STOCK PRICE PREDICTION

BY

USING LONG SHORT-TERM MEMORY

A Summer Project Progress Report

Submitted in partial fulfillment for the degree of

Bachelor of Technology

BY

S. GEETHA (N180228)

Under the Guidance of

MR. JAGADEESH DARA

Assistant Professor, Department of CSE

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge & Technologies

(Nuzvid, Eluru, Andhra Pradesh – 521202)

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge & Technologies

(Nuzvid, Eluru, Andhra Pradesh – 521202)

CERTIFICATE OF COMPLETION

This is to certify that the work entitled, "STOCK PRICE PREDICITON USING LONG SHORT-TERM MEMORY" is the bonafied work of **SESHAPU GEETHA (ID No: N180228)** carried out under my guidance and supervision for 3rd year Summer Intern project of Bachelor of Technology in the department of Computer Science and Engineering under RGUKT IIIT, Nuzvid. This work is done during the academic session February 2023 – May 2023, under our guidance.

Mr. Jagadeesh Dara

Assistant professor,

Department of CSE,

RGUKT - Nuzvid

Dr. Chiranjeevi Sadu

Assistant Professor,

Head of the Department,

Department of CSE,

RGUKT - Nuzvid

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge & Technologies

(Nuzvid, Eluru, Andhra Pradesh – 521202)

CERTIFICATE OF EXAMINATION

This is to certify that the work entitled, "STOCK PRICE PREDICITON USING LONG SHORT-TERM MEMORY" is the bonafied work of **SESHAPU GEETHA (ID No: N180228)** and here by accord our approval of it as a study carried out and presented in a manner required for its acceptance in 3rd year of Bachelor of Technology for which it has been submitted. This approval does not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn, as recorded in this thesis. It only signifies the acceptance of this thesis for the purpose for which it has been submitted.

Mr. Jagadeesh Dara
Assistant Professor,
Department of CSE,
RGUKT-Nuzvid

Project Examiner,
RGUKT-Nuzvid

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge & Technologies

(Nuzvid, Eluru, Andhra Pradesh – 521202)

CERTIFICATE OF DECLARATION

We "**SESHAPU GEETHA (ID No: N180228)**" hereby declare that the project report entitled "STOCK PRICE PREDICTION USING LONG SHORT-TERM MEMORY" done by me under the guidance of Mr. Jagadeesh Dara, Assistant Professor, is submitted for the fulfillment of summer intern project during the academic session February 2023- June 2023 at RGUKT - Nuzvid. We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references. The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

Date: 16-08-2023

Place: Nuzvid SESHEPU GEETHA(N180228)

ACKNOWLEDGEMENT

I would like to express my gratitude towards my advisor Mr. Jagadeesh Dara sir for guiding me through the extensive research process. I shall always cherish the time spent with him during this work due to the invaluable knowledge gained in the field of reliability engineering. I am extremely grateful for the confidence bestowed in me and entrusting our project entitled “STOCK PRICE PREDICTION USING LONG SHORT-TERM MEMORY”. I express my gratitude to Mr. Chiranjeevi Sadu (HoD of CSE) and other faculty members for being a source of inspiration and constant encouragement which helped me in completing the project successfully. Finally, yet importantly, I would like to express our heartfelt thanks to our beloved God and parents for their blessings, our friends for their help and wishes for the successful completion of this project.

ABSTRACT

In this project we attempt to implement a machine learning approach to predict stock prices. Machine learning is effectively implemented in forecasting stock prices. The objective is to predict the stock prices to make more informed and accurate investment decisions. We propose a stock price prediction system that integrates mathematical functions, machine learning, and other external factors for the purpose of achieving better stock prediction accuracy and issuing profitable trades. LSTMs are very powerful in sequence prediction problems because they're able to store past information. This is important in our case because the previous price of a stock is crucial in predicting its future price. While predicting the actual price of a stock is an uphill climb, we can build a model that will predict whether the price will go up or down.The main aim of the project is to predict the stock price closing value for a particular company by using machine learning Techniques which is used to build prediction models that can forecast the prices of stock with good accuracy. Prediction model have been proposed to incorporate all the major factors affecting the price of stocks one of the famous technique is Long short-term memory (LSTM) for training and predicting the stock price. To build the stock price prediction model, we will using Real Time company Dataset which consists last 23 years data and we are using Jupyter notebook as a software for Making the prediction. Display result by using Streamlit web app.

