StockX
(Visualization And Forecasting of Stocks)

A Mini Project Report Submitted
For
Partial fulfillment of the requirements of the
Degree of Bachelor of Engineering
IN
COMPUTER ENGINEERING

(Semester VI)
BY

Enrique Crasto (9247)
Suraj Naik (9274)
Harsh Parmar (9278)
Joseph William (9261)

Under the guidance of
Prof. Unik Lokhande

DEPARTMENT OF COMPUTER ENGINEERING
Fr. Conceicao Rodrigues College of Engineering
Bandra (W), Mumbai - 400050
University of Mumbai

2022-2023

This work is dedicated and devoted to our family.
We are grateful for their motivation, help, and support.

CERTIFICATE

This is to certify that the mini project entitled " StockX " is a bonafide work of " Enrique Crasto (9247), Suraj Naik (9274), Harsh Parmar (9278), Joseph William (9261)" submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering in Computer Engineering (Semester- VI).

Prof. Unik Lokhande
Guide/ Supervisor

Dr. Sujata Deshmukh
HOD-Computer Department

Dr. S. S. Rathod
Principal

Approval Sheet

Mini Project Report Approval for T.E. (Semester-VI)

This mini-project report entitled StockX submitted by Enrique Crasto (9247), Suraj Naik (9274), Harsh Parmar (9278), and Joseph William (9261) is approved for the degree of Bachelor of Engineering in Computer (Semester-VI).

Examiner 1.―――――――――――

Examiner 2. ―――――――――――

Date:

Place:

Declaration

We, Group 13, declare that this written submission accurately portrays our views in our own words, and that whenever other people's ideas or words have been incorporated, we have properly cited and referenced the original sources. We further swear that we have followed all academic honesty and integrity rules in our work and have not misrepresented, faked, or falsified any idea/data/fact/source. We accept that any breach of the foregoing will result in disciplinary action by the Institute, as well as penal action from the sources that were not correctly cited or from whom sufficient permission was not obtained when required.

Date:

Enrique Crasto
(9247)

Suraj Naik
(9274)

Harsh Parmar
(9278)

Joseph William
(9261)

Abstract

The demand for stocks has come huge with the rise in fashion ability of Stock Digital World. Stock forecast and breakdown can be salutary for people to suppose before buying or dealing stocks. So new stock price forecasts will be anatomized and imaged through machine literacy algorithms. One of the intelligent data mining methods that has been used by experimenters in colorful areas for the first ten times is the neural network. Forecasting and analysing stock request data has been critical to the current frugality. Colorful soothsaying algorithms can be divided into direct and non-linear models. In this paper, we forecast a company's stock price based on literal prices using four types of deep learning configurations: Multilayer Perceptron, Recurrent Neural Networks, Long Short-Term Memory, and Convolutional Neural Network.

Keywords: RNN, CNN, LSTM, MLP, GNP, RMSE, BSE, NSE

Acknowledgments

We are delighted to give the report on "StockX." We would like to use this occasion to convey our heartfelt appreciation and gratitude to our guide, Prof. Unik Lokhande, CRCE, Bandra(W), Mumbai, for offering technical guidance and ideas for the scope of this study. During our trips to the department, we loved discussing the job progress with him/her.

We would like to express our gratitude to Dr. Sujata Deshmukh, Head of the Computer Engineering Department, Principal, and the management of C.R.C.E., Mumbai, for encouraging and providing the required infrastructure for the project's success.

We would also like to thank all non-teaching staff for their invaluable assistance in completing our project.

Date:

Enrique Crasto (9247)
Suraj Naik (9274)
Harsh Parmar (9278)
Joseph William (9261)

Table of Content

List of Tables

List of Figures

Chapter 1

INTRODUCTION

As any of us could have guessed, the market is unpredictable and further than frequently changeable. For decades, experimenters have been tinkering with time- series data to forecast unknown values the most grueling and potentially economic operation of which is prognosticating the value of a given company's stock.

Still, as anticipated, request change depends on numerous parameters, only a many of which can be quantified, similar as literal stock data, trading volume, current prices. Of course, abecedarian factors similar as the company's natural value, means, daily performance, recent investments, and strategy influence dealers' confidence in a company, and therefore its stock price. Only a many of them can be effectively incorporated into a mathematical model.

Numerous proxies influence stock markets, producing distrust and extreme volatility. Despite the fact that humans Automated trading systems (ATS) that are handled by the execution of computer programs can follow through better and with advanced incentive in sending orders than any mortal. Nonetheless, in order to assess and manage the performance of ATSs, risk strategies and safety measures based on earthborn judgments must be implemented.

When designing an ATS, several aspects are incorporated and considered, such as the trading strategy to be adopted, complicated fine functions that mirror the state of a certain stock, machine learning algorithms that enable forecasting of the unknown stock value, and specific stock news being analyzed.

Weather forecasting and financial market forecasting are only two examples of the many real-world applications of time-series forecasting. It forecasts the outcome for the following time unit using the ongoing data over a period of time. Numerous time series forecasting methods have proven successful in the real world.

Chapter 2

OBJECTIVES OF THE PROJECT

1. Learn and explore different machine learning algorithms.
2. To study and improve these stock price forecasting algorithms.
3. Build an online system to predict long term trends and short term trends of various stocks.
4. To create a user-friendly web application which will allow users to view prediction results and visualize the trend.
5. Creating a website with a user-friendly interface and best experience.
6. Making a more accurate, cost-effective, and resource-efficient system.

Chapter 3

SCOPE OF THE PROJECT

The stock market is essentially an assortment of different stock buyers and sellers. Shares typically indicate ownership claims made by a certain person or group of people to a firm.

Stock market forecasting is the process of attempting to predict how much a stock will be worth in the future. It is anticipated that the predicting will be reliable, accurate, and useful. The system needs to function in real-world situations and be capable in those environments.

Additionally, it is expected that the algorithm will take into account every factor that could influence the stock's performance and valuation. There are many different approaches and ways to use a prediction system, including basic analysis, technical analysis, machine learning, market mimicking, and time series structuring. Recently, many experimenters have employed various ensemble learning techniques.

