iJRASET
International Journal For Research in Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 10 **Issue:** IV **Month of publication:** April 2022

**DOI:** https://doi.org/10.22214/ijraset.2022.42039

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 10 Issue IV Apr 2022- Available at www.ijraset.com*

# Stock Market Prediction Using LSTM

Shaikh Shoieb Abubaker[1], Syed Rouf Farid[2]
[1, 2]*Computer Engineering, Keystone School of Engineering, Pune, Maharashtra ,India*

***Abstract: Different machine learning algorithms are discussed in this literature review. These algorithms can be used for predicting the stock market. The prediction of the stock market is one of the challenging tasks that must have to be handled.In this paper, it is discussed how the machine learning algorithms can be used for predicting the stock value. Different attributes are identified that can be used for training the algorithm for this purpose. Some of the other factors are also discussed that can have an effect on the stock value.***
***Keywords: Machine learning, stock market prediction, literature review, artificial neural network, support vector machine, genetic algorithm, investment decision, RNN, LSTM.***

## I. INTRODUCTION

The stock market is the place where the buyers and sellers come for the purpose of buying the stocks and selling the stocks of a company. With the passage of time, more and more people are getting interested in the stock market. The
increasing interest of the people makes this topic more important for the research. Machine learning provides different predictive algorithms that can be very helpful in this domain. Different sectors are using the machine learning
for their improvement. The use of machine learning enables the computers to learn without the involvement of any external program.Recurrent neural networks (RNN) have proved one of the most powerful models for processing sequential data.Long Short-Term memory is one of the most successful RNNs architectures. LSTM
RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularised by many people in following work.1 They work tremendously well on a large variety of problems, and are now widely used. The paper that we have presented modelled and predicted the stock returns of Apple and 5 other major tech companies using LSTM. We are in the ever growing generation on financial literacy and the never ending scope on the stock market. Although people have been using pen paper and the traditional mathematical manner to predict stock movement. So after looking at the other few models and methods available we decided to take a dive into this field and have a solution that can predict the movement of stocks for an individual .

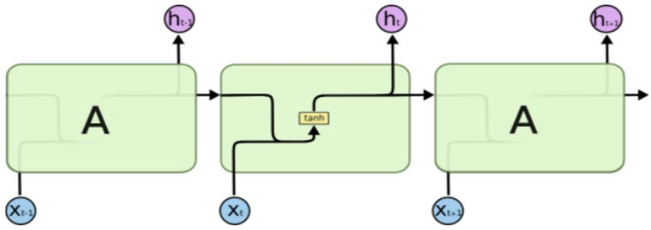
## II. METHODOLOGY

### *A. Proposed Systems*

The prediction methods can be mainly categorised into two ,statistical methods and artificial intelligence methods. Statistical methods consist of logistic regression model, ARCH model, etc. Artificial intelligence methods includes multi-layer
perceptron, convolutional neural network, Back propagation network,single-layer LSTM, recurrent neural network, etc. They used a Long short-term memory network (LSTM).

### *B. Long Short Term Memory*

Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularised by many people in following work.1 They work tremendously well on a large variety of problems, and are now widely used.



The repeating module in a standard RNN contains a single layer.

©IJRASET: All Rights are Reserved 

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 10 Issue IV Apr 2022- Available at www.ijraset.com*

### *C. Working of LSTM*

LSTM is a special network that has three gates in its structures. these gates are placed in an LSTM unit, called as input gate, forgetting gate and output gate.AS the information enters the LSTM's network, it can be selected by rules defined . Only the information that follows the algorithm will be left, and the information that does not follow the algorithm will be forgotten through the forgetting gate. The experimental data in this paper are the actual historical data downloaded from the Internet. Three data sets were used in the experiments. It is needed to find an optimization algorithm that requires less resources and has faster convergence speed. The System that we propose has several steps that lead to model generation . The steps involved are as follows .

