**Mini Project Report on**



**STOCK PRICE PREDICTION SYSTEM USING MACHINE LEARNING**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name: Gauri Gupta** **University Roll No. :2018809**

***Under the Mentorship of***

**Dr. Surendra Kumar Shukla**

**Assistant Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**July-2023**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Stock Price Prediction System Using Machine Learning”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr. Surendra Kumar Sharma, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Gauri Gupta 2018809

Name University Roll no.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | 1 |
| Chapter 2 | Literature Survey | 4 |
| Chapter 3 | Methodology | 6 |
| Chapter 4 | Result and Discussion | 9 |
| Chapter 5 | Conclusion and Future Work | 11 |
|  | References | 13 |

**Chapter 1**

**Introduction**

* 1. **General Introduction**

Stock market prediction is a challenging task due to the instability and unpredictability of the market. This can be attributed to a variety of factors, including market volatility and other independent and dependent variables that can impact the market. While some of these factors can be quantified, such as historical data, current prices, and trading volumes, many cannot. The complexity of these factors can make it difficult to make accurate predictions using machine learning and other techniques. Factors like assets, debts, company performance, and value can impact the confidence of traders and investors, but incorporating them into a mathematical model for prediction is challenging. These factors make it difficult and unreliable to predict stock prices using machine learning, even for experts. Additionally, unexpected events such as pandemics or wars can significantly impact the market and make it difficult to anticipate market movements with high precision.

Therefore, instead of trying to predict long-term market movements with high precision, most market prediction models focus on short-term predictions of market trends on an average, uneventful day. This approach aims to provide a general sense of the market's direction rather than trying to predict specific values.

* 1. **Problem Statement**

The stock market is often in the news, with reports of new highs and lows. The ability to predict the short-term price of an individual stock could potentially increase investment and business opportunities in the market.

Previous methods for stock predictions have utilized Artificial Neural Networks and Convolution Neural Networks, but these methods often have an error loss rate of around 20%. In this report, we will explore the potential for using a Recurrent Neural Network to devise a model that can predict stock prices with a lower error rate.

* 1. **Technologies**

**1.3.1 PYTHON**

Python was chosen as the language for this project for several reasons, one of which is its large and active community. This means that any problems or issues that may arise can be easily resolved by seeking help from the community. Additionally, Python has many popular libraries such as TensorFlow and scikit-learn which are widely used for machine learning and data preprocessing. The availability of powerful tools for scientific computing such as NumPy, Pandas and SciPy also makes it easy to perform complex tasks such as data manipulation, data analysis and data preprocessing. Another benefit of using Python is that it is a user-friendly language that allows for code that is easy to read and understand, making it a great choice for research and experimentation.

**1.3.2 NUMPY**

NumPy is a Python library that provides advanced mathematical capabilities and abstractions for scientific computing. It is particularly useful for working with arrays, which are efficient data structures commonly used in numerical computations, such as manipulating matrices. NumPy provides many functions that can be used to create and manipulate arrays, including mathematical functions and operations, linear algebra and more. It is widely used in data science and machine learning for data manipulation, data analysis and data preprocessing.

**1.3.3 SCIKIT LEARN**

Scikit-learn is a widely used machine learning library for Python that provides a variety of algorithms for classification, regression, and clustering. It includes popular algorithms like support vector machines, random forests, gradient boosting, and k-means. One of its key features is its seamless integration with other popular Python scientific computing libraries such as NumPy and SciPy, which makes it easy to integrate with other data analysis and data preprocessing tasks. This makes it a powerful tool for machine learning and data science projects. Additionally, it is open-source, easy to use, and has a large and active community of users and contributors.

**1.3.4 TENSORFLOW**

TensorFlow is an open-source library for numerical computation that uses data flow graphs. It allows for the creation and execution of mathematical operations represented as nodes in the graph, and the communication of multidimensional data arrays (tensors) between them. The architecture is flexible, which allows for deployment on various devices such as CPUs, GPUs, desktops, servers, or mobile devices, through a single API. It was originally developed by the Google Brain Team for machine learning and deep neural network research but can be applied to other fields as well.

**1.3.5 KERAS**

Keras is a high-level API for building neural networks, written in Python and able to run on TensorFlow, CNTK or Theano. It aims to enable fast experimentation and is designed for easy and fast prototyping, with features such as user-friendliness, modularity and extensibility. Keras supports both convolutional and recurrent networks and their combinations, and it can run on both CPU and GPU. It is a popular tool for building and training machine learning models.

