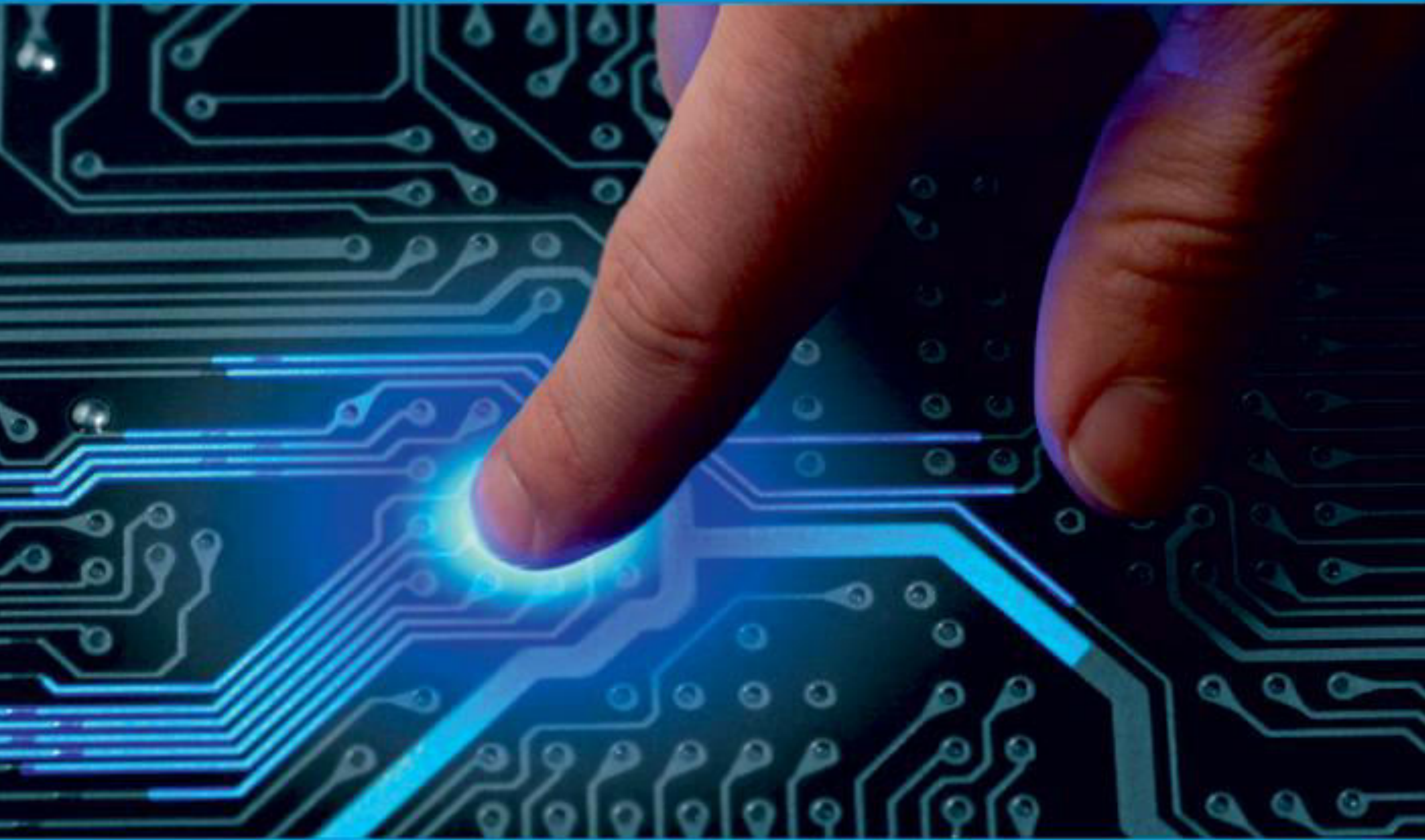


INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 1, January 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

·ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 1, January 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1201024 |

Accuracy of Stock Price Prediction Using the Sentiment Analysis

Dr. Chandrasekar Vadivelraju[1], Likhitha Chinnaobaiahgari [2], Korrapati Pavan Kalyan Gowd[3], Veerapu Reddy Suvarna Chandrika Reddy[4], Sallapuram Yaswanth Reddy[5], Tholla Ananth Kumar[6]

Professor, School of Computer Science and Engineering, Presidency University, Bangalore, India[1]

School of Computer Science and Engineering, Presidency University, Bengaluru, India[2,3 4,5,6]