1) *Stage 1:* Raw Data: In this stage, the historical stock data is collected from the internet and this historical data is used for the prediction of future stock prices.
2) *Stage 2:* Data Preprocessing: The pre-processing stage involves

*a)* Data discretization: Part of data reduction but with particular importance, especially for numerical data

*b)* Data transformation: Normalisation.

*c)* Data cleaning: Fill in missing values.

*d)* Data integration: Integration of data files. After the dataset is transformed into a clean dataset, the dataset is divided into training and testing sets so as to evaluate. Here, the training values are taken as the more recent values. Testing data is kept as 5-10 percent of the total dataset

3) *Stage 3:* Training Neural Network: In this stage, the data is fed to the neural network and trained the prediction is done by assigning random biases and weights. Our LSTM model is composed of a sequential input layer followed by 2 LSTM layers and 4 dense layer with ReLU activation and then finally a dense output layer.
4) *Stage 5:* Output Generation: In this layer, the output value generated by the output layer is compared to the target value from the dataset . The error or the difference between the target and the obtained output value is minimized by using back propagation algorithm which adjusts the weights and the biases of the network

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 5, 200)            161600
_________________________________________________________________
lstm_1 (LSTM)                (None, 200)               320800
_________________________________________________________________
dense (Dense)                (None, 200)               40200
_________________________________________________________________
dense_1 (Dense)              (None, 100)               20100
_________________________________________________________________
dense_2 (Dense)              (None, 50)                5050
_________________________________________________________________
dense_3 (Dense)              (None, 5)                 255
=================================================================
Total params: 548,005
Trainable params: 548,005
Non-trainable params: 0
_________________________________________________________________
```

### *D. Advantages of LSTM*

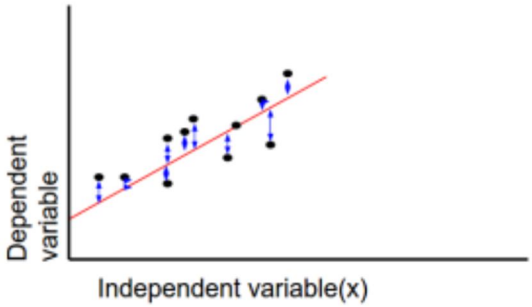
The advantage of the Long Short-Term Memory (LSTM) network over other recurrent networks back in 1997 came from an improved method of back propagating the error. Hochreiter and Schmidhuber called it "constant error back propagation"
One of the main advantages of LSTM is its capacity to read intermediary context. Each unit of cll remembers data for a long / short period without unambiguously utilising the activation function.
An important fact is that any cell state is repeated only with the release of the forget gate, which varies between range of 0 to 1. That's to say, the gateway for forgetting in the LSTM cell is responsible for both the hardware and the function of the cell state activation. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs.

### *E. Analysis*

In our system for analysis we have used the root mean squared error method . The error of the difference between the target and predicted value is minimized by the RMSE value . The use of RMSE is highly common. It makes an excellent general purpose error metric . Compared to the similar Mean Absolute Error, RMSE amplifies and severely punishes large errors.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum\left(\hat{Y}_i - Y_i\right)^2}$$

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |




## III. DATASET

The dataset that we have used is of 5 tech giants in the NASDAQ market . The dataset used are of Apple ,Amazon , Google , Microsoft and Tesla .The different columns or as we refer to them as features are as followed