Keywords: LSTM, ML, Trade Open, Trade Close, Trade Low, Trade High

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The financial market is a dynamic and composite system where people can buy and sell currencies, stocks, equities and derivatives over virtual platforms supported by brokers. The stock market allows investors to own shares of public companies through trading either by exchange or over the counter markets. This market has given investors the chance of gaining money and having a prosperous life through investing small initial amounts of money, low risk compared to the risk of opening a new business or the need for a high salary career. Stock markets are affected by many factors causing the uncertainty and high volatility in the market. Although humans can take orders and submit them to the market, automated trading systems (ATS) that are operated by the implementation of computer programs can perform better and with higher momentum in submitting orders than any human. However, to evaluate and control the performance of ATSs, the implementation of risk strategies and safety measures applied based on human judgement is required. Many factors are incorporated and considered when developing an ATS, for instance, trading strategy to be adopted, complex mathematical functions that reflect the state of a specific stock, machine learning algorithms that enable the prediction of the future stock value, and specific news related to the stock being analysed.

Time-series prediction is a common technique widely used in many real-world applications such as weather forecasting and financial market prediction. It uses the continuous data in a period of time to predict the result in the next time unit. Many time-series prediction algorithms have shown their effectiveness in practice. The most common algorithms now are based on Recurrent Neural Networks (RNN), as well as its special type Long Short-Term Memory (LSTM). The stock market is a typical area that presents time-series data and many researchers study on it and proposed various models. In this project, the LSTM model is used to predict the stock price.

1.1 MOTIVATION FOR WORK

Businesses primarily run over customer's satisfaction, customer reviews about their products. Shifts in sentiment on social media have been shown to correlate with shifts in stock markets. Identifying customer grievances and thereby resolving them leads to customer satisfaction as well as trustworthiness of an organization. Hence there is a necessity for an unbiased automated system to classify customer reviews regarding any problem. In today's environment where we're justifiably suffering from data overload (although this does not mean better or deeper insights), companies might have mountains of customer feedback collected; but for mere humans, it's still impossible to analyses it manually without any sort of error or bias. Oftentimes, companies with the best intentions find themselves in an insights vacuum. You know you need insights to inform your decision making and you know that you're lacking them, but don't know how best to get them. Sentiment analysis provides some answers into what the most important issues are, from the perspective of customers, at least. Because sentiment analysis can be automated, decisions can be made based on a significant amount of data rather than plain intuition.

1.2 PROBLEM STATEMENT

The main aim of this project is to predict stock prices using Long Short-Term Memory (LSTM). The stock market is known for being volatile, dynamic, and nonlinear. Accurate stock price prediction is extremely challenging because of multiple (macro and micro) factors, such as politics, global economic conditions, unexpected events, a company's financial performance.Time series analysis is a specialized branch of statistics used extensively in fields such as Econometrics & Operation Research. Time series is being widely used in data science. Stock prices are volatile in nature and price depends on various factors.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

"What other people think" has always been an important piece of information for most of us during the decision-making process. The Internet and the Web have now (among other things) made it possible to find out about the opinions and experiences of those in the vast pool of people that are neither our personal acquaintances nor well-known professional critics — that is, people we have never heard of. And conversely, more and more people are making their opinions available to strangers via the Internet. The interest that individual users show in online opinions about products and services, and the potential influence such opinions wield, is something that is the driving force for this area of interest. And there are many challenges involved in this process which need to be walked all over in order to attain proper outcomes out of them. In this survey we analysed basic methodology that usually happens in this process and measures that are to be taken to overcome the challenges being faced.

2.2 EXISTING METHODS

2.2.1 Stock price forecast with deep learning

FIRUZ KAMALOV, LINDA SMILE and IKHLAAS GURRIB, STOCK PRICE FORECASTING WITH DEEP LEARNING, 2021 [1] International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2021, pp. 1419-1426.