**Chapter 2**

**Literature Survey**

In this section, the study will investigate existing literature and studies in the field of technical and financial analysis. The objective is to understand the methods and techniques that have been used to analyze the stock market and predict future stock prices. The current state of research in these areas will be reviewed to gain a comprehensive understanding of the existing approaches and their limitations.

**2.1 Technical**

[3] Kim, K., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. Expert Systems with Applications, 19, 125-132.

[4] Liu, S., Zhang, C., & Ma, J. (2017). CNN-LSTM neural network model for quantitative strategy analysis in stock markets. International Journal of Machine Learning and Cybernetics, 1(1), 198-206.

[5] Long, W., Lu, Z., & Cui, L. (2018). Deep learning-based feature engineering for stock price movement prediction. Knowledge-Based Systems, 164, 163-173.

**2.2 Financial**

This section describes the data that was collected from public sources. The dataset includes records of AXISBANK's daily stock exchange from 2000 to 2021, published publicly by NSE (National Stock Exchange) India [1]. The dataset was saved in CSV format for easy retrieval and use throughout the project. The dataset includes information such as date, open, high, low, close, and volume prices.

|  |  |
| --- | --- |
| Price | Description |
| Open | The price of the stock at the beginning of the trading day |
| High | The highest price of the stock on the trading day |
| Low | The lowest price of the stock on the trading day |
| Close | The last price of the stock before the trading day ends |

**Table 2.1 Description of dataset**

**Chapter 3**

**Methodology**

**The Long Short-Term Memory**

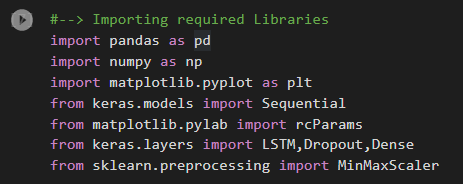
The Long Short-Term Memory (LSTM) method will be used to create a Machine Learning model to forecast Microsoft Corporation stock values. The LSTM is a type of deep learning recurrent neural network that is well-suited for time-series data. They are known for their ability to make small adjustments to information by multiplying and adding. Unlike traditional feed-forward neural networks, LSTMs have feedback connections, which enables them to handle both single data points and full sequences of data such as speech or video.

LSTMs work in a three-step process:

• The first step in an LSTM model is deciding which information to keep or discard in a given time step.

• The second layer contains two functions, the sigmoid function and the tanh function. The sigmoid function is used to decide which values to allow through (0 or 1), while the tanh function assigns a weight to the values passed, determining their level of importance from -1 to 1.

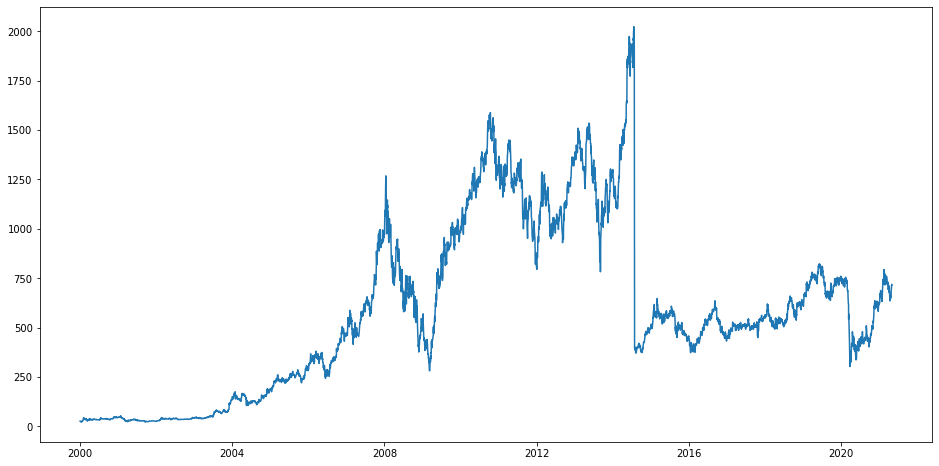
• The third step is deciding the final output of the model.

**3.1 Importing the Libraries**

The LSTM models will be constructed with the help of TensorFlow Keras, and the data will be cleaned and prepared using scikit-learn.