```
In [10]: 1 AAPL.head() # Checking the data
```

Out[10]:

| | Date | Open | High | Low | Close | Adj. Close | Volume |
|---|---|---|---|---|---|---|---|
| 0 | 2015-01-02 | 27.85 | 27.86 | 26.84 | 27.33 | 24.86 | 212818400.0 |
| 1 | 2015-01-05 | 27.07 | 27.16 | 26.35 | 26.56 | 24.16 | 257142000.0 |
| 2 | 2015-01-06 | 26.64 | 26.86 | 26.16 | 26.57 | 24.16 | 263188400.0 |
| 3 | 2015-01-07 | 26.80 | 27.05 | 26.67 | 26.94 | 24.50 | 160423600.0 |
| 4 | 2015-01-08 | 27.31 | 28.04 | 27.17 | 27.97 | 25.44 | 237458000.0 |

Here the values are as followed :-

1) *Date:* Date of which prices are listed.
2) *Open:* Opening price of stock on that date .
3) *High:* Highest value of the stock on that date.
4) *Low:* Lowest prices of the stock on that date .
5) *Close:* The closing price of the stock
6) *Adj, Close:* Adjusted close is the price of the stock after dividends and other related manipulation .
7) *Volume:* Volume of stock traded on that day .

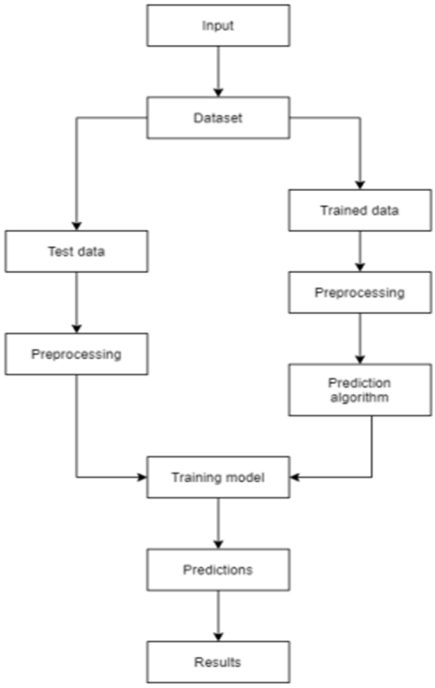
The main model was created using the dataset for Apple

## IV. PROPOSED SYSTEM

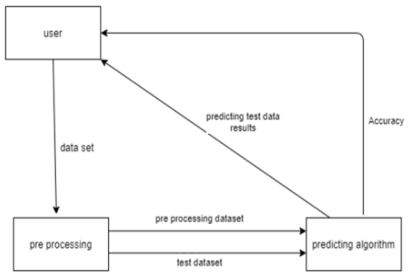
### A. *System Components*



*B. System Chart*




*C. Collaboration Diagram*




## V. EDA AND DATA PRE-PROCESSING

*A. Data Cleaning*

The dataset is not in the right format to use directly with the model hence data preprocessing is required . In our model we did a few steps as followed :

1. Dropping the null rows

```
AAPL.dropna(inplace = True)
```

2. Changing the data types of the column / features .