In this paper, we compare various approaches to stock price prediction using neural networks. We analyze the performance fully connected, convolutional, and recurrent architectures in predicting the next day value of S&P 500 index based on its previous values. We further expand our analysis by including three different optimization techniques: Stochastic Gradient Descent,

Root Mean Square Propagation, and Adaptive Moment Estimation. The authors in this paper proposed that LSTM is the best technique for time series data and stock prediction.

2.2.2 Recurrent Neural Network approach in predicting daily stock prices

QA. J. P. Samarawickrama and T. G. L. Fernando, A recurrent neural network (RNN)[2] approach in predicting daily stock prices an application to the Sri Lankan stock market, 2019 IEEE International Conference on Industrial and Information Systems (ICIIS), Peradeniya, 2021-March, pp. 1-6.

Recurrent Neural Networks (RNNs) is a sub type of neural networks that use feedback connections. Several types of RNN models are used in predicting financial time series. This study was conducted to develop models to predict daily stock prices of selected listed companies of Colombo Stock Exchange (CSE) based on Recurrent Neural Network (RNN) Approach and to measure the accuracy of the models developed and identify the shortcomings of the models if present. Feed-forward, Simple Recurrent Neural Network (SRNN), Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) [2] architectures were employed in building models. Closing, High and Low prices of the past two days were selected as input variables for each company. Feed-forward networks produce the highest and lowest forecasting errors. SRNN and LSTM networks generally produce lower errors compared with feed-forward networks but in some occasions, the error is higher than feed forward networks. Compared to the other two networks, GRU networks are having comparatively higher forecasting errors.

2.2.3 Stock market prediction using neural network through news on online social networks

Stock market prediction has attracted a lot of attention from both business and academia. In this paper, we implement a model based on Recurrent Neural Networks (RNN)[3] with Gated Recurrent Units (GRU) to predict the stock volatility in the Chinese stock market. We also propose many price-related features which are used as inputs for our model. Apart from that, we carefully select official accounts from Chinese largest online social networks - Sina Weibo and

extract the content posted by these accounts to analyze the public moods. An influence feature is derived based on the public moods to further improve the prediction model. The experimental results show that our model outperforms the baseline method and can achieve a good prediction performance.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 PROPOSED SYSTEMS

The prediction methods can be roughly divided into two categories, statistical methods and artificial intelligence methods. Statistical methods include logistic regression model, ARCH model, etc. Artificial intelligence methods include multi-layer perceptron, Recurrent neural network, naive Bayes network, back propagation network, single-layer LSTM, support vector machine, recurrent neural network, etc. They used Long short-term memory network (LSTM).

Long short-term memory network

Long Short-Term Memory (LSTM) network is a particular form of recurrent neural network (RNN).

3.2 WORKING OF LSTM

LSTM is a special network structure with three "gate" structures. Three gates are placed in an LSTM unit, called input gate, forgetting gate and output gate. When information enters the LSTM network, it can be selected by rules. Only the information conforms to the algorithm will be left, and the information that does not conform will be forgotten through the forgetting gate.

- The first step in LSTM is to decide which information is to be omitted from the cell in that time step. It is decided with the help of a sigmoid function. It looks at the previous state (ht-1) and the current input xt and computes the function.
- There are two functions in the second layer. The first is the sigmoid function, and the second is the tanh function. The sigmoid function decides which values to let through (0 or 1). The tanh function gives the weight age to the values passed, deciding their level of importance from -1 to 1.
- The third step is to decide what will be the final output. First, you need to run a sigmoid layer which determines what parts of the cell state make it to the output. Then, you must put the cell

tate through the tanh function to push the values between -1 and 1 and multiply it by the output of the sigmoid gate.

3.3 LSTM ARCHITECTURE

A special type of RNN, which can learn long-term dependence, is called Long-Short Term Memory (LSTM). LSTM enables RNN to remember long-term inputs. Contains information in memory, like computer memory. It can read, write and delete information from its memory. This memory can be seen as a closed cell, with a closed description, the cell decides to store or delete information. In LSTM, there are three gates: input, forget and exit gate. These gates determine whether new input (input gate) should be allowed, data deleted because it is not important (forget gate) or allow it to affect output at current timeline (output gate).