ABSTRACT: Forecasting share prices is a complex mission given the dynamic and non-stationary nature of capital markets. Accurate predictions are challenging due to the inherent volatility and non-linear patterns in stock market behavior. This involves expecting the future value of a company's stock or other financial instruments traded on exchanges, aiming to enhance shareholder's gains. The latest success of applying Artificial Intelligence in finance has led to increased reliance on stochastic models for market predictions. Stock market prediction, a longstanding research focus, incorporates various machine learning techniques and diverse datasets. While many studies utilize historical stock statistics and relevant real-time data (e.g., oil and gold prices), few investigate the integration of Economic news in forecasting stock price trends. To address this gap, A system is proposed that leverages deep learning techniques and the LSTM (Long Short-Term Memory) algorithm for stock price prediction based on historical data of previous stock prices.

I.INTRODUCTION

Predicting stock prices in the financial market is exceptionally difficult because of its dynamic and unexpected nature. A few of the elements that affect stock prices are the state of the global economy, political situations, and a company's financial reports and performance. It becomes essential to look at past trends in order to maximize gains and minimize losses when reacting to stock market fluctuations. There are two main methods that have historically been suggested to forecast the stock price of a firm. To predict future stock prices, a technical analysis method looks at previous stock prices that include closing, opening, volume traded, and neighboring close values.

The second method, qualitative analysis, takes into account outside variables including the company's profile, the state of the market, political and economic forces, and textual data from blogs by economic analysts, social media, and financial news items. Large companies list their equities on the stock exchange in an effort to draw in investors and boost liquidity. Stockbrokers and traders who purchase and sell shares use the stock market as a central marketplace. Despite its allure, investing in the stock market carries risks due to the rapid fluctuations in stock values. Because of this volatility, which emphasizes how hard it is to predict stock values, several scholars have decided to carry out more research.

A solution to these issues is provided by a system that forecasts stock values based on historical market data utilizes the Long Short-Term Memory (LSTM) algorithm and machine learning techniques. This approach aims to increase the accuracy of stock market forecasts and provide investors with more reliable information for decision-making by tackling non-stationary, noisy, and chaotic data. The proposed system is a newcomer to the field of stock market forecasting.

Objectives:

The main objective of this project to foresee the stock price using machine learning methodologies and LSTM (Long Short Term Memory) algorithm by considering previous stock data.

Problem Statement:

Forecasting financial market trends presents a significant challenge due to the dynamic, noisy, and unpredictable nature of the data. Consequently, investors face difficulties in making profitable investment decisions. Numerous methods

·ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 1, January 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1201024 |

have been developed to predict stock market trends within the current body of literature. However, none have proven consistently accurate in their results.

Proposed System:
The introduced system makes use of machine learning techniques and utilizes the Long Short-Term Memory (LSTM) algorithm for the prediction of stock prices by analyzing historical stock data.

II.LITERATURE SURVEY

ETLP. K. Bharne and S. S. Prabhune: This paper introduces the fundamental principles of swarm intelligence and its application in optimizing daily stock market prices. It outlines various swarm intelligence algorithms such as ACO, PSO, BAT, and Firefly. The comparative analysis of recent swarm intelligence-based approaches concludes that SI-ANN yields more optimized results than SI with machine learning algorithms. The paper provides insights into recent SI trends and future directions.

ETL Mehar Vijha, Vinay Anand Tikkiwalb, Deeksha Chandolab, and Arun Kumarc: The article discusses the shortcomings of historical datasets with few features and suggests adding new variables to increase stock price prediction accuracy. In order to forecast the closing stock price for the following day, the article uses Artificial Neural Networks (ANN) and compares them with Random Forest (RF). Based on RMSE, MAPE, and MBE values, the results show that ANN is a better stock price predictor than RF.

A. Ray, A. Upadhyay, B. Gour, A. U. Khan, and B. P. Maurya: ETLB The machine learning model that was developed improves the accuracy of stock price prediction by utilizing the most recent technical indicators, such as Moving Averages, P/R, and MACD. Moving Averages are a useful tool for stock prediction since they help identify areas of support and resistance. The model leverages online scraping for real-time data directly from the stock market, making it adaptable to real-world and real-time stock problems

Bhattacharjee, ETLI, and Bhattacharja, P. The aim of the work is to compare the prediction capabilities and accuracy of machine learning and statistical methodologies. The most successful machine learning techniques for stock price prediction are MLP and LSTM in particular, as they show the lowest MSE and MAPE values. The usefulness of MLP and LSTM for precise stock price forecasting is highlighted in the study.
ETLU. Pasupulety, B. R. Mohan, S. Anmol, and A. Abdullah Anees: Sentiment analysis and ensemble learning are used in the suggested methodical approach to stock prediction. The use of SVM and Extremely Randomized trees is combined with a Stacked Regressor to enhance the accuracy of the anticipated pricing based on each individual model. The stock dataset incorporates sentiment analysis of public opinion from tweets as an extra feature, which improves forecast accuracy overall.