```
AAPL["Date"] = pd.to_datetime(AAPL["Date"])
AAPL["Volume"] = AAPL["Volume"].str.replace(',', '')
AAPL = AAPL.astype({"Open": float, "Volume": float})
```

3. Sorting the data by date.

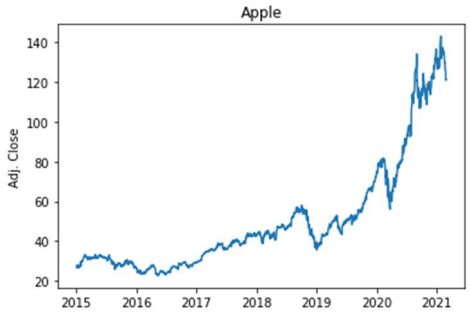
```
AAPL = AAPL.sort_values(by = 'Date', ignore_index = True)
```

4. Dropping older data

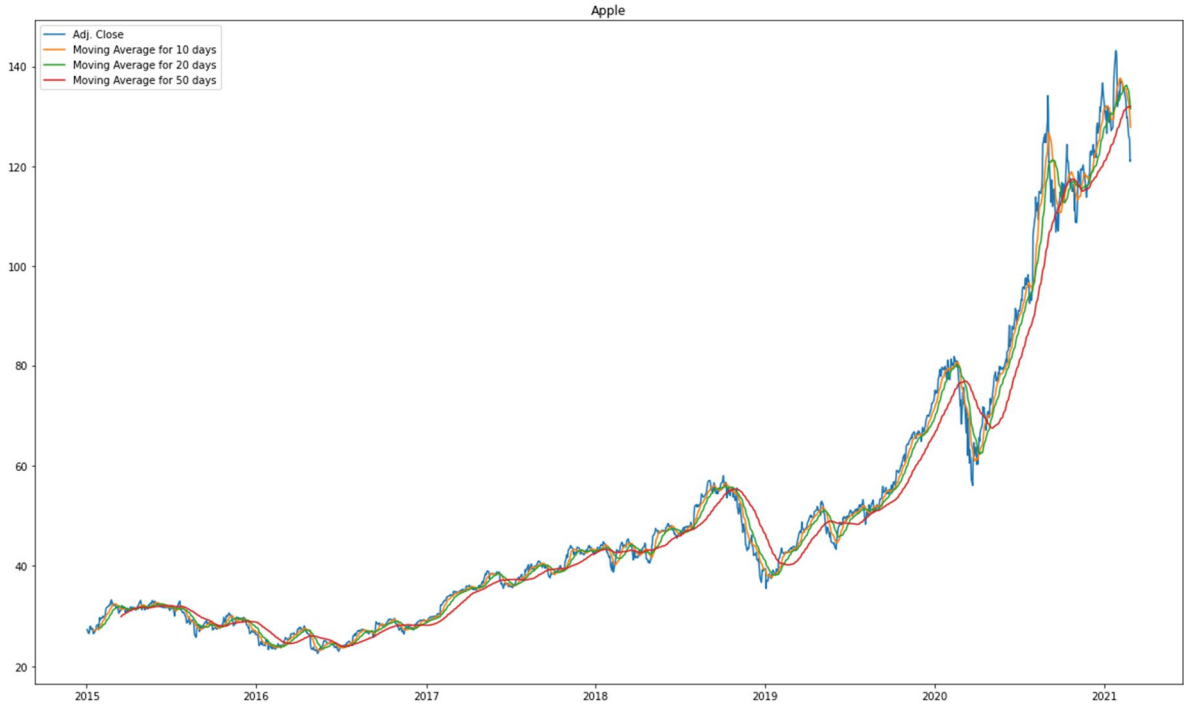
```
AAPL = AAPL[AAPL["Date"] >= '2015-01-01'].reset_index(drop = True)
```

*B. EDA*

We plotted the closing value of the stock to check the trends.




Moving Average for 10 , 20 and 50 days :




## VI. DEFINING CUSTOM LEARNING RATE

*A. Code*

We created a scheduler function that takes in the number of epochs and sets a custom learning rate and then we passed on the Learning rate scheduler function of keras via the tensorflow module . The code for the scheduler is as follows .

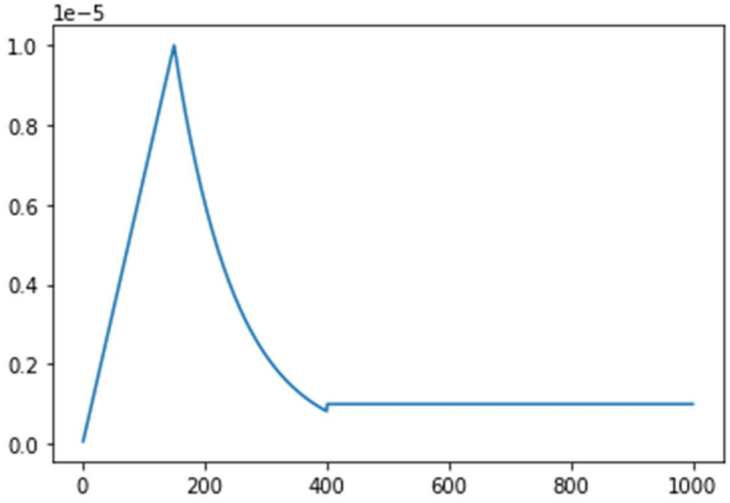
```
def scheduler(epoch):