Chapter 4

REVIEW OF LITERATURE

Table 1. Review of Literature

Paper	Algorithms	Results	Research Gap/ Review
[1]	To improve performance, a model (XGBoost) and a one-dimensional convolution neural network (CNN) called CGBoost are proposed in this paper.	This method significantly improves the calculation results. In each year and index, CGBoost and CGBoost6 have lower annual average forecast error.	Due to extensive data cleaning the data loss reduce the accuracy
[2]	This research presents an in-depth learning strategy based on CNN for predicting stock prices.	The CNN stock price forecasting method used in this paper is both accurate and useful.	The algorithm is not fit for 1D data
[3]	CLDNN is proposed by the author, which first imports input data to the CNN layers to obtain better features, then implements LSTMs based on the feature, and then produces output via fully connected DNNs.	Experiments show that DWNN outperforms general RNNs.	Due to extensive data cleaning the data loss reduce the accuracy
[4]	The paper proposed prediction models based on RNN, LSTM, and GRU.	The GRU-M and LSTM-M performed significantly better than the RNN-M, with the GRU-M slightly outperforming the LSTM-M.	Due to many input layers the number of hidden layers is slowing down the algorithm.
[5]	In this paper, a model for forecasting future stock values utilising time series data and a recurrent neural network containing an LSTM cell is built.	The paper demonstrates that the developed RNN-LSTM model can accurately predict the stock price one day ahead. The predicted stock price is nearly identical to the actual price.	The algorithm is not fit for 1D data.
[6]	To predict the upcoming daily stock price, a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) is used (High, Low, Open, Close). Applying Deep Belief Network (DBN) allows for a result-to-LSTM result comparison.	When the different techniques are compared, the result shows that LSTM outperforms CPALL, SCB, and KTB. LSTM is appropriate for stocks with low volatility.	Due to extensive data cleaning the data loss reduce the accuracy

[7]	In order to provide predictions for both short- and long-term horizons, this research introduces two innovative prediction algorithms integrated with a deep learning framework and utilising long short-term memory units.	Long-term stock price prediction performed reasonably well using RNN and LSTM models. Furthermore, in both overall prediction and selected intervals, the Hybrid method outperforms the Recursive method.	The algorithm is not fit for 1D data
[8]	This study builds an RNN model to estimate stock market movements using intraday stock data, which contains more information than daily stock data. The model, which includes more than 100,000 data for training and testing, uses a dataset containing hundreds of stocks spanning more than nine years.	The RNN model allows us to learn non-linear relationships from features and forecast targets, and we can use precision and average profit to determine how effective our model is. RNN is effective at forecasting stock price changes, and it can also help stock market investors make investment decisions.	Due to many input layers the number of hidden layers is slowing down the algorithm.
[9]	In the current work, a hybrid deep learning model that combines AM, MLP, and BiLSTM is proposed.	To forecast the closing prices of four stock indices, a novel hybrid deep learning model integrating attention mechanism (AM), MLP, and bidirectional long short term memory neural network (BiLSTM) was proposed. According to the results, the proposed model performed the best in terms of MAE, MSE, MSLE, MedAE, and R2.	Due to extensive data cleaning the data loss reduce the accuracy
[10]	Resilient propagation (RPROP), Levenberg-Marquardt (LM), scaled conjugate gradient (SCG), and one-step secant conjugate gradient (OSSCG) are some of the learning methods used for MLP training.	MLP networks do not have the ability to estimate the long-term dependance of an auto-similar process, and a more thorough analysis of its parameter optimization process using hybrid techniques is required to confirm such a claim.	Due to many input layers the number of hidden layers is slowing down the algorithm.
	Three nodes make up this. A programme with an enhanced	The simulation results show that the proposed	The algorithm is not fit for 1D data

[11]	GNP structure—GNP with reinforcement learning—is created (GNP-RL).	method with ensemble learning and buy&hold solves the effectiveness of ensemble learning using MLP for stock trading problems.	
[12]	Using two algorithms—Multilayer Perceptron (MLP) and Long Short Term Memory—for modelling a neural network (LSTM). The layer architecture of both algorithms is identical and consists of three levels: an input layer, two hidden layers, and an output layer.	According to the findings of this study, modelling with the MLP algorithm outperforms LSTM in terms of performance. RMSE and R-squared values for MLP are 86.86% and 3.19%, respectively; 74.24% and 3.94% for LSTM.	The algorithm is not fit for 1D data
[13]	In this research paper they have used LSTM along with RNN	They used one of the most precise forecasting technologies, the Long Short-Term Memory unit, which assists investors, analysts, and anyone interested in investing in the stock market by providing them with a good understanding of the stock market's future situation.	Due to extensive data cleaning the data loss reduce the accuracy
[14]	In this research paper they have used LSTM and BI-LSTM.	We can see that Deep Learning algorithms have had a significant impact on modern technologies, particularly in the development of various time series-based prediction models. They have the highest level of accuracy for stock price prediction when compared to other regression models. Among the various Deep Learning models, both LSTM and BI-LSTM can be used for stock price prediction with proper parameter adjustment.	Not able to explain dropout algorithm properly.
	In this research paper they have used stacked LSTM	Using data from the NASDAQ Composite (IXIC), this study	The algorithm is not fit for 1D data

[15]		proposes the use of a stacked LSTM network model for predicting stock market behaviour. The model was trained, and the results show that it can predict stock market behaviour with some accuracy.	
[16]	In this research paper they have used Attention-LSTM algorithm	In this paper, a feasibility analysis of a stock trend prediction method based on the attention mechanism LSTM deep learning model and the representative of China's stock market indexes the Shanghai index, Shenzhen index, and the csi 300 index was performed, and the algorithm's effectiveness was verified.	Not able to explain dropout algorithm properly.
[17]	In this research paper they have used LSTM algorithm along with RNN	They analysed the growth of companies from various sectors in this paper to determine the best time span for predicting the future price of the share. As a result, this leads to the important conclusion that companies in the same industry have the same dependencies as well as the same growth rate. If the model is trained on a larger number of data sets, the prediction will be more accurate.	LSTM prefer to initialize with little weight, so small dataset was used.

We learned that LSTM is more efficient in comparison to other algorithms.

LITERATURE SUMMARY

The papers in this list generally aim to predict stock prices using various deep learning techniques. However, they also have several shortcomings that are worth noting.

First, many of these papers focus on short-term predictions, which may not be useful for long-term investment strategies.

Second, they often use a limited amount of data, which can limit the accuracy of the predictions.

Third, some of the papers lack a clear explanation of the features used in the prediction models, making it difficult to assess their effectiveness.
Fourth, some papers focus only on a single stock or sector, which may not generalize well to other stocks or sectors. Fifth, there is a lack of comparative analysis between different deep learning techniques, making it difficult to determine the most effective approach for stock price prediction.
Sixth, some papers suffer from data leakage, where future price information is inadvertently included in the training data, leading to overly optimistic results. Seventh, many of the papers do not account for external factors such as news events, market sentiment, and economic indicators, which can have a significant impact on stock prices.

Overall, while these papers provide valuable insights into the use of deep learning for stock price prediction, there are still many challenges to overcome to improve the accuracy and usefulness of these models. Further research is needed to address these shortcomings and develop more effective approaches for stock price predict

Chapter 5

PROPOSED SYSTEM

5.1 DRAWBACKS OF THE EXISTING SYSTEM

1. Other data sets didn't produce as outstanding buy and sell accuracy when a CNN with the same framework was used. It does not incorporate demonstrative elements like news and public politics into the forecast, merely the influence of stock price data on closing prices.

2. The performance of the model depends on the caliber of the training in multilayer perceptrons, where the overall parameter composition might reach very high values.

3. The effectiveness of the model for multilayer perceptrons depends on how well it was trained.

4. LSTMs are affected by a different initialization of the random weights and therefore behave very similar to a feedforward neural network. Instead, they prefer to initialize with little weight.

5. LSTMs are prone to overfitting, making it difficult to use the dropout algorithm.

5.2 PROBLEM STATEMENT

Before anyone invests in any stock, we need to understand how the stock market behaves.