III.METHODOLOGY

The primary purpose of this project is to forecast the stock price for a specific date using a machine learning model. The first step is to get a reliable stock price dataset. After that, loading and preprocessing techniques, including normalization, scaling, and handling missing values, are applied. The preparation stage is finished, and the data is suitably divided into training and testing sets. Next, a model known as an LSTM (long short-term memory) is constructed. This kind of model is highly regarded for its ability to depict temporal dependencies.
The testing set is used for evaluation, and the training set is utilized for the process. Metrics like mean squared error and A framework named streamlit is used to develop a front end, which enhances user interaction. With this user interface, users can input their favorite date for stock price prediction. This input is seamlessly transferred to the model by the front end, which estimates the stock price for the specified date. The project concludes by giving the user access to the results via the front end, showcasing its worth as a helpful tool for stock market enthusiasts and investors alike.

It is essential to keep in mind that numerous factors influence the market, rendering stock value prediction unfeasible. Therefore, rather than providing precise predictions, a model may provide insights based on historical trends. The financial markets are extremely dynamic, so staying ahead of the curve requires both excellent algorithms and a keen understanding of market dynamics.

·ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 1, January 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1201024 |

IV.SYSTEM DESIGN

System Architecture

The first step is acquiring a dependable dataset containing stock prices and then proceeding to load it. Subsequently, the dataset undergoes preprocessing to purify and understand its features. This process encompasses tasks such as handling outliers, scaling, normalization, implementing feature engineering, and addressing missing values.
The data was partitioned into training and testing sets using a balanced ratio. Subsequently, the model was built utilizing Long Short-Term Memory (LSTM), a kind of recurrent neural network architecture specifically crafted to capture temporal dependencies within the input data. The model's efficacy was assessed by evaluating its performance on the testing set after undergoing training on the training set, employing accuracy and mean squared error as performance metrics. Following this, the finalized model was applied to predict stock prices for new dates based on user input from the front end.

The concluding step involves presenting the forecasted stock price to the user, delivering a valuable service for investors and enthusiasts in the stock market.

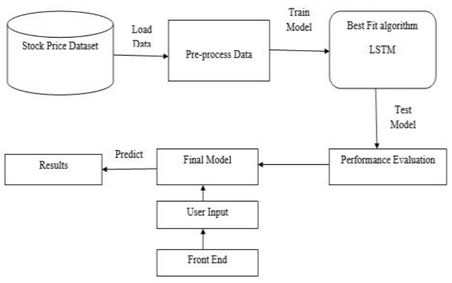


Figure 2.1

Use Case Diagram

Creating a machine learning model involves a systematic and intricate process, encompassing several essential stages. Initially, a diverse range of user data in varying formats like CSV, JSON, or Excel files is gathered. This raw data is then meticulously organized into a structured format conducive to thorough analysis. The second stage involves a detailed examination of the collected data, conducting exploratory analysis to uncover patterns and insights, and addressing potential data quality issues for the reliability and accuracy of the subsequent model.

Third stage is dedicated to preprocessing and model construction, a critical phase in the model development journey. Various techniques, including imputation, scaling, normalization, and encoding, are employed to transform raw data into a format suitable for the chosen machine learning algorithm. This stage also involves the pivotal task of selecting an appropriate machine learning algorithm aligned with the specific characteristics of the dataset. Once these preparatory steps are completed, the model is trained using the preprocessed data, refining its parameters and structure through methods like cross-validation or grid search.

In the fourth stage, the constructed model undergoes evaluation and testing using metrics such as mean squared error, mean absolute error and explained variance score R2_score. This process involves comparing the model's performance on training and testing sets while meticulously examining potential overfitting or underfitting issues. The fifth stage revolves around finalizing the model, with user feedback playing a pivotal role in reviewing the results. The completed model is then saved as a file or database object for future utilization.
The sixth stage involves the practical application of the model to predict results and showcase them. Users input new data, such as observations or queries, which undergo preprocessing before being input into the finalized model. The model generates predicted results, whether class labels or numerical values and presents these outcomes on a graphical user interface or web page. This user-friendly interface enhances the accessibility and usability of the machine learning model for both developers and end-users.

