measure Accuracy: Check how well the LSTM models can predict with each activation function. This would involve comparing metrics like MAE, R2, RMSE etc that are useful for time series forecasting.

3. Real World Market Data Use: Test the usefulness of the various LSTM models by applying them to real world stock market data.

The results of this Research could facilitate accurate and efficient forecasting of financial time series.

1.2 Research Contribution

1. Identifying Stock Market Trends: The primary objective is to explore how deep learning models, specifically (LSTM) networks, can be leveraged to identify stock market trends and extract valuable insights or patterns.

2. Impact of Activation Function: Investigating how the choice of activation function influences the performance of deep learning models, particularly LSTM, is crucialfor refining prediction accuracy.

3. Role of Technical Indicators: Assessing the contribution of technical indicators in boosting and generating more reliable results in stock price prediction.

1.3 Research Methodology

The research methodology involves experimenting with different combinations of technical indicators, macroeconomic levels, and various features to detect complex patterns and optimize results. The integration of deep learning with algorithmic trading strategies will be explored, emphasizing real-time execution through streaming data processing techniques. The research will employ LSTM models, focusing on their design, implementation, and integration with technical indicators for enhanced accuracy.

2 Related Work

2.1 LSTM Models and Activation Functions in Stock Prediction

The authors in (Pang et al,2020) introduced an innovative approach aimed at enhancing stock market predictions using neural networks. Their methodology involved collecting real-time and offline data from livestock markets and developing a digital internet platform for comprehensive stock analysis, monitoring, visualization, and result analysis. Introducing the" stock vector theory," based on word vector research techniques, the researchers departed from relying on specific indices and instead utilized historical data from various inventories. Comparing their LSTM- based approach with traditional neural network algorithms commonly employed for stock market predictions, they achieved a maximum accuracy of 57.2, highlightingthe superior performance of the LSTM-NN with an integrated layer over conventional techniques. They emphasized the potential for further accuracy improvement through model enhancement, data cleaning, feature refinement, and the exploration of various deep learning approaches.

The research conducted by (Hasan M Sami et al,2023). investigates the effectiveness of various activation functions in LSTM networks to predict. The research analyses 25 stocks from various exchanges and compares the performance of activation functions such as ReLU, ELU, and TanH. The research determines that the Hyperbolic Tangent TanH function, which achieves an accuracy of 80 % in predicting multiple variables, is the most efficient for LSTM based stock price prediction. This research emphasizes the importance of choosing suitable activation functions in LSTM models to improve the accuracy of financial forecasting.

The work by (Agrawal, M., Khan, A.U. and Shukla, P.K., 2019.) introduces a model that utilizes Optimal (O-LSTM) and Stock Technical Indicators (STIs) to forecast stock prices and trends. By incorporating a Correlation tensor to improve accuracy,this method surpasses conventional machine learning classifiers, achieving an average prediction accuracy of 59.25 % across multiple stocks. Significant contributions consist of the creation of the O-LSTM model, the introduction of the correlation tensor concept, and the execution of comparative performance evaluations. This research provides valuable assistance to investors in making well informed decisions, offering valuable insights for both short-term and long-term investment strategies.

The work done by (Ojo, Owolawi, Mphahlele, and Adisa,2019.) investigates the application of stacked (LSTM) networks in forecasting stock market trends. The research utilizes the time series data handling capabilities of LSTM to analyze the NASDAQ Composite. The outcomes are remarkable: the stacked LSTM model attained a 53.6 % accuracy in predicting fluctuations in the stock market. The assessment was conducted using the MSE and MAD metrics, resulting in values of 0.0022 and 0.0495, respectively. These findings highlight the capacity of deep learning to forecast intricate market behaviors, even in the face of challenges posed by market volatility.

2.2 Comparative Studies of LSTM with SOTA Models

The work by (Hum Nath Bhandari et al,2022). uses LSTM neural networks to predict stock market indices. He literature review emphasizes LSTM's superiority over traditional models in machine learning. The authors describe their LSTM model, which handles long-term financial data dependencies. They use SP 500 fundamental, macroeconomic, and technical indicators. The paper compares single-layerand multi-layer LSTM models using RMSE, MAPE, and Correlation Coefficient metrics. The single-layer LSTM model outperforms the multilayer model in prediction accuracy.

The researchers in (Kobiela, Krefta, Kr´ol, and Weichbroth,2022) provides a thorough analysis and comparison of ARIMA and LSTM models in their ability to forecast NASDAQ stock prices. The text highlights the difficulty in achieving precise financial predictions and employs the evaluation metrics of (MSE) and (MAPE). The research concludes that, overall, ARIMA performs better than LSTM, especially when it comes to making predictions for longer time periods. However, LSTM does have a slight advantage when it comes to forecasting for one day. This indicates that ARIMA demonstrates higher dependability despite having a limited set of features, such as historical prices. The paper concludes that ARIMA is a more effective method for analyzing NASDAQ data, offering valuable insights for selecting suitable forecasting models.

2.3 Deep Learning and LSTM in Diverse Market Analysis

The authors in (Nitin Sharma and Biju R Mohan,2022) explores the forecasting of stock prices within the automotive industry, with a specific focus on Tata Motors. The research employs ARIMA, LSTM, and ANN models for analysis. The research's distinctive methodology involves analyzing both the historical data of Tata Motors and its correlation with other companies in the same industry. The findings are enlightening: the LSTM model attained a MAE of 14.25 and RMSE of 22.23, whereas the ARIMA model exhibited an MAE of 24.62 and RMSE of 28.28, indicating consistent performance. Nevertheless, the Artificial Neural Network ANN model, which integrated data from other automotive companies, demonstrated superior performance with a MAE of 10.39 and RMSE of 16.47. The research concludes that the ANN model, which considers the interconnectedness within the automotive industry, provides the most efficient and precise forecast for Tata Motors' stock price. This research emphasizes the significance of considering the overall dynamics of the sector when conducting financial forecasting.