Investing in a good stock but at the wrong time can have catastrophic results, while investing in an average stock at the right time can bring gains.

Amateurs face this problem of trading because they don't understand duly which stocks to buy or which stocks to put up to get optimum gains.

In analytics and data science, time series are widely utilized. Stock prices fluctuate widely and are unpredictable in their nature.

Chapter 6

SYSTEM DESIGN

6.1 MODULE DESCRIPTION

Interface module:

Description and Priority-

This feature allows personnel to view the application. In general, the response of a web interface module should be designed to provide feedback to the user, indicating whether their input was successful, and if not, why it failed. This is an important feature, so it has high priority.

Stimulus/response sequences

Users first click on a button or link to start the registration process. The system then prompts the student to fill in their first name, last name, email address and password. Students enter the field. The system will verify the user data and a new account is created for students.

Data collection module:

Description and Priority-

It generally deals with composing the right data set. The data set to be used in market forecasting must be used to filter grounded on varied aspects. Data collection also complements the data set by adding fresh data that's external. Our data consists substantially of stock prices from the former time. We'll originally dissect the Kaggle data set and according to the accurateness, we will use the model with the data to directly the forecasts can be analyzed.

Administrator module:

Description and Priority-

Computer servers and Networks are supported and maintained by administrators

Functional requirements

REQ-1: Separate login.

REQ-2: To login as normal users, master password and ID is required.

REQ-3: Tools for implementing automated software solutions.

REQ-4: Tools for replacing hardware and software components.

Preprocessing module:

Description and Priority-

Data preprocessing involves looking for missing values, looking for categorical values, splitting the dataset into a training and testing set, and finally performing feature scaling to reduce the range of variables.

6.2 ALGORITHMS USED

RNN (Recurrent Neural Network)

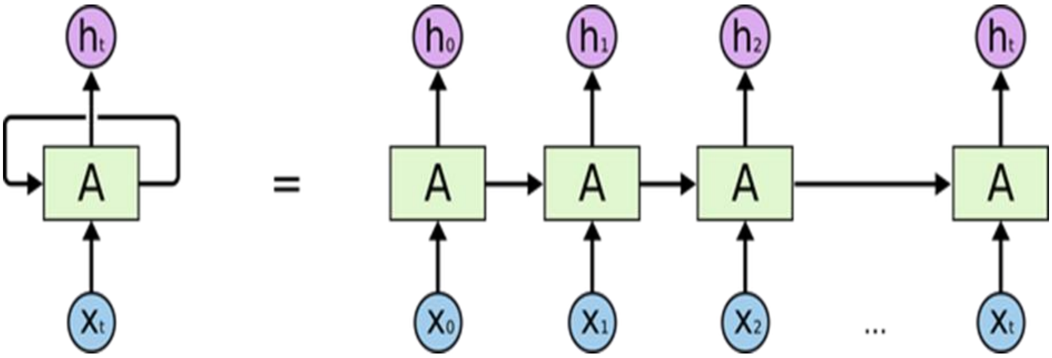

Figure 1 RNN Model

As in the feed-forward case, we feed the input into our neuron (block A), it does some computation, and we get the output h_t However, note an additional recurrent connection feeding the same output h_t back into A (the connection is drawn leaving block A on the right size, and entering on the left). Note this recurrent connection will be used as part of the input computation at the next time step. We will get another input, xt+1, and will feed that into our neuron (Block A), however recall our recurrent connection from the previous time step, ht, this is also part of our input. In a simple function mapping, a RNN's computation will be, following the diagram notation above: ht=f(xt, ht−1)

This means that for RNN, for each neuron, there are two weights, a feed-forward weight (just like we would have in an MLP) and a recurrent weight.

Unrolling Through Time

If we took our RNN computation for say 10-time steps, and unrolled this computation through time, we will arrive at the picture below.

In this diagram, we have unrolled our RNN's computation through time, starting at T=0 all the way to T=t. sees the similarity to the MLP now? The recurrent connection at every time step acts as the bridge connection between time steps. (This is the component often referred when we talk about RNN's back-propagation through time). We can think of RNN as similar to a MLP/Feed-forward network, but instead of expanding through layers, we expand an RNN computation through time, as well an RNN has the same weights applied at every time step (note we're using the same block A for computation at all time steps).An alternative way of thinking about RNN is that it has a persistent state throughout all time steps of computation, specifically the h_t. The RNN should learn to encode information from earlier time steps into h_t, so the network can use this encoded information for computation at much later time steps.

For example, a well-trained RNN on generating sentences should learn to encode the fact it opened a bracket at say time step 2 into h2, which gets carried forward in every step of computation, and the RNN will close the bracket at time step 10 with the use of h9.

LSTM (Long-short Term memory)

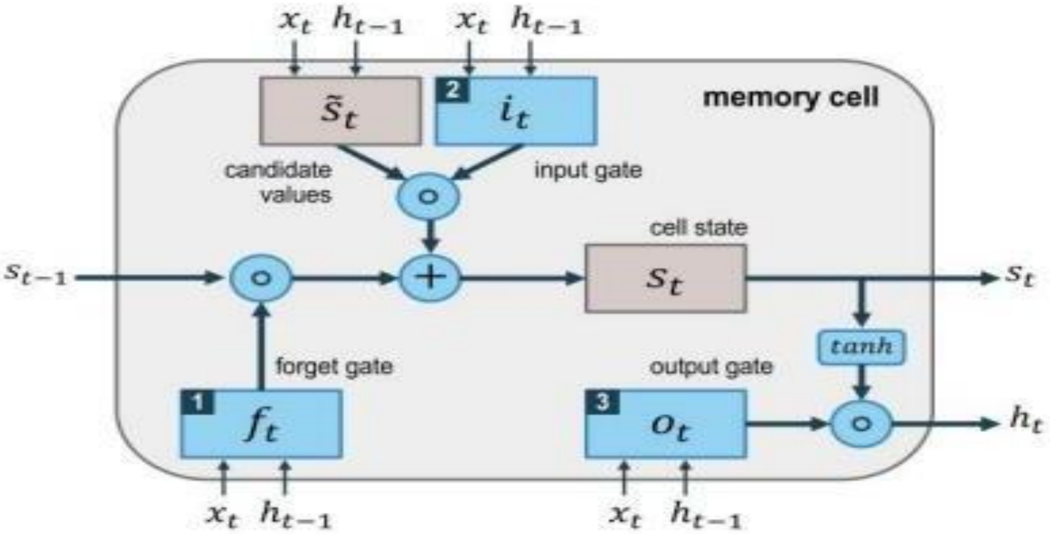

Figure 2 LSTM Model

Recurrent neural networks (RNNs) of the Long Short-Term Memory (LSTM) type have been extensively used to predict stock prices. LSTM can successfully address the vanishing gradient issue that typical RNNs encounter and can capture long-term dependencies in time series data. In [6], Jeenanunta et al. proposed an LSTM-based model for stock price prediction that used past stock prices and technical indicators to forecast the closing price for the following day. The results showed that the LSTM model performed better than conventional machine learning models. Similarly, Roondiwala et al. used past stock prices to train an LSTM model to forecast the closing price for the following day in [13], and the results demonstrated that the LSTM model was capable of making precise stock price predictions. Sunny et al. introduced a deep learning-based stock price prediction model in [14] that used LSTM and Bidirectional LSTM (Bi-LSTM) and achieved better performance than the conventional LSTM model. Chen et al. developed a hybrid deep learning model for stock price prediction in [9] that included an attention mechanism, a Multi-Layer Perceptron (MLP), and a Bidirectional LSTM (Bi-LSTM) and showed better performance than conventional machine learning models. Pratama et al. used an LSTM model in [12] to forecast the Indonesian Composite Stock Price Index by using macroeconomic parameters as inputs, and the results demonstrated that the LSTM model performed better than conventional machine learning models. In general, LSTM models have produced encouraging outcomes in stock price prediction and are popular due to their ability to efficiently address the vanishing gradient problem and capture long-term interdependence.