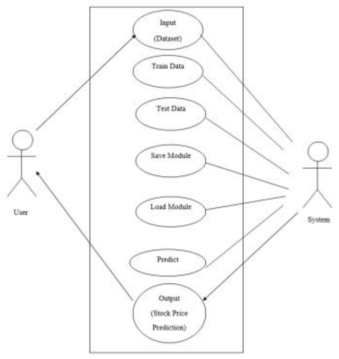


Figure 2.2

The primary focus of this project is to offer an extensive dataset containing historical stock prices for a selected company or index. This dataset is carefully curated from either a file or database and structured in a format that ensures its suitability for subsequent modeling processes. To enhance efficiency, the dataset undergoes a vital preprocessing phase, involving thorough cleaning and transformation procedures to tailor it to the chosen modeling approach.

In the operational phase, users input new data, which undergoes the same preprocessing procedures to ensure compatibility with the model. This preprocessed data is then input into the final model, which generates predicted stock prices based on the temporal patterns learned during training. The outcomes are presented to users through a user-friendly graphical interface or web page, enhancing accessibility and usability. This initiative holds significant potential to provide a valuable service for stock market enthusiasts and investors, delivering insights and predictions based on historical data and advanced modeling techniques.

Sequence Diagram

In the complex framework of a data processing system that leverages a machine learning model for predicting results based on user input, four integral components interact seamlessly: the User, Dataset, System, and Database.

The Dataset plays a pivotal role as a reservoir, housing crucial information necessary for generating predictions. Within its confines lie historical data, intricate patterns, and essential features that form the foundation upon which the machine learning model can formulate well-informed predictions. The dataset emerges as a cornerstone, shaping both the training and evaluation phases of the machine learning model.

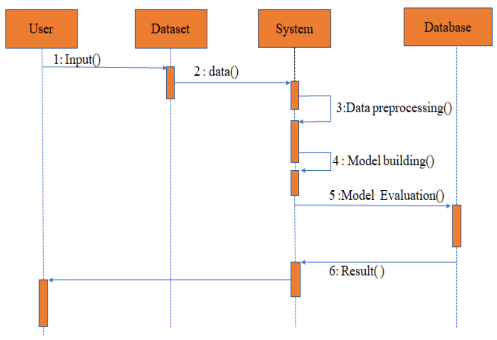


Figure 2.4

Following this, the system initiates the preprocessing of the data and the construction of the model, utilizing the LSTM (Long Short-Term Memory) algorithm. The system undergoes evaluation and finalization before being deployed to predict results based on the user's newly provided data. The outcomes of these predictions are stored in the database and subsequently presented to the user. This cohesive sequence of steps illustrates the synergy among the integral components, ensuring the delivery of efficient and user-centric data predictions.

e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 1, January 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1201024 |

Flow Chart

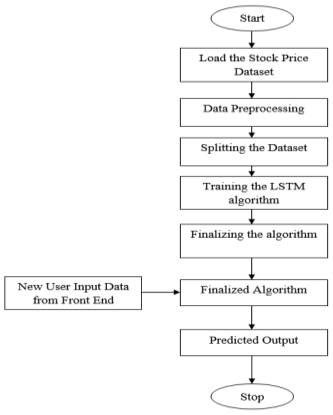


Figure 2.6

V.IMPLEMENTATION

Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNNs) belong to a class of neural networks where the preceding step's output is utilized as input for the upcoming step. In contrast to traditional neural networks where outputs and inputs function independently, RNNs prove advantageous in tasks involving sequential data, such as predicting the subsequent word in a sentence. The introduction of a Hidden Layer in RNNs addresses the need to consider the context of previous elements in a sequence. A noteworthy and distinctive feature of RNNs is the Hidden State, a mechanism designed to preserve information about the sequence.

How RNN works

Take a look at this example to learn how an RNN functions: Think of a deep network which has one input layer, one output layer, and three hidden layers. Similar to various neural networks, each hidden layer in a neural network has a distinct set of biases and weights. Weights and biases, for instance, would be (w1, b1) in the initial hidden layer, (w2, b2) in the second hidden layer, and (w3, b3) in the final hidden layer. This shows that no layer can store data from previous outputs and that each layer operates independently.