The work by (Damrongsakmethee and Neagoe,2020) forecasts stock prices using a Deep LSTM network. This deep learning research uses eleven LSTM layers and ADAM optimization to analyze a decade of Yahoo Finance stock data for the Dow Jones Industrial and S&P 500 indices. Inaccuracy and error metrics, the model excels. For the Dow Jones Industrial dataset, the Deep LSTM model had an RMSE of 9.2103, MSE of 8.4830, MAPE of 0.1637, and accuracy of 83.62 %. Even better was the S & P 500 Index dataset, where the model had an RMSE of 8.4165, MSEof 7.1673, MAPE of 0.1413, and accuracy of 85.86 %. The model's effective stock market price prediction shows Deep LSTM networks' potential in financial market forecasting. The paper shows deep learning's stock market prediction capabilities, advancing financial technology.

In (S. Banik, N. Sharma, M. Mangla, et al.2022), the authors highlighted the effectiveness of LSTM models in predicting stock market trends. The research primarily focuses on the model's capability to forecast various stock market parameters, emphasizing its accuracy in predicting prices and volumes. The performance of the LSTM model was rigorously evaluated against actual market values, using metrics such as RMSE, MAE, and MAPE. Key findings include RMSE values for different stock features, notably 7.90 for opening prices, 4.13 for closing prices, 8.89 for thehighest price, 8.72 for the lowest price, 16.04 for stock volume traded, and 50.39 for the closing price of NIFTY Industry Average. These results underline the LSTM model's potential as a robust tool for swing traders, offering significant insights forinformed decision making in the stock market.

The researchers in (Ali, M. et al., 2023.) forecasts the KSE 100 index using a novel hybrid method. This method uses an LSTM network and a new EMD based on Akima spline interpolation. Traditional stock market forecasting methods due to complexity, nonlinearity, and volatility creates LSTM networks from noisy stock data using intrinsic mode functions (IMFs) and a monotone residue. Long term data sequences suit the enhanced RNN LSTM network. The model's efficiency was assessed using KSE 100 index data from January 1, 2015, to August 25, 2022. Data was divided into 90 % for model training and 10 % for testing. RMSE,MAE, and MAPE values were lower for the hybrid Akima EMD LSTM model, indicating greater accuracy. The model can predict complex financial time series data, including stock market non stationarity and nonlinearity.

The research by (Zaheer, S.,2023) investigates the application of deep learning in predicting stock prices. The research examines the performance of CNN, LSTM, and a single-layer RNN model using Shanghai Composite Index data, demonstratingtheir superiority over traditional methods. The key findings demonstrate the superior performance of the single-layer RNN, characterized by significant enhancementsin both forecasting precision and computational efficiency. The model demonstrated substantial decreases in MAE and RMSE, as well as an R2 value approaching 1, indicating a high level of predictive precision. The research specifically emphasizes the model's improved accuracy in predicting two parameters (close and high price), demonstrating its potential in forecasting complex financial time series.

The research by (A. Chatterjee, H. Bhowmick and J. Sen,2022) assesses the volatility of stocks in the banking, IT, and pharma sectors of the NSE by employing GARCH models and LSTM. The LSTM model consistently achieved higher accuracy than the GARCH variants. The key findings indicate that the LSTM model outperforms other models in terms of RMSE values in various sectors, specifically 0.0147 in banking, 0.0125 in IT, and 0.0115 in pharma. EGARCH demonstrated superior effectiveness in the banking sector (RMSE 9.9195) and the pharmaceutical industry (4.5245) among the various GARCH models. On the other hand, GJR-GARCH performed exceptionally well in the IT sector (5.4157). The findingsemphasize the potential of LSTM in predicting stock volatility, surpassing traditional models across different market sectors.

Summary: The existing works employ ANN and LSTM models in various combinations to predict stock market trends. Certain research projects, like those by (Pang et al. 2020) and (Hasan M Sami et al. 2023), focused on adaptive choices of, different LSTM activation functions, suchas TanH and ReLU. These works highlight how advanced deep learning methods, such as multilayer LSTM and hybrid models with features like Empirical Mode Decomposition (EMD), have changed the way financial forecasting is done. In the stock market, they stress how important it is to choose the right models and methods for different information and goals. LSTM works well, especially with complicated patterns and long-term dependencies, which shows that it could be useful in financial forecasts.These works show that predictive analytics in finance is dynamic and always changing. This research uniquely integrates LSTM models with various activation functions differentiating it in its approach to improve prediction accuracy

3 Methodology

The research utilizes various Python libraries for data processing, visualization, and machine learning tasks. Libraries such as numpy and pandas are instrumental for numerical computations and handling data, respectively. For data visualization, the code employs matplotlib.pyplot and seaborn. Date and time data are managed using datetime

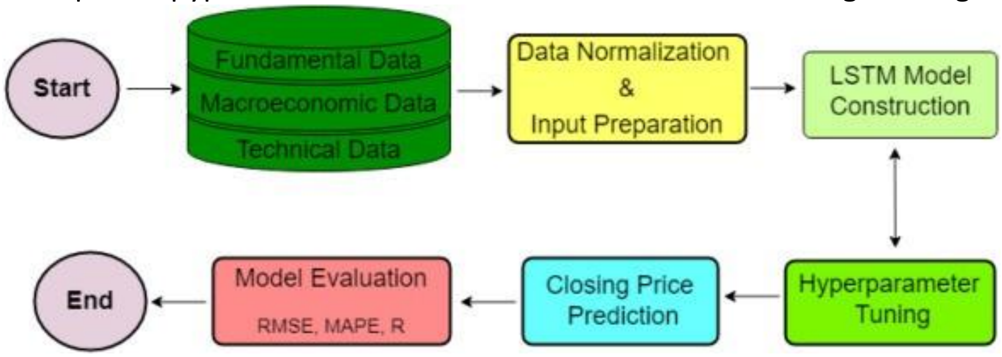

Figure 1: Schematic Diagram of the Proposed Research Framework Humnath Bhandari "Predicting stock market index using LSTM",Machine Learning with Applications,Volume 9,2022.

and date. The proposed model fetches financial data from Yahoo Finance through pandasdatareader and yfinance.The MinMaxScaler from scikit-learn normalizes data values. The machine learning aspect is handled by Keras, where models are built using Sequen tial, with layers like Dense, LSTM, and Dropout for configuring neural networks. The script specifically retrieves Microsoft's stock data, focusing on the adjusted close prices, which are then renamed to 'Price' for clarity. The initial rows of this data are displayed, showcasing the start of the dataset.