  if epoch <= 150:
    lrate = (10 ** -5) * (epoch / 150)
  elif epoch <= 400:
    initial_lrate = (10 ** -5)
    k = 0.01
    lrate = initial_lrate * math.exp(-k * (epoch - 150))
  else:
    lrate = (10 ** -6)

  return lrate
```

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 10 Issue IV Apr 2022- Available at www.ijraset.com*

*B. Plotting the Learning Rate*




## VII. MODEL CREATION

*A. Splitting the dataset*

For model creation we split the data for training and testing based on the period / date the stock price was recorded . We defined a function that splits the dataset . The function divided the dataset in training and testing and further into the training dependent and independent variables. We then split the data based on time as this is a time series based problem .

```
# Change Dtype of Date column
AAPL["Date"] = pd.to_datetime(AAPL["Date"])
```

```
#Spliting  the Data into Training and Test set Training Period: 2015-01-02 - 2020-09-30
```

```
AAPL_Date = '2020-10-01'
AAPL_Train_X, AAPL_Train_Y, AAPL_Test_X, AAPL_Test_Y = Dataset(AAPL, AAPL_Date)
```

*B. Fitting the Model*

The model that we created was compiled using the Adam optimizer that we imported from Keras . The loss function used for mse and the metric for evaluation that we used is RMSE as defined above that was also used from the keras metric function of keras .

```
def Model():
  model = tf.keras.models.Sequential([
                tf.keras.layers.LSTM(200, input_shape = (5, 1), activation = tf.nn.leaky_relu, return_sequences = True),
                tf.keras.layers.LSTM(200, activation = tf.nn.leaky_relu),
                tf.keras.layers.Dense(200, activation = tf.nn.leaky_relu),
                tf.keras.layers.Dense(100, activation = tf.nn.leaky_relu),
                tf.keras.layers.Dense(50, activation = tf.nn.leaky_relu),
                tf.keras.layers.Dense(5, activation = tf.nn.leaky_relu)
                ])
  return model
```

```
model = Model() # Creating object of the model
```

```
AAPL_Model = Model()
```

```
AAPL_Model.compile(optimizer = tf.keras.optimizers.Adam(), loss = 'mse', metrics = tf.keras.metrics.RootMeanSquaredError())
```