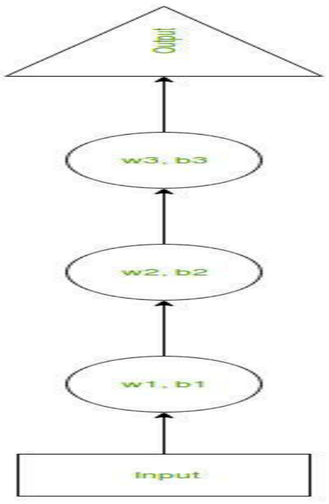


Figure 3.2

Recurrent Neural Networks (RNNs) redefine independent activations into interdependent ones by employing uniform weights and biases across all layers. This approach mitigates the need for an exponential surge in parameters and circumvents the challenge of memorizing each preceding output. The network accomplishes this by feeding the output of each layer as input into the subsequent hidden layer, establishing a sequential information flow. This intrinsic feature

·ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 1, January 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1201024 |

equips RNNs with proficiency in handling tasks involving sequential data, effectively capturing dependencies across various time steps.
As such, these three layers can be combined to create a single recurrent layer that incorporates the weights and biases of all hidden layers

Formula for calculating current state:

$$h_t = f(ht\text{-}1, X_t)$$

Where:

h_t ->current state

h_{t-1} -> previous state

x_t -> input state

Formula for applying Activation function(tanh):

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}X_t)$$

where:

w_{hh} -> weight at recurrent neuron

w_{xh} -> weight at input neuron

Formula for calculating output:

$$y_t = W_{hy}h_t$$

where:

Y_t -> output

W_{hy} -> weight at output layer

Training through RNN

1. The network undergoes the processing of a singular time step of the input.
2. The computation of the current state transpires by merging the current input with the previous state.
3. The resultant current state (ht) assumes the role of the input for the subsequent time step, displacing the preceding state (ht-1).
4. This cyclic process iterates for the necessary time steps dictated by the given problem, assimilating information from all antecedent states.
5. Upon the completion of all time steps, the ultimate current state is utilized to measuring the output.
6. The produced output is then compared to the target output, and the ensuing error is ascertained.
7. The error propagates backward through the network via the backpropagation mechanism, facilitating the fine-tuning of weights and the training of the Recurrent Neural Network (RNN).

Advantages of Recurrent Neural Networks (RNNs):

RNNs demonstrate proficiency in preserving and leveraging information over time, rendering them highly advantageous for time series prediction, courtesy of their Long Short-Term Memory (LSTM) capability. Furthermore, their effectiveness can be augmented by integrating convolutional layers to enhance the thorough consideration of pixel neighborhoods.

·ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 1, January 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1201024 |

Drawbacks of Recurrent Neural Networks (RNNs):
Training RNNs is a challenging task.When employing tanh or relu as an activation function, processing extremely lengthy sequences can be challenging.6.2.2 Memory for Long Short Term (LSTM)

Long Short-Term Memory (LSTM):
The Long Short-Term Memory Network (LSTM), an enhanced variant of the Recurrent Neural Network (RNN) designed specifically for successive tasks, offers a solution to the vanishing gradient problem that ordinary RNNs commonly face. Because of their ability to store data for a long time, RNNs are known in the field of persistent memory.
Consider reading a book and recalling the earlier chapters, or seeing yourself watching a movie and recalling the scenario that occurred before. Similar to this, RNNs work by processing incoming data using previously stored data. However, it is evident from the vanishing gradient problem that RNNs have limitations in terms of capturing long-term dependencies. LSTMs are a useful solution for managing and integrating long-term dependencies because they are expressly made to address these problems.

LSTM Architecture
In general, the Long Short-Term Memory (LSTM) and an RNN cell work similarly. The accompanying figure depicts the three primary parts, each of which plays a distinct role in the inside operations of the LSTM network.

The decision of whether to retain or discard the information from the preceding timestamp is made by the initial section of the LSTM. Following that, the purpose of the second section is to extract fresh data from the input that this cell has received. The third and final section is responsible for transmitting the updated data from the current timestamp to the subsequent one. The input gate, forget gate, and output gate are the terms for the initial, second, and final components of an LSTM cell, respectively. All of these components are called gates together.