3.1 Dataset

3.1.1 Dataset Selection

I This study utilized stock return data from the Nasdaq market, with a specific focus on Microsoft, in the context of a deep neural network. The dataset, obtained from the Yahoo Finance API, covers the time from January 3, 2001, to December 8, 2023, and consists of 6021 data entries. To construct and validate the model, the dataset was divided, with 80 % (about 4817 rows) allocated for training and 20 % (roughly 1204 rows) for testing. This methodology enables the neural network to acquire knowledge from a significant chunk of the data and thereafter undergo evaluation based on its capacity to forecast stock returns on unfamiliar data. This ensures a reliable assessment of its performance under real world circumstances.

Variable	Description
Date	Unique ID
Open	The stock's opening price for that day
High	The stock's highest price for that day
Low	The stock's lowest price for that day
Close	The closing price for that day
Adjusted Close	Adjusted closing price after split and dividend distributions
Volume	Number of trades happening in that day

Table 1: Description of Stock Market Dataset Variables

3.1.2 Data Preprocessing

The Following Data had no Null values, but we have scaled the data between [0,1] using MinMaxScaler from the machine learning library, often used for normalization in data preprocessing. The MinMaxScaler transforms each feature to a given range, typically between 0 and 1. This scaling makes the neural network model more efficient and stable.The output data training scaled is the scaled version of the training data. Scaling the data is a critical step in preparing it for use in a deep learning model, ensuring thatthe input features contribute equally to the model’s learning process

3.2 Model Building

We built an LSTM model, a popular deep learning technique in RNN for time series prediction. for the stock market for time series prediction using RNN, LSTM is a prominent deep learning technique. That addresses both classification and regression problems. Ordinary RNN struggles to learn long-term dependencies due to the vanishing gradient problem, despite being better at memory than conventional networks (Hochreiter, 1998). The vanishing gradient problem is solved by LSTM using. memory cells. According to

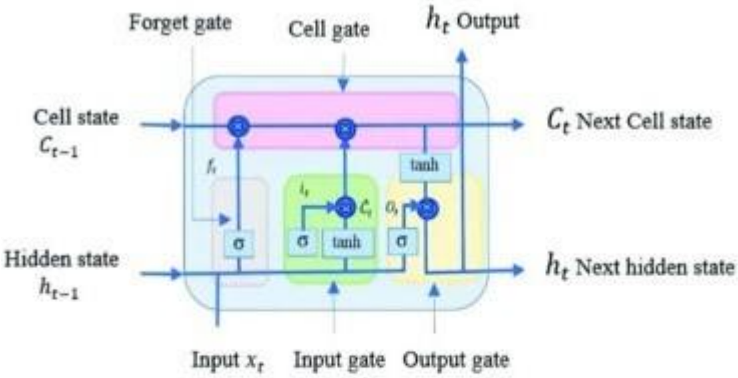

Figure 2: LSTM Architecture

Gers et al. (2000), Gers et al. (2003), Hochreiter, and Schmidhuber 1997), it has an input layer, a hidden layer, a cell state, and an output layer. The cell state, which moves through the chain with only linear interaction, is the key component of the LSTM architecture because it preserves the integrity of information flow. The gate mechanism ofthe LSTM modifies or deletes information regarding the cell state. The sigmoid layer, hyperbolic tangent layer, and point wise multiplication operation make up this method of selective information transmission.: Represents matrix point-by-point multiplication; represents matrix addition; At time t, a forget gate(ft), an input gate(it), an output

$$ft = \sigma\ (Wf\ [ht\text{-}1, xt] + bf)$$
$$it = \sigma\ (W_1 \cdot [ht\text{-}1,Xt] + b_1)$$
$$C = tanh(Wc\ [ht\_1,2t] + bc)$$
$$Ct = ft * Ct\text{-}1 + it * \check{C}t$$
$$oto\ (Wo\ [ht\text{-}1, xt] + bo)$$
$$ht = Ot * tanh\ (Ct)$$

gate (Ot), an input candidate(gt), a cell state (Ct),a hidden state (ht), Sigmoid function (*σ*), *tanhfunction*

3.2.1 Activation Functions in LSTM

An activation function defines the range of activation values for an artificial neuron. This is applicable to the total of the weighted input data of the neuron. An activation function is defined by its non-linear feature. In the absence of an activation function, the computation of the output in a multilayer perceptron would include only the linear multiplication of the weights and input values. To make the neural network not follow a straight line, we need to use activation functions like ReLu, Swish, and tanh. So, the network can make models of connections and patterns in the data that are more complicated

1. ReLu: In the past few years, the Rectified Linear Unit, or ReLU, has gained a lotof attention. When the input value is bigger than zero, the activation is linear: In other words, R(x) = max (0, x) Figure 3 show's how this function. There are pros and cons to employing it:
   (+) It accelerates gradient descent toward the loss function's global minimum compared to other activation functions. Due to its linear, non-saturating nature.
   (+) It can be implemented by thresholding a value vector at zero, unlike tanh and sigmoid, which require computationally expensive operations like exponents.
   (-) Unfortunately, the activation function is similarly flawed. Since this function outputs zero for input values below zero, network neurons might become brittle and even "die" during training.

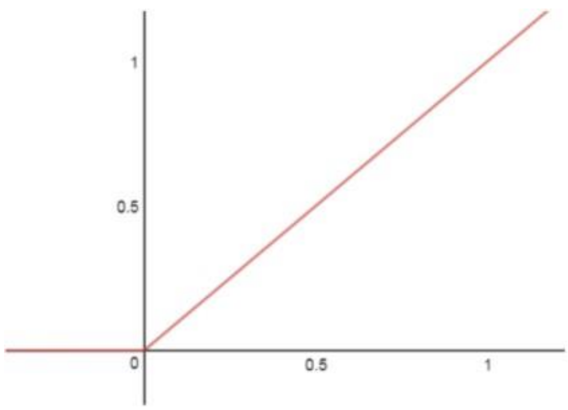

Figure 3: Relu Activation Graph (Artem Opperman,2021)

2. Tanh: The Tanh function is another popular Deep Learning activation function. Figure: Tangent hyperbolic function Figure 4 shows the function Graph
   (+) Most of the times Tanh function is usually used in hidden layers of a neuralnetwork because its values lie between -1 to 1 that's why the mean for the hidden layer comes out be 0 or its very close to 0, hence tanh functions help in centring the data by bringing mean close to 0 which makes learning for the next layer much easier. So, tanh function is useful.
   (-) Like the sigmoid function, Tanh has the vanishing gradient problem.