**3.2 Load The Training Dataset**

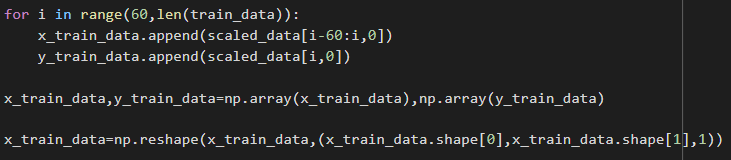
****

**3.3 Plot the true Close Value of the stock**

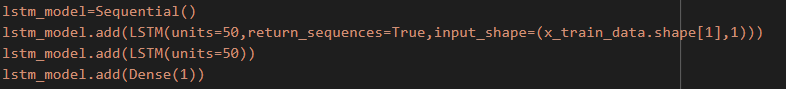
**Fig 3.1 Graph of actual closing values of stock between 2000 to 2021**

**3.4 Splitting Data for Training and Testing**

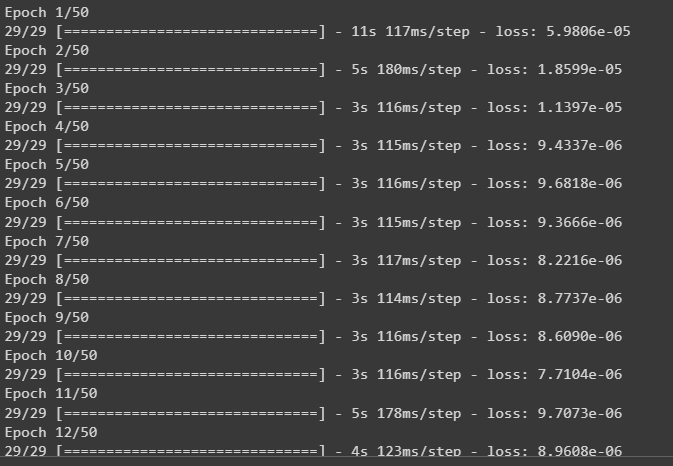
The entire dataset needs to be divided into a training and test set before feeding it into the training model. The LSTM model will be trained on the data in the training set and tested for accuracy and backpropagation on the test set. This is a common practice in machine learning and helps to evaluate the performance of the model on unseen data.

****

**3.5 Training and building the LSTM model**

[2] A Sequential Keras model with one LSTM layer will be constructed. The LSTM layer will have 32 units and will be followed by one Dense layer with one neuron. This simple architecture will be used to predict the stock prices of AXISBANK.****

**3.6 Taking a sample dataset and predicting stock price**

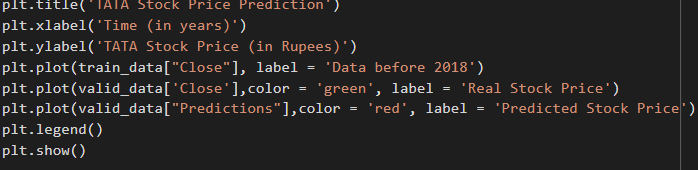
****

**3.7 Saving LSTM model and Prediction**

****

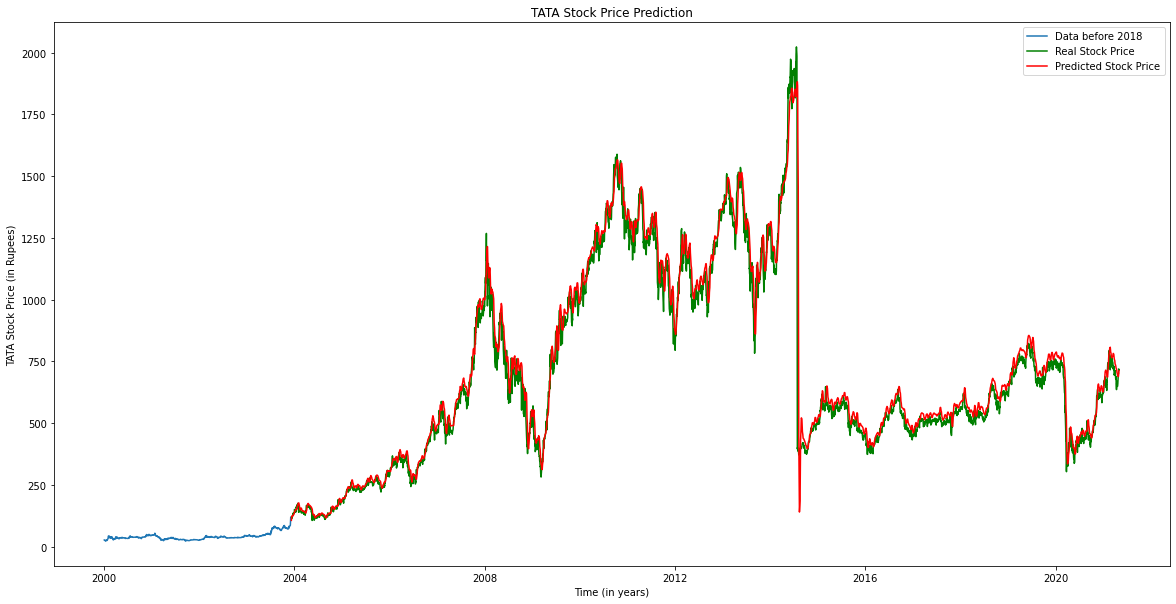
With the model in place, it can be used to forecast the Adjacent Close Value of the stock by using a model trained with the LSTM network on the test set. The predict function of the LSTM model is used to accomplish this task. This function will use the trained model on the test set to generate predictions.****