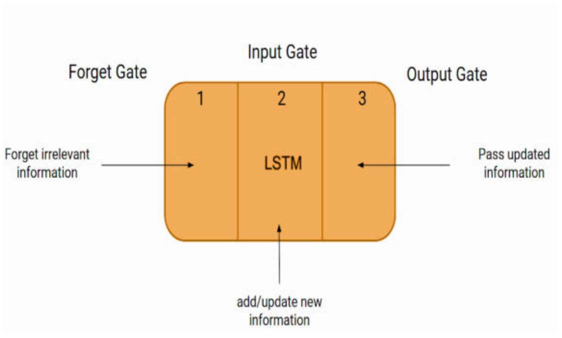


Figure 3.5

Like a basic RNN, an LSTM has a hidden state, where Ht-1 is the secret state of the timestamp that came before, and H t is the hidden state of the timestamp that came before that. Additionally, an LSTM has a cell state, denoted by C (t-1) for the preceding timestamp and C (t) for the succeeding timestamp. In this way, the hidden state is sometimes called short-term memory, whereas the cell state is known as long-term memory. Please see the image that is attached for more details.

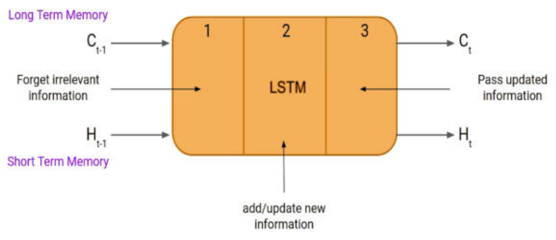


Figure 3.6

A notable observation is the ability of the cell state to preserve information across all timestamps.

·ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 1, January 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1201024 |

To illustrate the functionality of an LSTM, let's consider an example with two line sentences separated by a full stop. The initial line reads, "Bob is a nice person," followed by the second line, "Dan, on the other hand, is evil." The transformation from the first statement to the second requires the network to recognize the shift in focus from Bob to Dan prompted by the introduction of the full stop (.).

This transition depends on the Forget gate in the LSTM architecture, which allows the network to discard Bob-related data and detects the change in subject. Consequently, the network may now concentrate on taking in and comprehending the data related to Dan. This illustration highlights the crucial functions that various gates perform inside the LSTM architecture.

Forget Gate
In the initial phase of a network incorporating an LSTM cell, a crucial task is to determine if the data from the preceding timestamp should be preserved or discarded

Forget Gate:

Let's break down the equation:

- Xt represents the input at the current timestamp.
- Uf denotes the weight related with the input.
- Ht-1 signifies the hidden state of the preceding timestamp.
- Wf is the weight matrix related with the hidden state.

- $\mathbf{f_t = \sigma (xt * U_f + H_{t-1} * W_f)}$

Subsequently, a sigmoid function is utilized to transform f_t into a value within the range of 0 to 1. This resulting f_t is then multiplied with the cell state from the preceding timestamp, as illustrated below.

$C_{t-1} * f_t = 0 \quad ...if\ f_t = 0$ (forget everything)

$C_{t-1} * f_t = C_{t-1} \quad ...if\ f_t = 1$ (forget nothing)

If f_t equals 0, the network erases all stored information when t f_t is 1, it retains everything. In our example, the initial sentence centers on Bob. After encountering a full stop, the network shifts focus to Dan. Ideally, the network should forget about Bob in this scenario.

Input Gate
Let's look at another example: "Bob is a skilled swimmer. He informed me over the phone about his four challenging years in the service."
Although Bob is the subject of both remarks, they offer distinct insights on him. The first statement demonstrates his swimming ability, while the second one discusses his four years of navy service and phone use.
Given the context from the first statement, it becomes crucial to ascertain the important information in the second sentence, whether it has to do with phone usage or naval duty. In this perspective, it doesn't matter that he spoke with others over the phone; what matters is that he served in the navy.
The input gate manages this determination of importance. This evaluation process is captured by the input gate's equation, which determines the significance of the new information it has introduced.
Input Gate

- $i_t = \boldsymbol{\sigma}\ (x_t * U_i + H_{t-1} * W_i)$

·ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 1, January 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1201024 |

Here,

- X_t : Input at the current timestamp t
- U_i : weight matrix of input
- H_{t-1} : A hidden state at the previous timestamp
- W_i : Weight matrix of input associated with hidden state

Again the sigmoid function is employed on 'I,' resulting in a value at timestamp t that ranges between 0 and 1.