3. Swish: One of the new activation functions that was first suggested in 2017 is called Swish. It uses both exhaustive and reinforcement learning based search. The Swish Activation Function works all the time. The shape of the Swish Activation Function

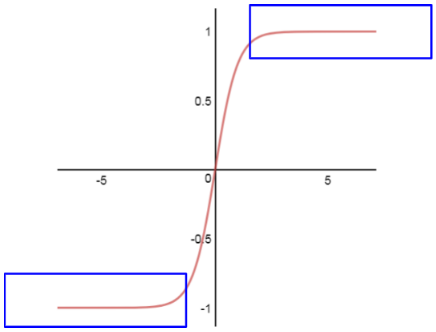

Figure 4: TanH Activation Graph (Artem Opperman,2021)

is like the shape of the ReLU function because it is both unlimited above and below. 0. On the other hand, it is not monotonic like ReLU; it can be differentiable everywhere. It looks a lot like ReLU because when x gets big enough, f(x) approaches1, which means that swish activation function values approach x. Also, when the negative number is big enough, f(x) is close to 0, which means that the values of the swish function are also close to 0. (+) It is always differentiable and remains ongoing. Above figure 5 shows the function graph.
(+) It's basic and simple to use.
(+) Unlike ReLU, it doesn't have the problem of neurons dying.

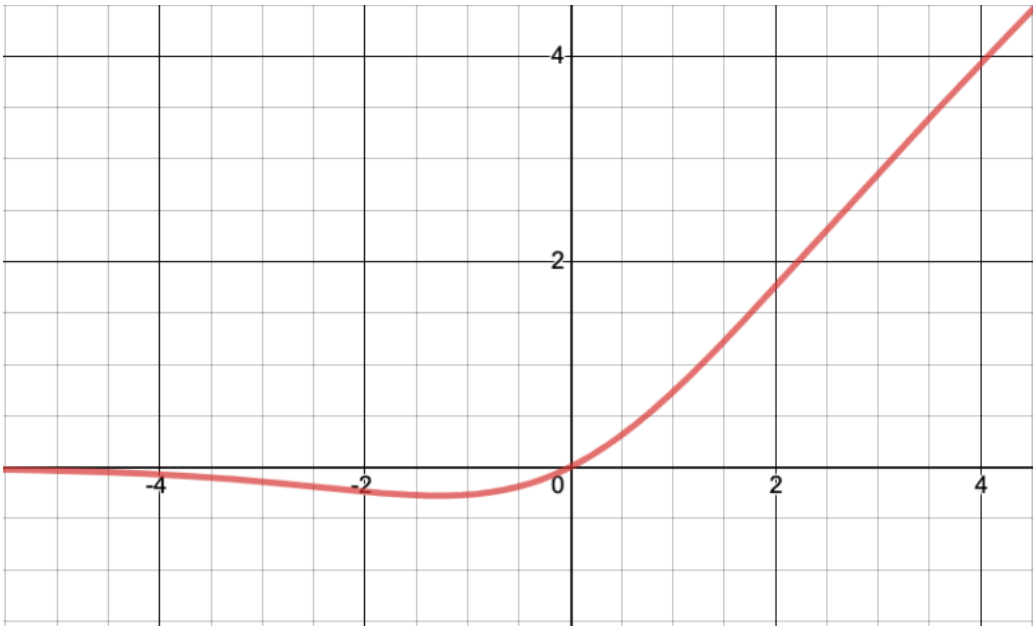

Figure 5: Swish Activation Function

3.2.2 Combinational Hyperparameter’s in LSTM

An RNN architecture called Long Short-Term Memory (LSTM) can remember patterns and relationships in sequential data that occur over a longer period. Several hyperparameters of LSTMs are essential in determining the model’s behavior and performance.The following are descriptions of several important hyperparameters. LSTM Units/Neurons Count: The model’s ability to grasp intricate patterns inthe data is dictated by the quantity of LSTM units or neurons in a layer. The model can learn more complex features with an increase in the number of units, but it runs the risk of overfitting, particularly when data is scarce.

1. Layer Count: - Explanation: A deep LSTM network is formed by stacking LSTMsin several layers. It’s possible that more training data and computing resources willbe needed to train deeper networks, but they are better able to capture hierarchical characteristics in the data. Issues with vanishing or ballooning gradients may also arise in deep architectures.

2. Dropout: When training, the regularization approach known as” dropout” randomly assigns zero to a fraction of the input units. - Impact: To avoid overfitting, dropout lessens the model’s dependence on individual neurons during training, lead- ing to a stronger final product.

3. Batch Size: -: The number of samples handled in one iteration is represented bybatch size. The training process can be made more stable with a larger batch size,but it does come with a memory cost. While processing smaller batches may bemore efficient, it may also add more noise.

4. Learning Rate: -: The optimization step size is determined. by the learning rate. - Impact: While a higher learning rate could potentially speed up convergence, it also raises the possibility of exceeding the minimum. While training may take more time with a slower learning rate, more stability is possible.

5. Epochs: - A single epoch is a full traversal of the whole training dataset. Impact: The amount of training epochs dictates the frequency with which the model encounters the complete dataset. Underfitting can happen if there aren’t enough epochs,while overfitting can happen if there are too many.

3.3 Technical Indicators

In this research for Technical Analysis, we have used Stock stats which is a Python tool for working with time series data that is used in finance, especially when analysing the stock market. It's like a wrapper for Panda's data frames, adding extra features and making them easier to use for technical analysis and algorithmic trade.