**3.8 Comparing the Predicted Value and True Value and Visualizing it**

****

**Chapter 4**

**Result and Discussion**

****

**Fig 3.1 Graph of Real Values vs Predicted Values**

The model has shown the ability to predict stock trends with a high level of accuracy. However, additional training on more data and incorporating more LSTM layers may further improve its performance.

The results of stock market prediction using machine learning can vary based on factors such as the techniques and models employed, the quantity and quality of data used, and the specific stock or market being predicted. Possible outcomes include:

1. Predictions of future stock prices which can aid in trading decisions.
2. Predictions of stock trends, such as whether prices will rise or fall, which can also aid in trading decisions.
3. Predictions of stock volatility, which can assist in managing risk and making trading decisions.
4. Predicted buy and sell signals: The model generates buy or sell signals, which can be used in making trading decisions.
5. Model evaluation metrics: The model can be evaluated using various metrics like accuracy, precision, recall, and F1-score to measure its performance and identify areas for improvement.
6. Real-time prediction: The model can make real-time predictions which can be used to make trading decisions in real-time.

It is important to note that prediction results are not always accurate, and other factors such as market conditions, company performance, and investor sentiment can also have a significant impact on stock prices.

**Chapter 5**

**Conclusion and Future Work**

Although not perfect, LSTM models have shown to be able to predict stock price behavior with a certain level of accuracy. Some advantages of using LSTMs include:

* Handling sequential data: LSTMs are particularly well-suited for handling sequential data such as time-series data, making them a popular choice for stock market prediction.
* Handling long-term dependencies: LSTMs are able to capture long-term dependencies in data, which is crucial for stock market prediction as stock prices are affected by historical events and trends.
* Handling noise in data: LSTMs are able to handle noise and missing values in data, which is common in financial data.
* Handling varying sequence lengths: LSTMs can handle sequences of varying lengths, which is important for stock market prediction as the number of trading days in a year can vary.
* Handling non-linear relationships: LSTMs can handle non-linear relationships in data, which is important for stock market prediction as stock prices can be affected by multiple, non-linear factors.
* Handling large datasets: LSTMs can handle large datasets, which is important for stock market prediction as financial data can be voluminous.

There are several areas for future work in stock market prediction using machine learning:

* Incorporating more diverse and extensive data sources: This can include financial statements, news articles, and social media sentiment to improve the accuracy of the predictions.
* Ensemble models: Combining multiple models to improve the prediction accuracy.
* Adding more features: Including more features such as volatility, volume, and moving averages, can make the model more robust.
* Real-time prediction: Developing systems that can monitor the stock market in real-time and make predictions on the fly.
* Exploration of advanced techniques: Researching and experimenting with techniques such as reinforcement learning to see if it can improve the model's predictions.
* Incorporating other markets: Examining the possibility of integrating other markets such as the commodity market or bond market to see if it improves the stock market predictions.

**References**

[1] Stock Market Dataset of different Stocks in Nifty-50 NSE from 2000-2021. Accessed on 23 January 2023.

[https://www.kaggle.com/datasets/rohanrao/nifty50-stock-market-data](%20%20%20%20%20%20https://www.kaggle.com/datasets/rohanrao/nifty50-stock-market-data)

[2] Training Model inspired and developed.

<https://youtu.be/lncoLfue_Y4>

<https://youtu.be/OXwZtlcTiuk>

[3] Kim K, Han I. Genetic algorithms approach

<https://doi.org/10.1016/S0957-4174(00)00027-0>.

[4] <https://doi.org/10.1007/978-3-319-70096-0>.

[5] <https://doi.org/10.1016/j.knosys.2018.10.034>.