LSTM FORMULAS:

i_t: Represents input gate, f_t: Represents forgot gate, o_t: Represents output gate, σ: Represents sigmoid function, w_x: weight for the respective gate(x) neurons, h_{t-1}: Output of the previous lstm block, x_t: Input at current timestamp,
b_x: biases for the respective gates(x).

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$
$$f_t = \sigma\left(w_f[h_{t-1}, x_t] + b_f\right)$$
$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

The equations for the cell state, candidate cell state and the final output:

$$\tilde{c}_t = \tanh(w_c[h_{t-1}, x_t] + b_c)$$
$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$
$$h_t = o_t * \tanh(c^t)$$

c_t : cell state at timestamp(t) , c_{t-1}: represents candidate for cell state at timestamp (t).

6.2.1 UML DIAGRAM(S)

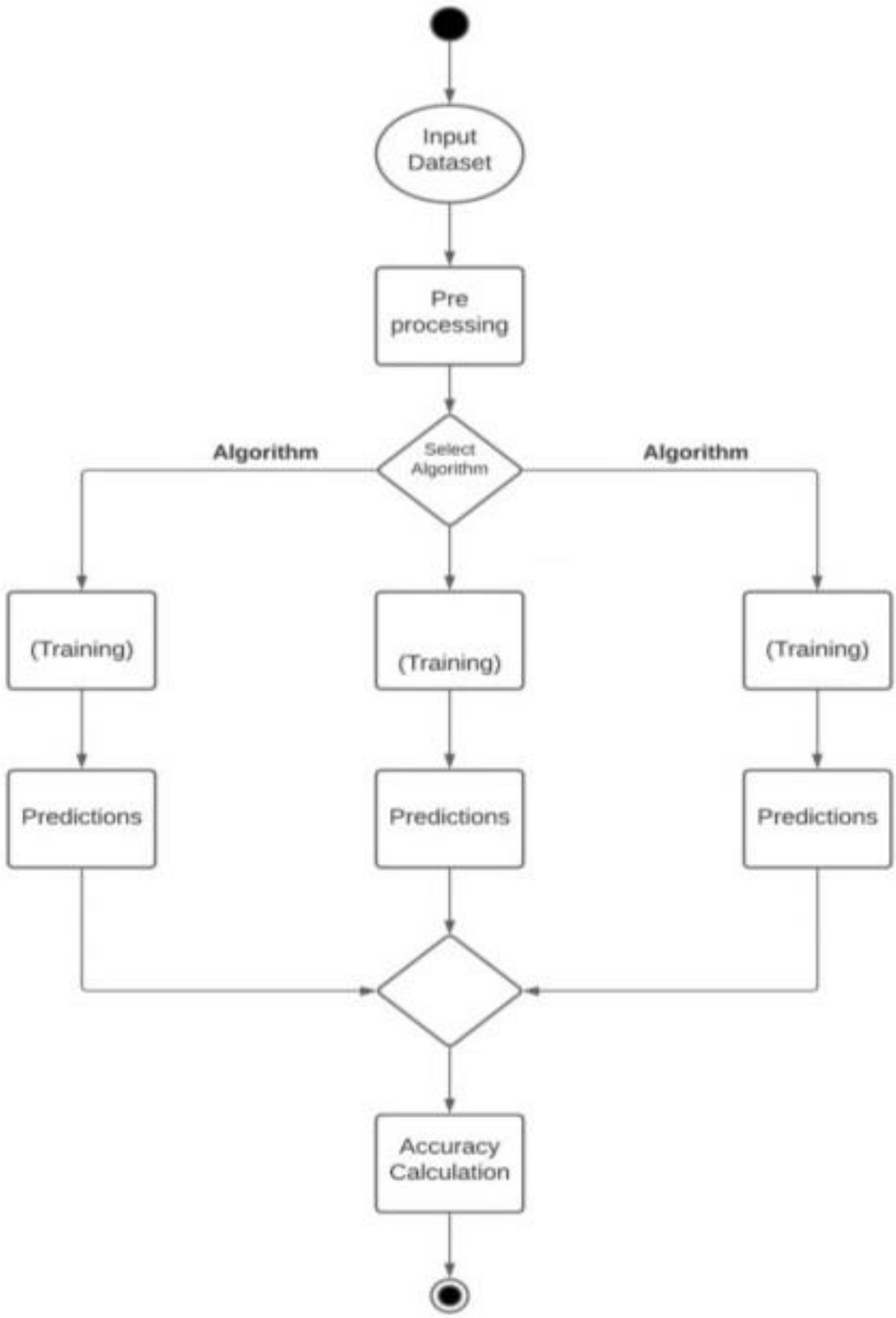

Figure 3 Activity Diagram

Fig. 3 represents how the dataset goes into pre-processing and then into training. Finally, giving the result.