1. Simple Moving Average: We will begin by utilizing a straightforward indicator known as the Simple Moving Averages (SMA). The underlying principle is quite uncomplicated: we calculate the average of the closing prices over a specific time frame to smooth out the trend and illustrate the overall direction of the trend flow. We own two SMA one with a duration of 20 days and the other with a duration of50 days.
   Strategy: We have developed a straightforward program that executes a purchase when the Short Moving Average (MA) exceeds the Long MA and sells when the Long MA surpasses the Short MA. Initially, we created two lines named signalBuy and signalSell. Upon the fulfillment of the Buy condition, we appended the Close Price to the signalBuy. Alternatively, we included Null. Consequently, we included these lines in our dataset. We have included a variable named "position" to ensure that we execute the reverse transaction following the prior trade. Thus, if the prior trade was a "buy", the subsequent deal will only be a "sell" as the position is set to true.

2. Moving Average Convergence Divergence indicator: A short-term and long-term exponential moving average EMA are used to calculate (MACD). The signal line, an EMA of MACD, indicates momentum direction. EMA emphasizes recent data items. We'll compute MACD using pandas-ta. MACD, the difference between the two EMAs, is the first of three columns. The second column is MACD EMA, called Signal. The MACD histogram (third column) compares MACD and Signal
   Strategy: If the MACD line surpasses the signal line, which represents its moving average, we will adopt a long position. Shorting a stock is advisable when the MACD line crosses below the signal line, as this indicates a reversal in the trend. Our strategy incorporates risk as well. We will sell if the price above the specified limitations or %. Stop loss and trailing stop loss are integral components of our strategy. The level of risk tolerance is 2.5%. If the purchase price of an item is 100 units and the price surpasses 97.5 units (100 multiplied by the difference of 1 and 0.025), we shall proceed to sell it. Assuming we purchase an asset at a price of 100 and its value subsequently increases to 110, our trailing stop loss will be set at 107.25. This is calculated by multiplying 110 by (1 minus 0.025).

3. Bollinger Bands: Due to its formidable efficacy and straightforwardness, Bollinger bands have gained significant popularity as a trading indicator. The Bollinger bands consist of three lines: the upper band, the middle band, and the bottom band. The lower and upper Bollinger bands are positioned at a distance of two standard deviations from the mean average of the Close Price. The two bands constitute more than 80% of the price movement, rendering any price above or below them noteworthy.
   Strategy: While it is possible to utilize it as a signal, it is not advisable to only depend on it as the movement of the stock could be attributed to other factors. Our method is predicated on the notion that there is a high probability of the price reverting to its previous level after deviating beyond the upper or lower Bollinger bands.

4 Implementation

4.1 Implementation Settings

The required system parameters for running on Google Colab consist of Python 3, a Google Compute Engine backend, 12.7 GB of total RAM with 0.8 GB now in use, and107.7 GB of disk space with 26.9 GB being utilized. This configuration is highly compatible with LSTM models and datasets of medium size.

4.2 LSTM Model Configurations

After the Execution of Code, it was Found out that the LSTM model with
1. First LSTM Layer: With 90 units and 'tanh' activation, it returns sequences. X_train is the input time series data processed by this layer.
2. First Dropout Layer: Removes 20 units.
3. Second and Third LSTM Layers: 90 and 80 units, 'tanh' activation, and return sequences. Layers give the model depth to learn more complex patterns.
4. Drop 30% and 20% of features in the Second and Third Dropout Layers for regularization.
5. Fourth LSTM Layer: 120 units prepare the model for final output by not returningsequences.
6. Fourth Dropout Layer: Removes 20% of features.
7. Dense Layer: A fully connected layer that outputs the final prediction, with one unit representing a stock price.

4.3 Comparative Methods FacebookProphet Prediction Method

We used FbProphet as well for the same Data FB Prophet is a forecasting tool developed by Facebook for time series data. It's particularly effective for datasets with strong seasonal patterns and several seasons of historical data It is a Predefined Library wherewe have to just call few functions Time series cross-validation is performed using the 'cross validation 'function on a Prophet model 'm'. Its parameters:
- initial='8500 days'': Initial training duration.
- period='90 days'': Sets cross-validation cutoff date spacing.
- horizon='90 days'': Sets prediction horizon for each cutoff.

The 'df_cv' dataframe contains forecast and actual values for each cross-validation simulated forecast date. Use 'df cv.head()' to view the initial rows of the dataframe. This helps evaluate Prophet model performance over time.of 90,60,30 Days

4.4 Model Evaluation Metrics

When assessing a deep learning project, such as stock market prediction using LSTM networks, three frequently employed metrics are Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R2). Every metric offers distinct perspectives on the performance of the model.

4.4.1 Root Mean Square Error (RMSE):is calculated by taking the square rootof the average of the squared differences between the predicted and actual values.It measures the standard deviation of the residuals (prediction errors). Within the realm of stock market prediction, a reduced RMSE value signifies that the LSTM model's forecasts align more closely with the real stock prices, indicating a greater level of accuracy in theprediction process. RMSE is highly responsive to outliers, rendering it valuable in financial markets where abrupt price fluctuations can transpire.

RMSE = SQRT (1/N * SUM ((y i - y^ i) 2̂))

Note: y i with actual values, y^ i with predicted values, and N with the number of observations.

4.4.2 Mean Absolute Error (MAE): MAE quantifies the average absolute value of errorsin each set of predictions, regardless of their direction. The average of the absolute differences between predicted and actual values is calculated over the test sample.

the MAE serves as a clear and precise metric to assess the accuracy of predictions in stock market forecasting. A lower MAE signifies that the LSTM model is producing predictions that are, on average, in closer proximityto the actual values. Compared to RMSE, it exhibits lower sensitivity to outliers, providing a more resilient assessment of model performance.

MAE = 1/N * SUM (ABS (y i - y^ i))

4.4.3 R2: Coefficient of Determination The term R2 refers to the proportion of the variance in the dependent variable that can be predicted by the independent variables.The scale ranges from 0 to 1, with a value of 1 indicating flawless prediction. A high R2 value, close to 1, indicates that the LSTM model effectively accounts for a significant amount of the variability in stock prices. This is essential for predicting stock market trends as it demonstrates the model's ability to accurately represent the fundamental patterns and dynamics of the market.

R2̂ = 1 - (SUM ((y i - y^ i) 2̂) / SUM ((y i - y bar) 2̂))

Note: y bar with the meaning of the actual values.