New information

- $N_t = tanh\ (x_t * U_c + H_{t-1} * W_c)$ (new information)

The updated data allotted for the cell state is determined by the input (x) at timestamp t and the hidden state at timestamp t-1. To ensure that the new information value (N_t) is restricted to a value between -1 and 1, the tanh activation function is employed in this situation. If Nt is positive at the current timestamp, information is combined with the cell state; if Nt is negative, data is subtracted from the cell state. However, instead of explicitly adding Nt to the cell state, the altered equation is shown. But by avoiding adding Nt directly to the cell state, the modified equation is introduced.

$$C_t = f_t * C_{t-1} + i_t * N_t \text{ (updating cell state)}$$

Here, C_{t-1} is the cell state at the current timestamp and others are the values that are measured previously.

Output Gate

Using the following line as an example, "Bob killed the enemy on his own and gave his life in defense of his nation." For his contributions, brave." Sentence completion challenges are mostly focused on the continuing task. It's critical to understand that when the word "brave" is used, it refers to one individual, in this case, Bob. Bob is the only one with this trait of bravery; it cannot be applied to the nation or the opponent.

Therefore, the task involves selecting a phrase that aligns with our expectations in order to give the statement a meaningful end. In this case, that particular term is the desired output, and the output gate's task is to provide it.Using the following line as an example, "Bob killed the enemy on his own and gave his life in defense of his nation." For his contributions, brave."

Sentence completion challenges are mostly focused on the continuing task. It's critical to understand that when the word "brave" is used, it refers to one individual, in this case, Bob. Bob is the only one with this trait of bravery; it cannot be applied to the nation or the opponent.

Therefore, the task involves selecting a phrase that aligns with our expectations in order to give the statement a meaningful end. In this case, that particular term is the desired output, and the output gate's task is to provide it.

Output Gate

- $O_t = \sigma\ (x_t * U_o + H_{t-1} * W_0)$

The sigmoid function helps to restrict the value of the output gate (Ot) to be between 0 and 1. As can be seen here, Ot and the hyperbolic tangent (tanh) of the updated cell state are used to measure the current concealed state.

- $H_t = O_t * tanh\ (C_t)$

·ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 1, January 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1201024 |

The current output and the long-term memory (C_t) appear to be related to the concealed state. Applying the SoftMax activation to the hidden state (H_t) will yield the output at the present timestamp.

- $Output = Softmax\ (H_t)$

Here the token with the mostest score in the output is the prediction.
This is the most intuitive figure of the LSTM network.

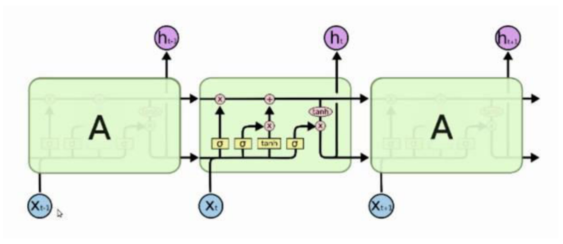


Figure 3.8

Dense layers

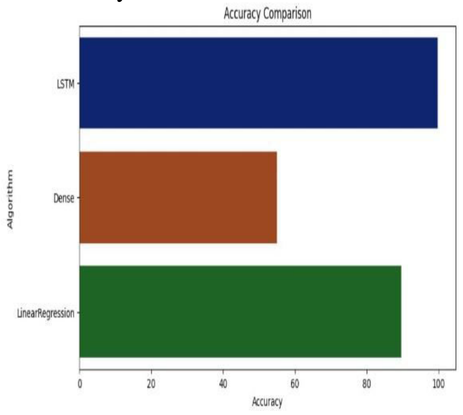
To extract the long-term dependencies that are present in stock data, LSTM (Long Short-Term Memory) models are widely used in the field of stock price prediction. Nevertheless, relying solely on LSTM models might not be sufficient to properly capture all the nuances included in the data. This is where the use of dense layers is necessary.

Neural network layers with fully linked layers—also called dense layers—have every input node tied to every output node. LSTM layers are typically placed before dense layers toward the conclusion of the model. They serve to categorize the traits that the LSTM was able to extract. They generate predictions that are appropriate for the specific task at hand, like as predicting the closing price of a stock, by utilizing the high-level, abstract representations generated by the LSTM layers.

For instance, in a stock prediction model, three LSTM layers might be used to find temporal dependencies in stock data, followed by three Dense layers which use the Rectified Linear Unit a kind of activation function. The Dense layers' task is to provide predictions after receiving the outputs from the LSTM layers, which contain newly discovered information about the input sequences.