6.2.2 USE CASE DIAGRAM

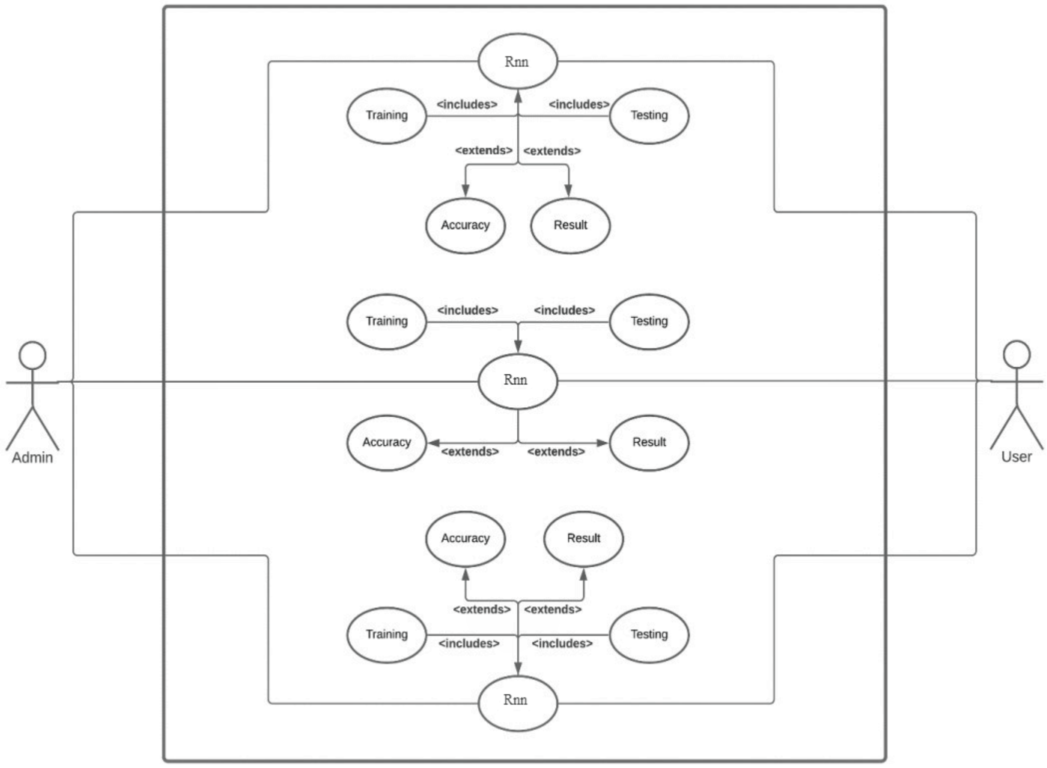

Figure 4 Use case Diagram

6.2.3 BLOCK DIAGRAM

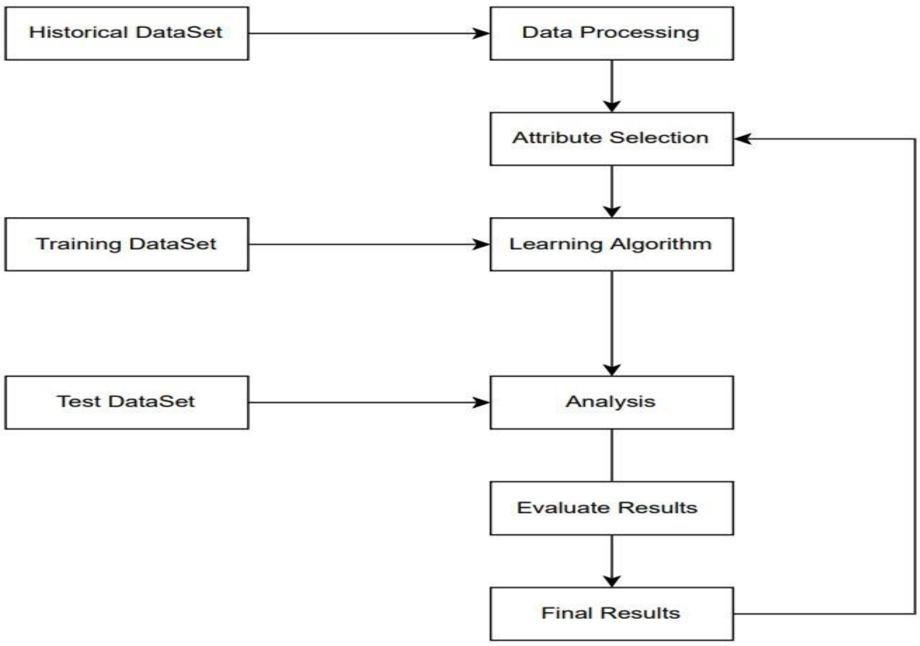

Figure 5 Block Diagram

In Fig. 5 we can observe the entire machine learning process starting from how the data set is given into the model.

6.2.4 SOFTWARE AND HARDWARE USED

Hardware Requirements:

RAM: 4 GB
Storage: 32 GB

Software requirements:

Python 3.5 is used for data pre-processing, model training and prediction.
JavaScript, Html5, CSS for creating the website which will host the entire model.
Any web browser with stable internet connection.

Chapter 7

IMPLEMENTATION

Creating LSTM Model

Libraries used for the implementation are as follows:

Pandas:

- Data manipulation and dissection is quick and efficient.
- It is possible to load data from different file objects.
- Both Data Frame and advanced dimensional objects allow for the fitting and deletion of size mutability columns
- Including and joining a data collection.
- Pivoting and flexible reconfiguration of data collections.
- Provides capability for time series.
- Important group by features for splitting, applying, and combining data sets

Keras:

- User-Friendly and Fast Deployment
- Quality Documentation and Large Community Support
- Pretrained models
- Multiple Backend and Modularity

Streamlit:

- Web framework
- Data visualization
- Python-based
- Interactive apps
- Machine learning

Yahoo Finance:

- Financial data
- Stock market
- Investment tools
- Company profiles
- Historical prices
- Real-time quotes
- Customizable watchlists
- Portfolio tracking

Chapter 8

RESULTS

The following graphs represent the losses that occur when the model is training.

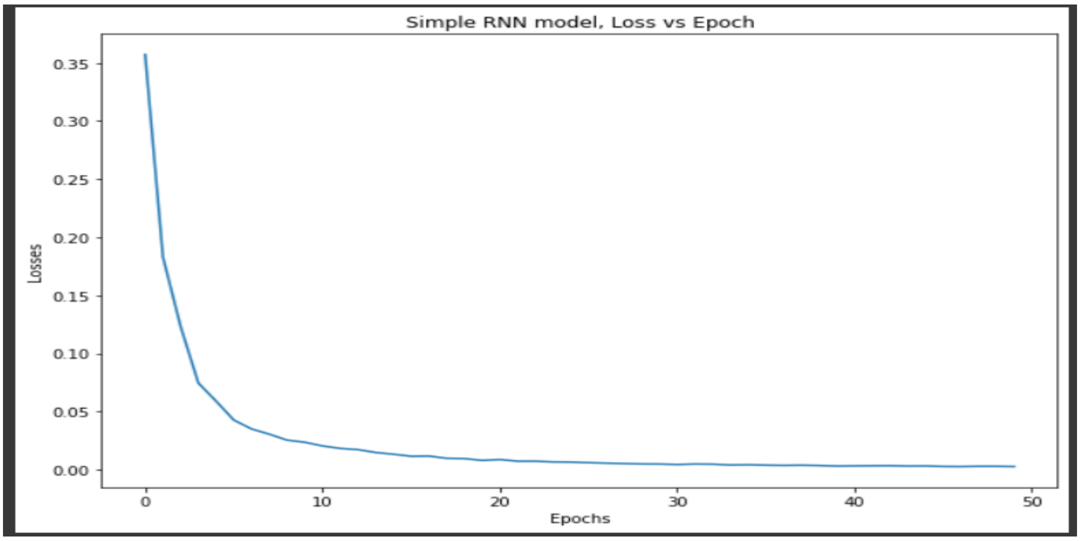

Figure 6 Loss vs Epoch

Fig. 6 graph plots the loss with respect to each epoch.

The resulting loss vs epoch graph shows the training loss and validation loss over the course of training. The training loss represents the value of the loss function on the training data, while the validation loss represents the value of the loss function on a separate validation dataset that is not used for training.
Typically, we want to see the training loss decrease over time as the model learns to fit the training data.