RMSE and MAE offer insights into the accuracy and magnitude of errors in predictions, while R2 assesses the overall quality of the model's fit to the data.

5. Evaluation and Result Analysis

In this Research LSTM with Tanh as activation Function this Model was proved to be the best for 90,60,30 Days Forecasting with the Batch Size was kept constant which was 64 which means that model will process 64 sample at a time for all the models. With different Activation Function and there were 100 Epochs which means that model will go through data 100 times with 64 samples each time Training encompasses the process of propagating data forward through the network, computing the loss (the discrepancy between predictionsand actual values), and performing backpropagation to optimize the network's weights for improved accuracy.

5.1 Results and Analysis

5.1.1 Results for 90 Days Forecasting

RMSE, MAE, and R2 values show that the TanH activation function model outperformed other stock market forecasting models. TanH had the lowest RMSE of 4.51370, MAE of 3.650, and R2 score of 0.94827, for 90 Days Prediction indicating superior predictive accuracy and reliability. TanH's effectiveness in this context is shown by the model's abilityto closely match actual values and its consistency across the Table 2.

Model	RMSE	MAE	R2
LSTM ReLu	10.0663	8.853	0.74288
LSTM TanH	4.51370	3.650	0.94827
LSTM Swish	10.28436	8.747	0.73147
LSTM with No Activation Function	4.97797	4.180	0.93708
FB Prophet	11.84743	5.8809	0.93558

Table 2: Model Performance Metrics for 90 Days

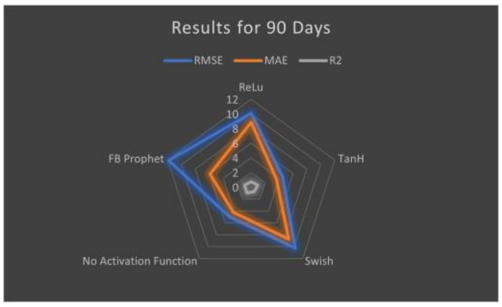

Figure 6: Radar Chart for the Evaluation metrics

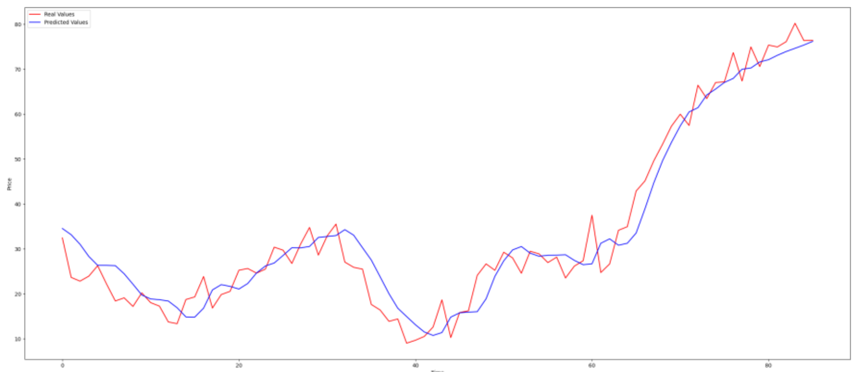

Figure 8: Actual vs Predicted Values Graph For 90 Days Prediction

5.1.2 Results for 60 Days Forecasting

Similarly, TanH was Proven to be Best for 60 Days Prediction with lowest RMSE of 5.4895, MAE of 4.545 and R2 of 0.94404 which can be Checked in Table 3

Model	RMSE	MAE	R2
LSTM ReLu	12.1714	10.322	0.70720
LSTM TanH	5.48945	4.545	0.94044
LSTM Swish	10.6689	8.960	0.77503
LSTM with No Activation Function	6.60651	5.4936	0.91373
FB Prophet	8.88867	6.7024	0.91623

Table 3: Model Performance Metrics for 60 Days Forecasting

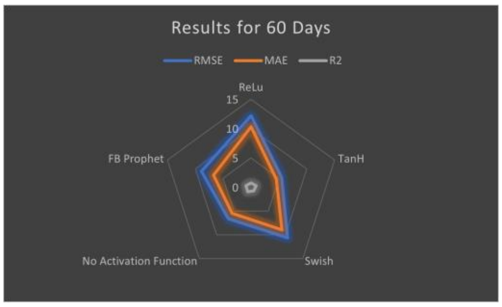

Figure 9 : Radar Chart for 60 Days Evalution Metrics

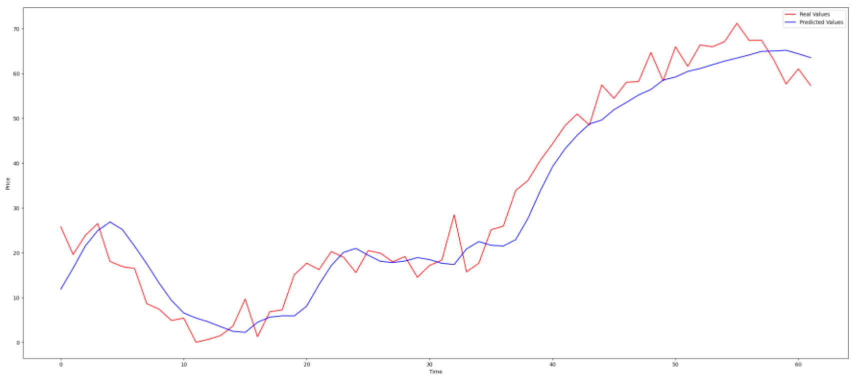

Figure 10: Actual vs Predicted Values Graph For 60 Days Prediction

5.1.3 Results for 30 Days Forecasting

Similarly for 30 Days Forecast the TanH activation function model. It has the lowest RMSE (4.7140) and MAE (3.798), indicating high prediction accuracy. Moreover, its R2 score of 0.93384 indicates a strong correlation between predicted and actual values which can be Checked from Table 4

Model	RMSE	MAE	R2
LSTM ReLu	12.6872	10.434	0.52083
LSTM TanH	4.7140	3.798	0.93384
LSTM Swish	12.4168	10.245	0.54104
LSTM with No Activation Function	5.62393	4.7412	0.68330
FB Prophet	14.5749	15.745	0.875