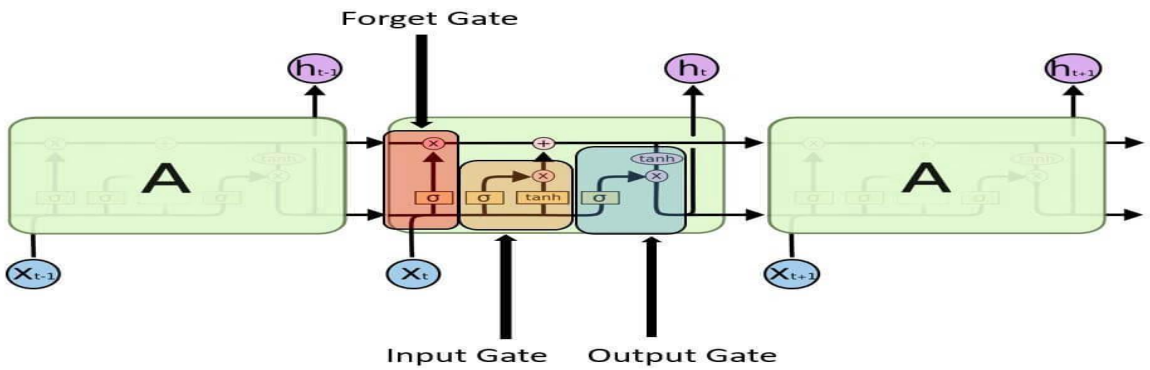


Fig-3.1: LSTM Memory Cell

Forget Gate

A forget gate is responsible for removing information from the cell state. The forget gateway determines when certain parts of the cell will be inserted with information that is more recent. It subtracts almost 1 in parts of the cell state to be kept, and zero in values to be ignored.

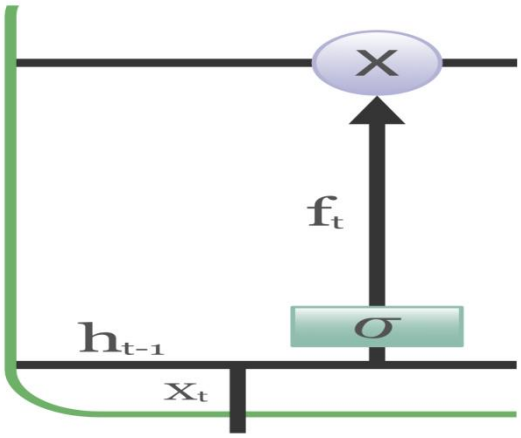


Fig-3.2: Forgot gate in LSTM Cell

- The information that is no longer required for the LSTM to understand things or the information that is of less importance is removed via multiplication of a filter.
- This is required for optimizing the performance of the LSTM network.
- This gate takes in two inputs; h_t-1 and x_t. h_t-1 is the hidden state from the previous cell or the output of the previous cell and x_t is the input at that time step.

Input Gate

Based on the input (e.g., previous output o (t-1), input x(t), and the previous state of cell c (t-1)), this network category reads the conditions under which any information should be stored (or updated) in the state cell.

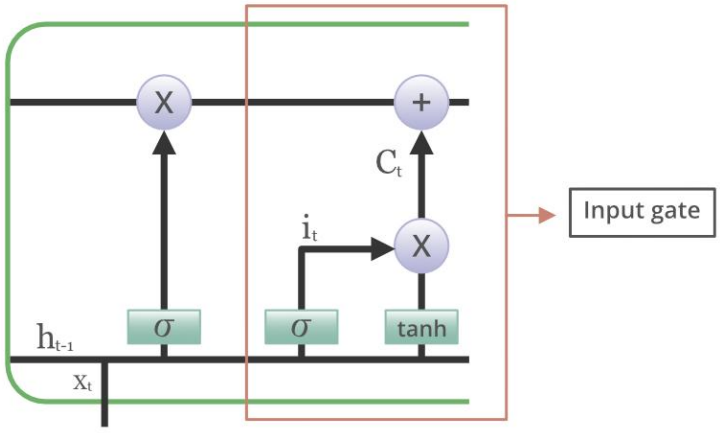


Fig-3.3: Input gate in LSTM cell

- Regulating what values need to be added to the cell state by involving a sigmoid function. This is basically very similar to the forget gate and acts as a filter for all the information from hi-1 and x_t.
- Creating a vector containing all possible values that can be added (as perceived from h_t-1 and x_t) to the cell state. This is done using the tanh function, which outputs values from -1 to +1.
- Multiplying the value of the regulatory filter (the sigmoid gate) to the created vector (the tanh function) and then adding this useful information to the cell state via addition operation. 15

Output Gate

Depending on the input mode and the cell, this component determines which information is forwarded to the next location in the network.