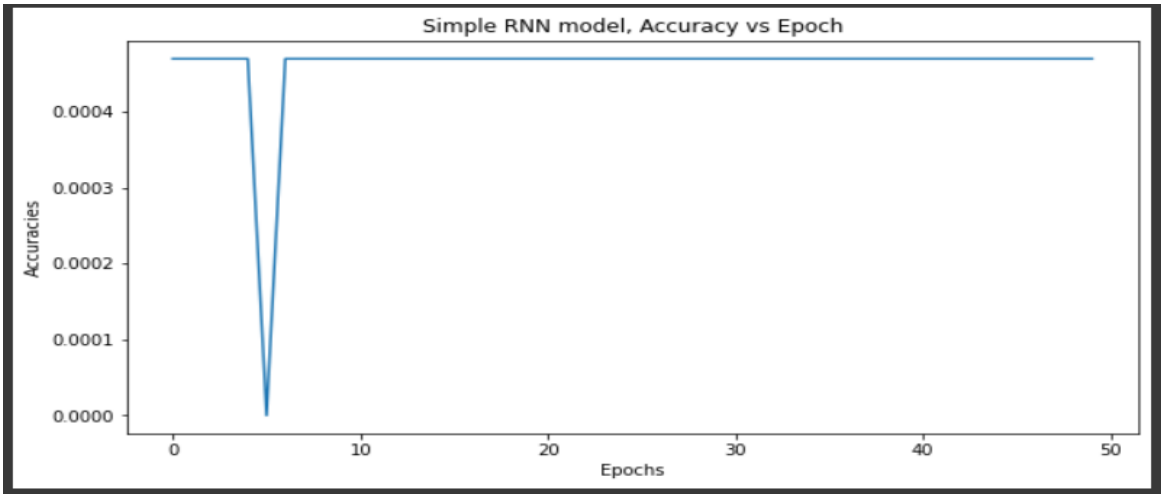

Figure 7 Accuracy vs Epoch

Fig. 7 graph plots the accuracy of the system with each epoch.

Typically, the x-axis represents the number of epochs (or iterations) that the model has been trained for, while the y-axis represents the accuracy of the model on a validation dataset. As the model is trained, the accuracy on the validation dataset will typically improve, although there may be fluctuations in accuracy from one epoch to the next.

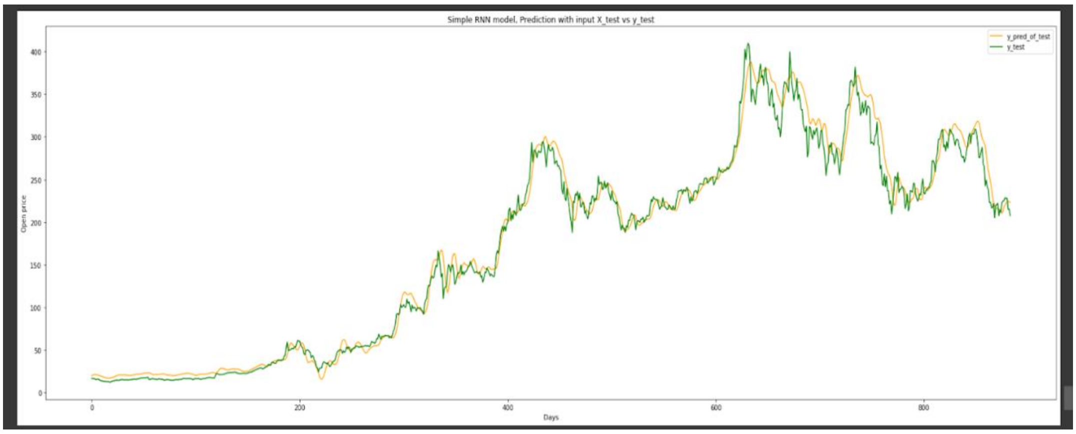

Figure 8 X-test vs Y-test

Fig. 8 graph shows the comparison between the test closing price(green) and the predicted test closing price(yellow).

If the yellow line deviates significantly from the green line, this may indicate that the model is not accurately capturing the patterns and trends in the data, and may require further tuning or adjustment. By comparing the predicted and actual closing prices in this way, we can get a sense of how well our simple RNN model is performing on the test data, and make decisions about how to improve its accuracy if necessary.

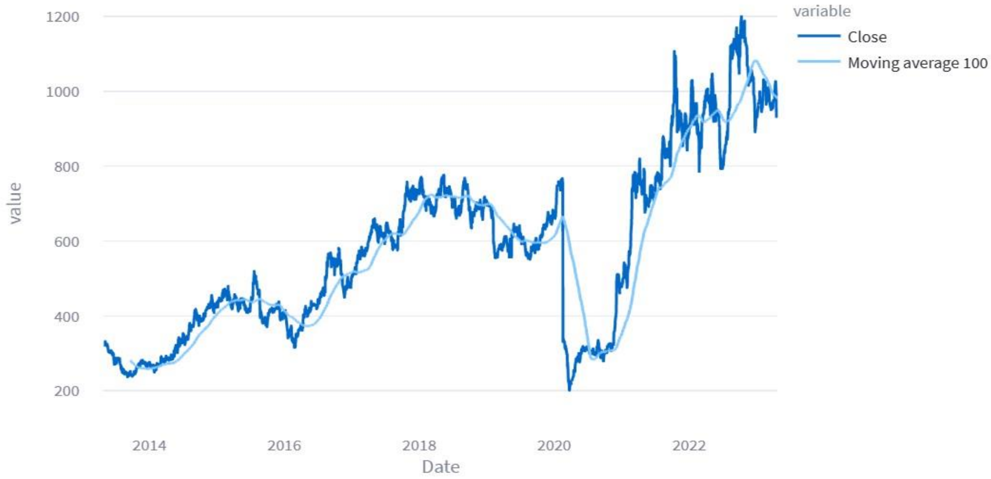

Figure 9 100-day Moving Average vs Closing

Fig. 9 is the 100-day Moving Average vs Closing represented using plotly.js on a StockX

By comparing the blue and dark blue lines, we can get a sense of how the actual closing prices are trending compared to the smoothed-out trend represented by the 100-day moving average. If the blue line is consistently above the dark blue line, this may indicate that the asset is in an uptrend, while if the blue line is consistently below the dark blue line, this may indicate that the asset is in a downtrend.