Table 4: Model Performance Metrics for 30 Days Forecasting

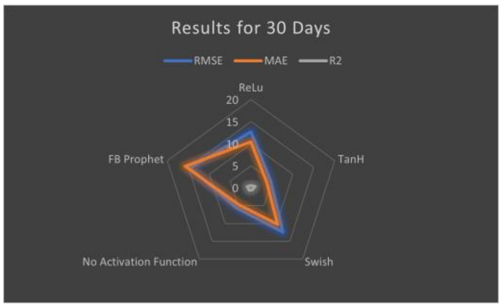

Figure 11: Radar Chart for the Evaluation metrics

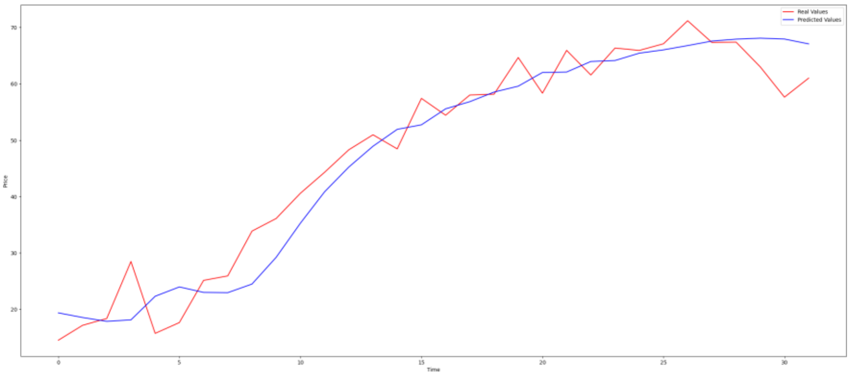

Figure 12: Actual vs Predicted Values Graph For 30 Days Prediction

5.2 Analysis of Technical Indicator

Technical analysis focuses on analyzing the price movement of a stock or ticker to find patterns, Technical Analysis is mostly based on quantitative data, particularly the priceand volume. While this approach is effective for short-term predictions, long-term growth of your money is ultimately determined by the success of the underlying business. ForMy research we have Used Pandas Technical Analysis (Pandas TA) is a user friendly library that utilizes the Pandas package. It offers a wide range of features, including over130 indicators, utility functions, and more than 60 candlestick patterns. The Data we have used if fetched from yahoo finance API the analysis for SMA is done only for last5 months which is 150 days as I need average for 50 days for analysis we have done this analysis to make sure that when the Investor Picks a stock he can Check the Forecastsfrom Two Methods one is LSTM which will Forecast the Stock Prices and other is by Checking this Technical Indicators which will help him to be double Sure before pickingup the right Stock.

5.2.1 Analysis on Simple Moving Average

The Red Line represents the average of last 20 days while the yellow line represents the average of past 50 days the strategy that we have build is whenever the red line (Moving average 20 days) crosses the yellow line(Moving Average 50 Days) and if it goes up it means that in future the stock price is going to go up and if the red line(moving average 20 days) crosses yellow line(moving average 50 days) and goes down it means that the price of stock is going to go down and someone who holds the stock should sell it to refrain from losses. Figure 12 above shows the red and green the red is to indicate when to sell stock and green one is to indicate when to buy a particular stock.

Figure 13: Analysis of Stock based on Simple Moving Average

5.2.2 Analysis on Moving Average Convergence Divergence

The Moving Average Convergence Divergence indicator (MACD) uses short-term and long-term exponential moving averages (EMA). We'll calculate MACD with pandas-ta, which will provide us three columns: MACD, Signal, and MACD histogram. If the MACD line intersects the Signal line (which is the moving average of the MACD line),we will initiate a long position. Conversely, when the MACD line intersects below the signal line, it is more prudent to initiate a short position on the stock as it indicates an expected reversal in the trend. Our strategy includes risk. We'll sell if the price exceeds the acceptable limits or percentage. Our strategy now includes a STOP LOSS and a Trailing Stop Loss. Figure 13 shows the working of MACD the green marker show indicates that there will be growth in price of Stock and Red indicate the fall in price of that stock.

Figure 14: MACD IMPLEMENTATION

5.2.3 Analysis on Bollinger Bands

The upper Bollinger band, middle Bollinger band, and lower Bollinger band. The upper and lower Bollinger bands are graphed at a distance of two standard deviations from the mean of the Close Price. We invoked the 'bbbands' function from the pandas_ta library and subsequently merged the resulting data frame with our original data. The strategy logic has been developed to trigger a Buy signal when the Close Price reaches the lower band (BBL_length standard-deviation), indicating an oversold condition. Conversely,a Sell signal is generated when the Close Price reaches the upper band (BBU_length standard-deviation), indicating an overbought condition. Figure 14 is graph for Bollinger band analysis.

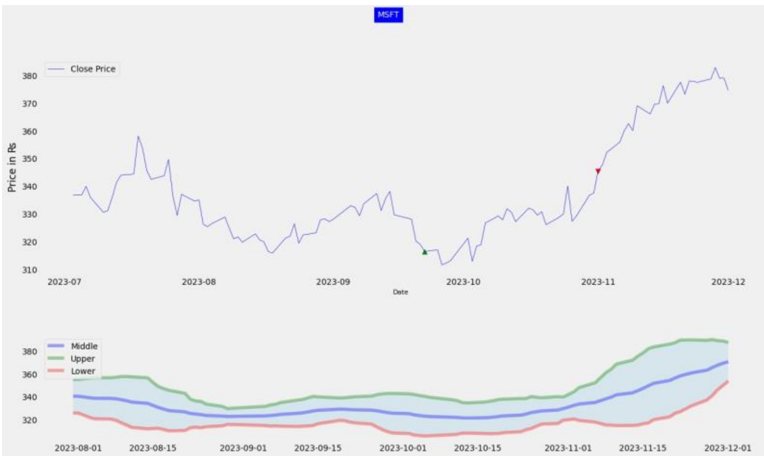

Figure 15: Bollinger Bands Graph Analysis

6 Conclusion and Future Works

The Project focuses on the effectiveness of deep learning techniques, especially LSTM models, in stock market prediction. The main thing that I want to highlight is the significance of selecting the right activation functions and the role of technical indicators in enhancing prediction accuracy. This research demonstrated that LSTM models, particularly those with the TanH activation function, outperformed other models in the short-term.

forecasting of 30,60,90 Days, as evidenced by lower RMSE and MAE and higher R2 values, which are almost close to 1. The Results of the Model were Comparatively higher thanthe previous works done in same. Also, I would mention this paper's conclusion should also address the research problem directly which also reflects on the research's success.