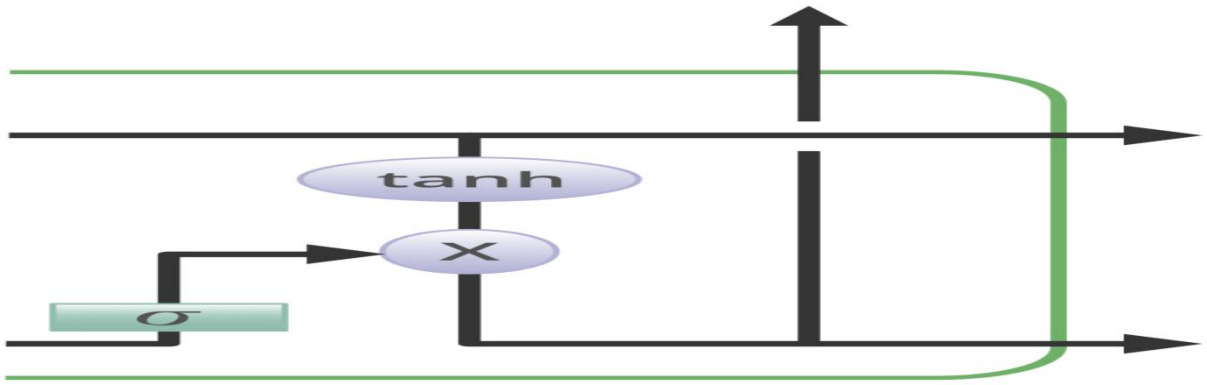


Fig-3.4: Output gate in LSTM cell

The functioning of an output gate can again be broken down into three steps

- Creating a vector after applying tanh function to the cell state, thereby scaling the values to the range -1 to +1.
- Making a filter using the values of h_t-1 and x_t, such that it can regulate the values that need to be output from the vector created above. This filter again employs a sigmoid function.
- Multiplying the value of this regulatory filter to the vector created in step 1, and sending it out as an output and to the hidden state of the next cell.