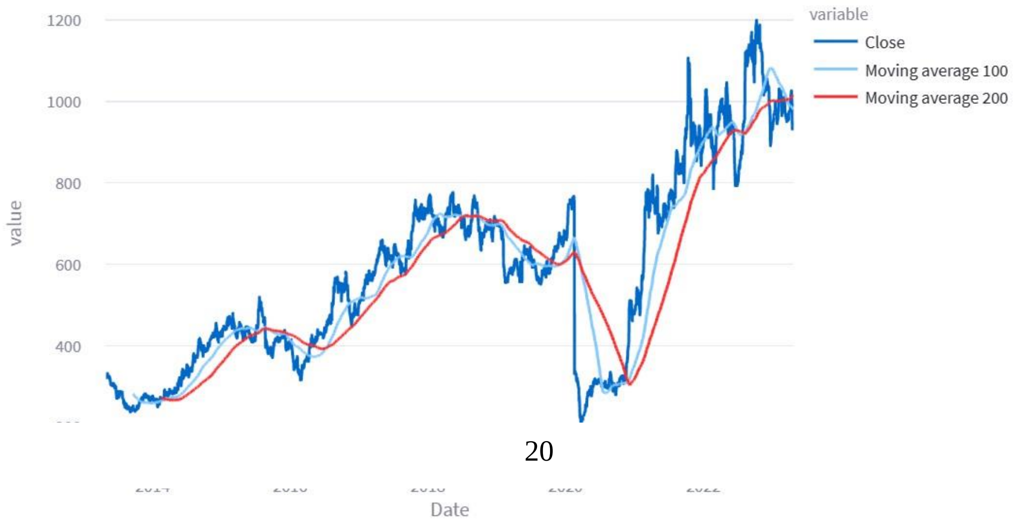

Figure 10 200-day Moving Average vs Closing

Fig. 10 is the 200-day Moving Average vs Closing represented using plotly.js on StockX.

By comparing the blue and red lines, we can get a sense of how the actual closing prices are trending compared to the smoothed-out trend represented by the 200-day moving average. If the blue line is consistently above the red line, this may indicate that the asset is in an uptrend, while if the blue line is consistently below the red line, this may indicate that the asset is in a downtrend.

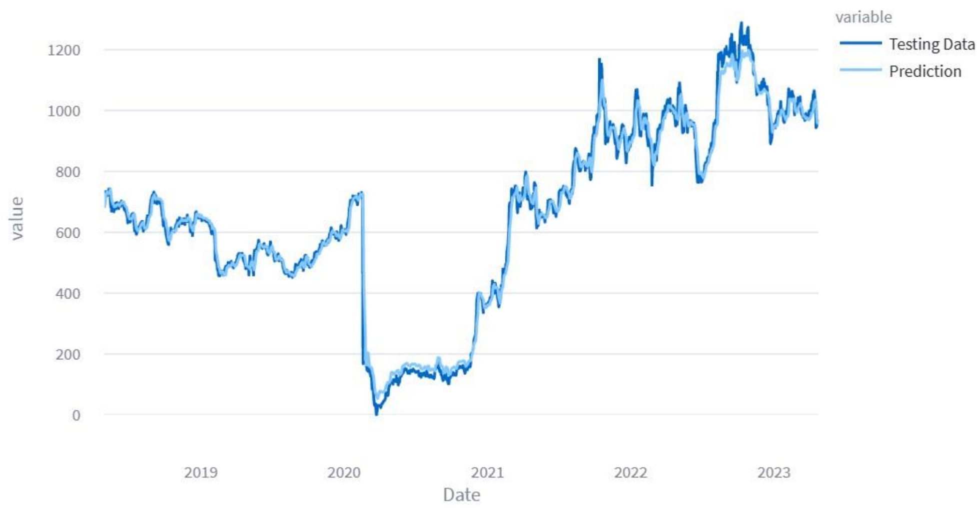

Figure 11 Testing data vs Prediction data

Fig 12 is the testing data vs prediction data represented using plotly.js on StockX.
By comparing the blue and dark blue lines, we can get a sense of how well the model is predicting the testing data. Ideally, we would like to see the dark blue line closely following the blue line, indicating that the model is accurately predicting the trend and fluctuations of the testing data.

After running the model for 10 epochs the following are the results of the optimizers:

Table 2: Optimizer and Loss

Optimizer	Loss
Adam	0.0032
SGD	0.0029
RMSprop	0.0026
Adadelta	0.0024
Adagrad	0.0024
Adamax	0.0023
Nadam	0.0028
Ftrl	0.1173

Adam: Based on estimations for the first and second moments of the gradients, the Adam optimizer determines an adaptive learning rate for each parameter. It records both the exponentially decaying average of the squared gradients in the past and the average of the historical gradients.

SGD: SGD (Stochastic Gradient Descent) is a type of optimization algorithm commonly used in machine learning and deep learning to optimize the parameters of a model during training.

RMSprop: RMSprop (Root Mean Square Propagation) is a gradient descent optimization algorithm used for neural networks. It is similar to the gradient descent and Adagrad algorithms, but it uses a moving average of squared gradients to scale the learning rate.

Adamax: A variation of the Adam optimizer called Adamax is made to work well on gradient problems that have sparse or noisy gradients. While it is similar to Adam, it scales the step size using the L-infinity norm rather than the gradients' L2 norm.

Nadam: Adam and Nesterov momentum are variations of the optimizer algorithm Nadam. Similar to Adam, Nadam calculates adaptive learning rates for each parameter, but it also takes into account Nesterov momentum, which in some situations helps to speed up the learning process.

Adam was choosen since for 100 epochs the loss was the most minimum compared to other optimizers.

For Optimizer ‘Adam’ the different activation functions used are as follows:

Table 3: Activation Function and Loss

Activation Function	Loss
ReLU	0.0313
Sigmoid	0.0111
Tanh	0.0197
Softmax	0.4760

ReLU:
(Rectified Linear Unit) is an activation function commonly used in neural networks. It is defined as:

$$f(x) = \max(0, x)$$

In other words, if the input value (x) is positive, the output will be equal to the input value (f(x) = x). If the input value is negative, the output will be zero (f(x) = 0).

Sigmoid:
The sigmoid function is a commonly used activation function in neural networks. It is a mathematical function that maps any input value to a value between 0 and 1. The formula for the sigmoid function is:

$$f(x) = \frac{1}{1 + e^{-x}}$$

where x is the input value.

Tanh:
Tanh, short for "hyperbolic tangent", is a commonly used activation function in neural networks. It is a smooth, S-shaped function that is symmetric around the origin.

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

where e is Euler's number and x is the input to the function.

Softmax:
Softmax is an activation function that is commonly used in the output layer of neural networks for multi-class classification problems. The Softmax function converts a vector of real values to a probability distribution that sums up to 1.0. The formula for the Softmax function is as follows:

$$Softmax(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

The output of the Softmax function is a vector of probabilities where each element represents the predicted probability of the corresponding class.

Chapter 9

CONCLUSION

Recurrent Neural Networks (RNNs), a type of neural network that can account for the sequential structure of data, are ideal for a variety of applications, including time series analysis, speech recognition, and natural language processing. They can handle input sequences of varying length because they can model time dependencies using feedback loops within the network. RNNs come in a variety of forms, such as the fundamental RNN, LSTM, and gated recurrent units. (GRUs). These modifications enhance the fundamental RNN architecture by tackling the vanishing gradient issue, which can have an impact on how deep networks are trained.