The research highlights the potential of integrating Deep Learning Models with Different Optimizers with Different Activation Functions Combinations. This research had Forecast for a maximum period of 90 Days but in future we can do prediction for 365 days and see what outcomes we are getting. What more can be done is: Combining LSTM with other advanced machine learning models like GANs (Generative Adversarial Networks) or reinforcement learning for enhanced prediction capabilities. Extending the model to forecast global stock markets, considering different time zones, and economic conditions. This offers promising avenues for future research in financial forecasting utilizing advanced machine learning methods.

References

[1] Moghar, A. (2020). Challenges in Stock Price Prediction Using Machine Learning. International Journal of Computer Applications.

[2] Nabi pour, M. (2020). Feature Selection in Stock Price Prediction. International Journal of Advanced Computer Science and Applications.

[3] Joiner, D. (2022). Algorithmic Trading: A State-of-the-Art Review. Journal of Financial Research.

[4] C. D. K. and J. Dahlquist, Technical Analysis: The Complete Resource for Financial Market Technicians, FT Press, nov 2010.

[5] F. A. de Oliveira, C. N. Nobre and L. E. Z´arate, "Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index - case research of PETR 4 Petrobras Brazil", Expert Systems with Applications, vol. 40, no. 18, pp. 7596-7606, dec 2013.

[6] X. Pang, Y. Zhou, P. Wang, W. Lin and V. Chang, "An innovative neural network approach for stock market prediction," The Journal of Supercomputing, vol. 76, pp. 2098–2118, 2020.

[7] Hum Nath Bhandari, Binod Rimal, Nawa Raj Pokhrel, Ramchandra Rimal, Keshab R. Dahal, Rajendra K.C. Khatri,Predicting stock market index using LSTM,Machine Learning with Applications,Volume 9,2022,100320,ISSN 2666-8270,https://doi.org/10.1016/j.mlwa.2022.100320.

[8] F. A. Gers, J. Schmidhuber and F. Cummins, ”Learning to Forget: Continual Prediction with LSTM,” in Neural Computation, vol. 12, no. 10, pp. 2451-2471, 1 Oct. 2000, doi: 10.1162/089976600300015015.

[9] Sami HM, Ahshan KA, and Rozario PN. (2023). Determining the best activation functions for predicting stock prices in different (stock exchanges) through multivariable time series forecasting of LSTM. Aust. J. Eng. Innov. Technol., 5(2), 63-71. https://doi.org/10.34104/ajeit.023.063071Agrawal, M., Khan, A.U. and Shukla, P.K.,

2019. Stock Price Prediction using Technical Indicators: A Predictive Model using Optimal Deep Learning. International Journal of Recent Technology and Engineering (IJRTE), 8(2). DOI: 10.35940/ijrteB3048.078219.

[10] Kobiela, D., Krefta, D., Kr´ol, W., Weichbroth, P., 2022. ARIMA vs LSTM on NASDAQ Stock Exchange Data. Procedia Computer Science, 207, pp.3830-3839. DOI: 10.1016/j.procs.2022.09.445

[11] S. O. Ojo, P. A. Owolawi, M. Mphahlele and J. A. Adisa, ”Stock Market Behaviour Prediction using Stacked LSTM Networks,” 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Vanderbijlpark, South Africa, 2019, pp. 1-5, doi:10.1109/IMITEC45504.2019.9015840.

[12] N. Sharma and B. R. Mohan,” Stock Market Prediction Using Historical Stock Prices And Dependence On Other Companies In Automotive Sector,” 2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC), Hyderabad, India, 2022, pp. 425-431, doi: 10.1109/R10-HTC54060.2022.9930042.

[13] T. Damrongsakmethee and V. -E. Neagoe,” Stock Market Prediction Using a Deep Learning Approach,” 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, Romania, 2020, pp. 1-6, doi: 10.1109/ECAI50035.2020.9223142.

[14] Shouvik Banik, Nonita Sharma, Monika Mangla, Sachi Nandan Mohanty, Shitharth S.,LSTM based decision support system for swing trading in stock market, Knowledge Based Systems,Volume 239,2022,107994, ISSN 0950-7051,https://doi.org/10.1016/j.knosys.2021.107994.

[15] ” Prediction of Complex Stock Market Data Using an Improved Hybrid EMD-LSTM Model” Ali, M., Khan, D.M., Alshanbari, H.M. and El-Bagoury, A.A.-A.H., 2023. Prediction of Complex Stock Market Data Usingan Improved Hybrid EMD-LSTM Model. *Applied Sciences*, 13(1429). Available at: https://doi.org/10.3390/app13031429.

[16] Zaheer, S.; Anjum, N.; Hussain, S.; Algarni, A.D.; Iqbal, J.; Bourouis, S.; Ullah, S.S. A Multi Parameter Forecasting for Stock Time Series Data Using LSTM and Deep Learning Model. Mathematics 2023, 11, 590. https://doi.org/10.3390/math11030590

[17] A. Chatterjee, H. Bhowmick and J. Sen, ”Stock Volatility Prediction using Time Series and Deep Learning Approach,” 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, 2022, pp. 1-6, doi: 10.1109/Mysuru-Con55714.2022.997255.

[18] Hyun Jun Park and Youngjun Kim and Ha Young Kim,” Stock market forecasting

using a multi-task approach integrating long short-term memory and the random forest framework” 2022 Applied Soft Computing ,114,doi-https://doi.org/10.1016/j.asoc.2021.108106

[19] Artem Oppermann, ”Activation Function in Deep Learning Sigmoid TanHReLu,2021Available:https://artemoppermann.com/activation-functions-In-deep-learning-sigmoid-tanh-relu/comments