```
APL_Model.fit(AAPL_Train_X, AAPL_Train_Y, epochs = 1000, validation_data = (AAPL_Test_X, AAPL_Test_Y), callbacks=[callback])
```

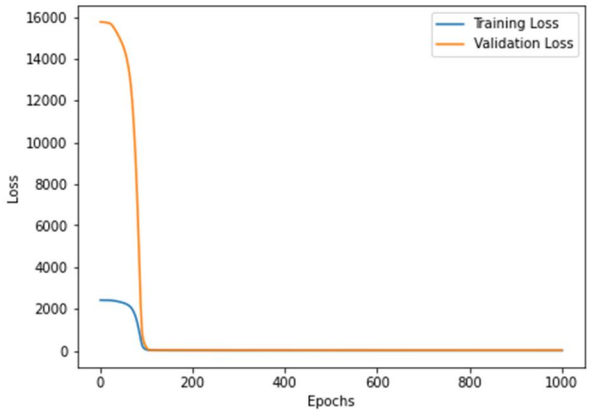
We fit the model using the training split of the dataset by passing the dependent and the independent variables . We passed the testing dataset in the validation dataset and passed the callback function that we made defining the learning rate scheduler .

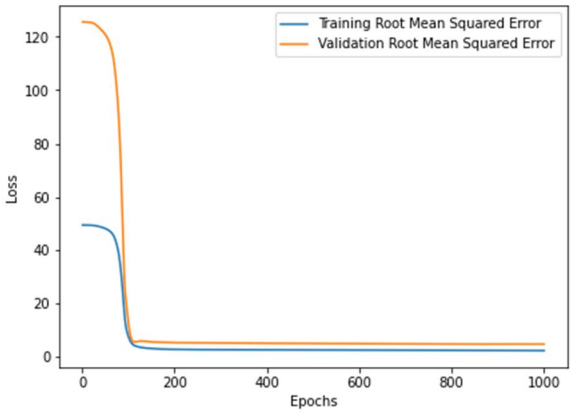
©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

## VII. EXPERIMENTAL ANALYSIS

### A. *Loss and Error*

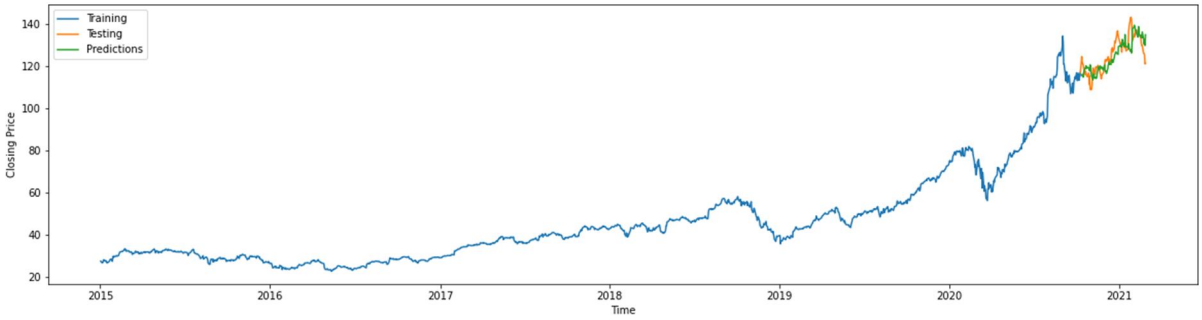
We recorded the loss and error that we realised came down as the number of epoch kept increasing . The loss and error is as follows for the model we created .




### B. *Prediction Of The Future Price Using Testing Dataset*

We used the the predict function of the model that was created compiled and fitted using the training dataset and we passed on the testing dataset to find the prediction . The predicted and the target value was plotted and the results are as shown below.




## VIII. CONCLUSION

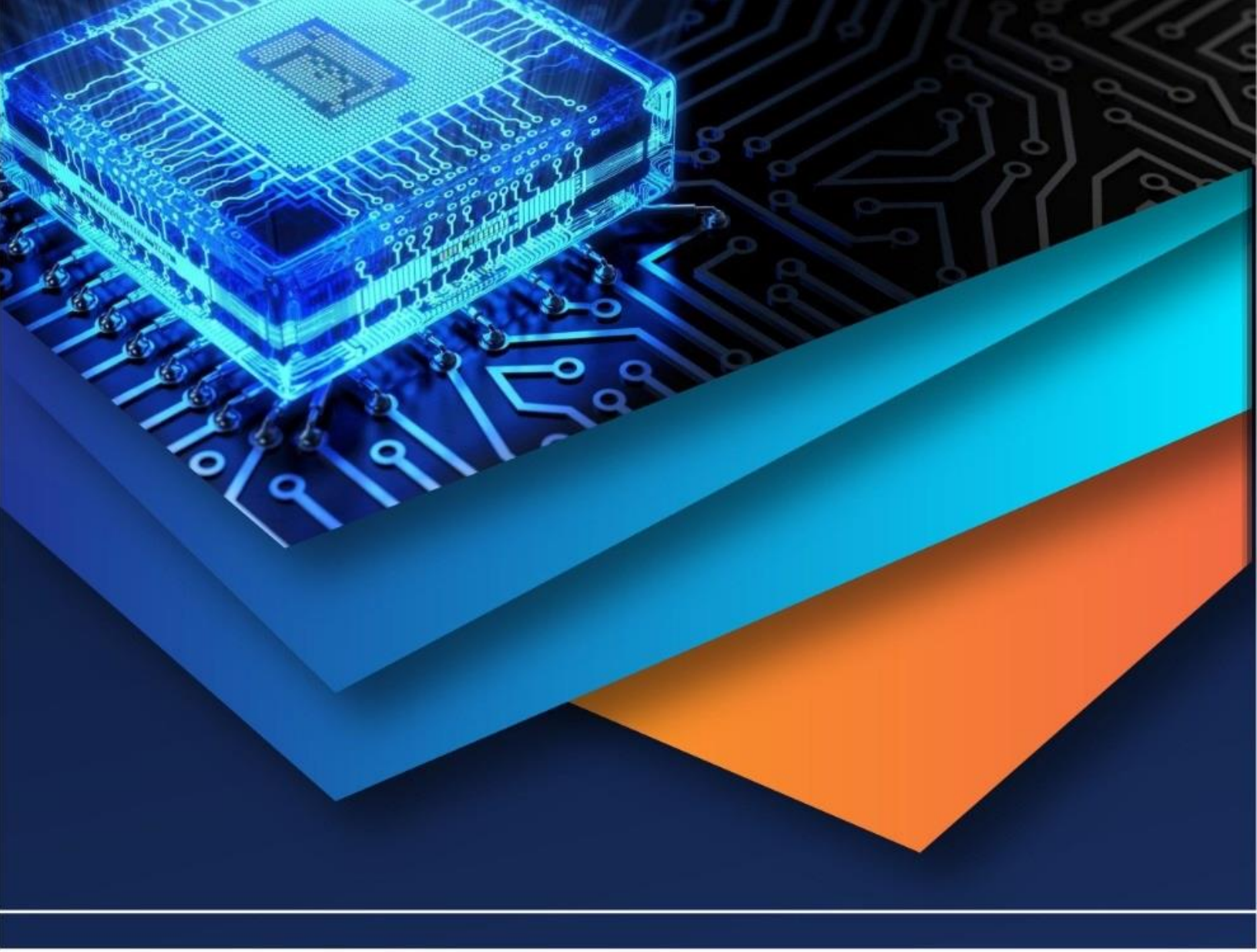
The popularity of stock market trading is growing rapidly between the domestic and international marker , which is encouraging researchers to find new methods for prediction using new techniques and using more and more of the historical dataset with time as it goes on increasing .