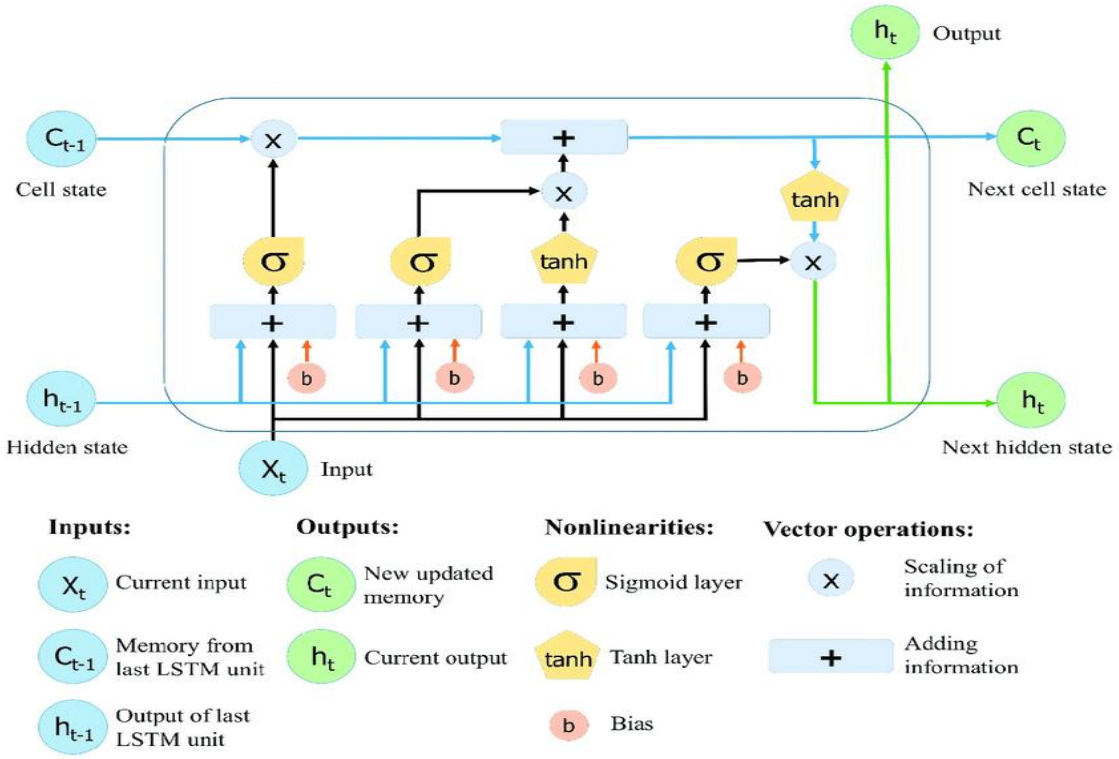


Fig-3.5: Structure of the LSTM

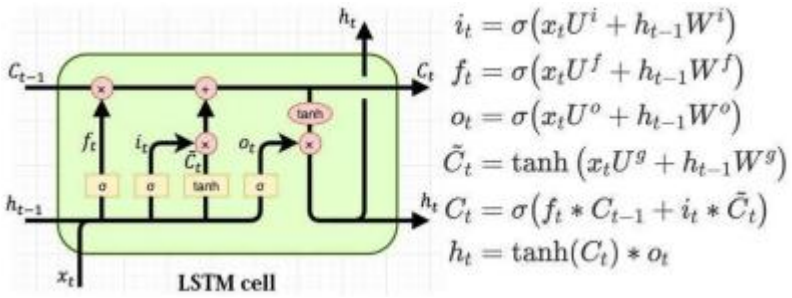


FIg-3.6: Inner process of Memory Cell

ADVANTAGE OF LSTM

The main advantage of LSTM is its ability to read intermediate context. Each unit remembers details for a long or short period without explicitly utilizing the activation function within the recurring components. An important fact is that any cell state is repeated only with the release of the forget gate, which varies between 0 and 1. That is to say, the gateway for forgetting in the LSTM cell is responsible for both the hardware and the function of the cell state activation. Thus, the data from the previous cell can pass through the unchanged cell instead of explicitly increasing or decreasing in each step or layer, and the instruments can convert to their appropriate values over a limited time. This allows LSTM to solve a perishable gradient problem because the amount stored in the memory cell is not converted in a recurring manner, the gradient does not end when trained to distribute backwards.