Recent years have seen a rise in the use of Long Short-Term Memory (LSTM), a type of Recurrent Neural Network (RNN), for market price forecasting. The temporal and long-term dependencies present in finance time series data have been successfully captured by LSTMs. In order to shed light on the advantages and disadvantages of this strategy, this paper aims to provide an overview of current studies on LSTM-based models for stock price prediction. The majority of the studies that used LSTMs for stock price forecasts outperformed conventional statistical models. The ability of LSTMs to manage noisy and non-linear data is another benefit. For stock price prediction in the presence of noise and non-linearities, LSTMs have been used in a number of experiments.

LSTMs have some drawbacks that must be taken into account when using them to forecast stock prices despite their benefits. The inability to easily understand the findings is one drawback. LSTMs are frequently referred to as "black box" models, which makes it challenging to understand how they make forecasts. The sensitivity of LSTMs to hyperparameter choices is another drawback. The number of layers, the number of neurons in each layer, the learning rate, and the dropout rate are examples of hyperparameters, which are parameters specified before the model is trained. The selection of hyperparameters can have a significant impact on how well an LSTM model performs.

Chapter 10

REFERENCES

[1] R. Zhang, Z. Wu and S. Wang, "Prediction of Stock Based on Convolution Neural Network," 2020 Chinese Control And Decision Conference (CCDC), 2020, pp. 3175-3178, doi: 10.1109/CCDC49329.2020.9164222.

[2] L. Sayavong, Z. Wu and S. Chalita, "Research on Stock Price Prediction Method Based on Convolutional Neural Network," 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), 2019, pp. 173-176, doi: 10.1109/ICVRIS.2019.00050.

[3] R. Zhang, Z. Yuan and X. Shao, "A New Combined CNN-RNN Model for Sector Stock Price Analysis," 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), 2018, pp. 546-551, doi: 10.1109/COMPSAC.2018.10292.

[4] J. Zhao, D. Zeng, S. Liang, H. Kang, and Q. Liu, “Prediction model for stock price trend based on recurrent neural network - journal of ambient intelligence and humanized computing,” *SpringerLink*, 12-May-2020. [Online]. Available: https://link.springer.com/article/10.1007/s12652-020-02057-0. [Accessed: 06-Nov-2022].

[5] I. Jahan, S. Z. Sajal and K. E. Nygard, "Prediction Model Using Recurrent Neural Networks," 2019 IEEE International Conference on Electro Information Technology (EIT), 2019, pp. 1-6, doi: 10.1109/EIT.2019.8834336.

[6] C. Jeenanunta, R. Chaysiri and L. Thong, "Stock Price Prediction With Long Short-Term Memory Recurrent Neural Network," 2018 International Conference on Embedded Systems and Intelligent Technology & International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES), 2018, pp. 1-7, doi: 10.1109/ICESIT-ICICTES.2018.8442069.

[7] C. Peng, Z. Yin, X. Wei and A. Zhu, "Stock Price Prediction based on Recurrent Neural Network with Long Short-Term Memory Units," 2019 International Conference on Engineering, Science, and Industrial Applications (ICESI), 2019, pp. 1-5, doi: 10.1109/ICESI.2019.8863005.

[8] X.-K. Xie, H. Wang, and Y. Du, “Recurrent neural network for forecasting stock market trend,” *World Scientific*, 2016. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/9789813200449_0049. [Accessed: 06-Nov-2022].

[9] Q. Chen, W. Zhang and Y. Lou, "Forecasting Stock Prices Using a Hybrid Deep Learning Model Integrating Attention Mechanism, Multi-Layer Perceptron, and Bidirectional Long-Short Term Memory Neural Network," in IEEE Access, vol. 8, pp. 117365-117376, 2020, doi: 10.1109/ACCESS.2020.3004284.

[10] R. d. A. Araújo, A. L. I. Oliveira and S. Meira, "On the high-frequency financial forecasting problem with multilayer perceptron models," 2015 Latin America Congress on Computational Intelligence (LA-CCI), 2015, pp. 1-6, doi: 10.1109/LA-CCI.2015.7435935.

[11] S. Mabu, M. Obayashi and T. Kuremoto, "Ensemble learning of rule-based evolutionary algorithm using multi layer perceptron for stock trading models," 2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS), 2014, pp. 624-629, doi: 10.1109/SCIS-ISIS.2014.7044647.

[12] M. Nizar Yoga Pratama and Ernastuti, "Multilayer Perceptron and Long Short-Term Memory for Predicting Indonesian Composite Stock Price Index Using Macroeconomic Factors," 2019 Fourth International Conference on Informatics and Computing (ICIC), 2019, pp. 1-5, doi: 10.1109/ICIC47613.2019.8985785.

[13] M. R. Roondiwala, H. Patel, and S. Varma, “Predicting stock prices using LSTM - ijsr.net,” *www.ijsr.net*, 2015. [Online]. Available: https://www.ijsr.net/archive/v6i4/ART20172755.pdf. [Accessed: 06-Nov-2022].

[14] M. A. Istiake Sunny, M. M. S. Maswood and A. G. Alharbi, "Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model," 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), 2020, pp. 87-92, doi: 10.1109/NILES50944.2020.9257950.S.

[15] 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), 2019, pp. 1-5, doi: 10.1109/IMITEC45504.2019.9015840.

[16] D. Wei, "Prediction of Stock Price Based on LSTM Neural Network," 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), 2019, pp. 544-547, doi: 10.1109/AIAM48774.2019.00113.

[17] A. Ghosh, S. Bose, G. Maji, N. C Debnath, and S. Sen, “Stock price prediction using LSTM on Indian share market,” *ResearhGate*, Sep-2019. [Online]. Available: https://www.researchgate.net/profile/Giridhar-Maji/publication/346450872_Stock_Price_Prediction_Using_LSTM_on_Indian_Share_Market/links/5fc3142b458515b797843e38/Stock-Price-Prediction-Using-LSTM-on-Indian-Share-Market.pdf. [Accessed: 06-Nov-2022].

[18] R. Gupta and M. Chen, "Sentiment Analysis for Stock Price Prediction," 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), 2020, pp. 213-218, doi: 10.1109/MIPR49039.2020.00051.

[19] B. N. Varaprasad, C. Kundan Kanth, G. Jeevan and Y. K. Chakravarti, "Stock Price Prediction using Machine Learning," 2022 International Conference on Electronics and Renewable Systems (ICEARS), 2022, pp. 1309-1313, doi:DOI.1109/ICEARS53579.2022.9752248.

[20] A. Maiti and P. Shetty D, "Indian Stock Market Prediction using Deep Learning," 2020 IEEE REGION 10 CONFERENCE (TENCON), 2020, pp. 1215-1220, doi:DOI.1109/TENCON50793.2020.9293712.

[21] R. Zhang, Z. Wu and S. Wang, "Prediction of Stock Based on Convolution Neural Network," 2020 Chinese Control And Decision Conference (CCDC), 2020, pp. 3175-3178, doi:DOI.1109/CCDC49329.2020.9164222.

[22] L. Sayavong, Z. Wu and S. Chalita, "Research on Stock Price Prediction Method Based on Convolutional Neural Network," 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), 2019, pp. 173-176, DOI: 10.1109/ICVRIS.2019.00050.