The forecasting technique is not only helping the researchers but it also helps investors and any individual dealing with the stock market. In order to help predict the stock indices, a forecasting model with good accuracy is required.

In our work, we have used one of the most precise forecasting technologies using Recurrent Neural Network and LSTM unit which will help investors, analysts or any person interested in investing in the stock market by providing them a good knowledge of the future situation of the stock market of domestic and international markets .

## REFERENCES

[1] Prediction of Stock Market Using Recurrent Neural Network - Sadman Bin Islam; Mohammad Mahabubul Hasan; Mohammad Monirujjaman Khan - 27-30 Oct. 2021.
[2] Stock market prediction using LSTM Recurrent Neural Network - Adil Moghar, Mahmed Hamiche -April 6-9, 2020 .
[3] Comparison of Predictive Algorithms: Backpropagation, SVM,LSTM and Kalman Filter for Stock Market - Divit Karmiani ,Ruman Kazi,Ameya Nambisan, Aastha Shah, Vijaya Kamble.
[4] Predicting Stock Market Price: A Logical Strategy using Deep Learning - Milon Biswas; Atanu Shome; Md. Ashraful Islam; Arafat Jahan Nova; Shamim Ahmed - 3-4 April 2021 .
[5] Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model - Md. Arif Istiake Sunny; Mirza Mohd Shahriar Maswood; Abdullah G. Alharbi - 24-26 Oct. 2020



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)