To understand the temporal dependencies in stock data, for example, a stock prediction model could surely be constructed with three LSTM layers, then three Dense layers with an activation function of ReLU (Rectified Linear Unit). These Dense layers generate predictions once they receive the outputs from the LSTM layers, which include the knowledge they have learnt about the input sequences.

It is important to understand that, even while this strategy has the potential to be successful, stock price prediction is fundamentally difficult because there are so many variables at play, and no model can provide 100% accuracy.



ACCURACY TABLE:

S.NO	X-axis	Y-axis
1.	99%	LSTM
2.	55%	Dense
3.	85%	Linear Regression

Table 3.1

VI.CONCLUSION

Inventory trading is an essential activity in the intricate world of finance. One aspect of stock market prediction is predicting future prices for stocks and other financial instruments that are traded on different exchanges. This is a difficult assignment. The three main methods used by stockbrokers to try and anticipate stocks accurately are time

·ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 1, January 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1201024 |

sequence, technical, and fundamental analysis. Stock price forecasting has always been difficult because of the market's intrinsic volatility due to its dynamic and real-time nature, even with the widespread use of these analytical techniques.

To handle the difficulties of predicting stock market trends, a number of strategies have been developed. In this context, we propose a machine learning-based approach to enhance the accuracy of stock price projections. The primary goal of our approach is to apply neural networks, namely the RNN (Recurrent Neural Networks) subclass known as the LSTM (Long Short-Term Memory) algorithm.

This sophisticated model, which employs recurrent neural networks to capture temporal connections, provides a strong foundation for precise stock price predictions. When our suggested system is put into practice, it makes use of a recently curated stock price dataset, and the results show a notable improvement in the accuracy of stock price prediction. Through the utilization of LSTM and RNN algorithms, our model skillfully navigates the intricacies present in the dynamics of the stock market, thereby contributing to increasingly precise and dependable forecasts of the next day's stock values.

REFERENCES

[1] S. S. Prabhune and P. K. Bharne: The basic ideas of swarm intelligence are presented in this study along with how it can be used to optimize daily stock market pricing. It describes a number of swarm intelligence algorithms, including Firefly, ACO, PSO, and BAT. Based on a comparative review of modern swarm intelligence technologies, SI-ANN outperforms SI using machine learning algorithms in terms of optimum results. The report sheds light on current developments in SI and suggests directions for the future.
[2] Mehar Vijha, Vinay Anand Tikkiwalb, Arun Kumarc, and Deeksha Chandolab: In order to improve the accuracy of stock price predictions, the article highlights the drawbacks of using historical datasets with limited features and recommends the addition of new variables. The article compares Random Forest (RF) with Artificial Neural Networks (ANN) to predict the closing stock price for the next day. Based on RMSE, MAPE, and MBE values, the results demonstrate that ANN is a stronger stock price predictor than RF.
[3] Osman Hegazy and Omar S. Soliman: This work provides a machine learning model for financial technical indicator-based stock price prediction that integrates the LS-SVM and particle swarm optimization (PSO) algorithms. A stochastic oscillator, exponential moving average, money flow index, moving average convergence/divergence, and relative strength index are some of the indicators used in the study. LS-SVM-PSO outperforms single LS-SVM in terms of error value, and performs worse than the ANN-BP technique.
[4] B. Gour, A. U. Khan, B. B. P. Maurya, A. Ray, and A. Upadhyay: The machine learning model that was created uses the most recent technical indicators, like Moving Averages, P/R, and MACD, to increase the accuracy of stock price prediction. Because they make it easier to see regions of support and resistance, moving averages are a helpful tool for stock prediction. The model may be adjusted to real-world and real-time stock problems since it uses web scraping to obtain real-time data straight from the stock market.
[5] Bhattacharja and Bhattacharjee, P. and I.: Comparing the forecast accuracy and capability of statistical and machine learning approaches is the work's goal. Given that they exhibit the lowest MSE and MAPE values, MLP and LSTM in particular are the most effective machine learning approaches for stock price prediction. The study emphasizes the value of MLP and LSTM for accurate stock price forecasting.
[6] Abdullah Anees, A., B. R. Mohan, S. Anmol, and U. Pasupulety: The methodical approach to stock prediction that has been recommended makes use of sentiment analysis and ensemble learning. A Stacked Regressor is used in conjunction with SVM and Extremely Randomized trees to improve the precision of the expected pricing for each unique model. As an additional feature, sentiment analysis of public opinion from tweets is incorporated into the stock dataset, improving forecast accuracy overall.