3.4 SYSTEM ARCHITECTURE

3.4.1 Overall Architecture

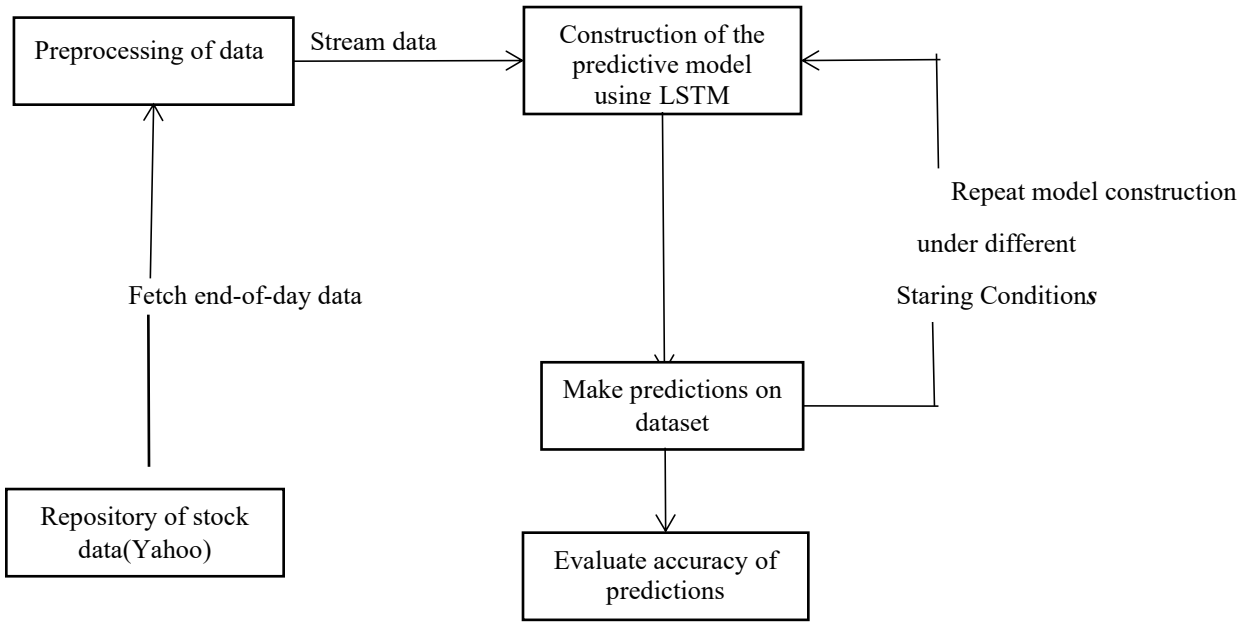
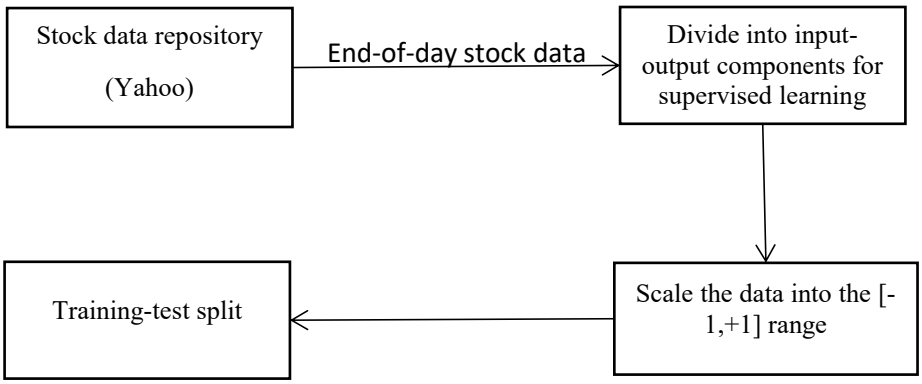


Fig-3.7: LSTM-based stock price prediction system

The stock prediction system depicted in the above figure has three main components. A brief explanation of each is given below

Obtaining dataset and preprocessing



Fig–3.8: Data preprocessing

Stock market data (for end-of-day prices of various ticker symbols i.e., companies) was obtained primary source Yahoo Finance. These websites offer URL-based APIs from which historical stock data for various companies can be obtained by simply specifying some parameters in the URL. The entries are present in the dataset. The null values are removed using df = df.dropna() where df is the data frame. The categorical attributes (Date, High, Low, Close, Adj value) are converted into numeric using Label Encoder. The date attribute is splitted into new attributes like total which can be used as feature for the model.

Feature selection

Features selection is done which can be used to build the model. The attributes used for feature selection are Date, Price, Adj close, Forecast X coordinate, Y coordinate, Latitude, Longitude, Hour and month.

Building and Training Model

After feature selection location and month attribute are used for training. The dataset is divided into pairs of x_train, y_train and x_test, y test. The algorithms model is imported from sklearn. Building model is done using model. Fit (x_train, y_train).

Terminologies used

Given below is a summary of the various terminologies relating to our proposed stock prediction system

- **Training set:** subsection of the original data that is used to train the neural network model for predicting the output values
- **Test set:** part of the original data that is used to make predictions of the output value, which are then compared with the actual values to evaluate the performance of model
- **Validation set:** portion of the original data that is used to tune the parameters of the neural network model
- **Activation function:** in a neural network, the activation function of a node defines the output of that node as a weighted sum of inputs.

  Activation function=Σ(Inputs*weights) +Bias

  ReLU – has the following formula

  y=max (0, x)

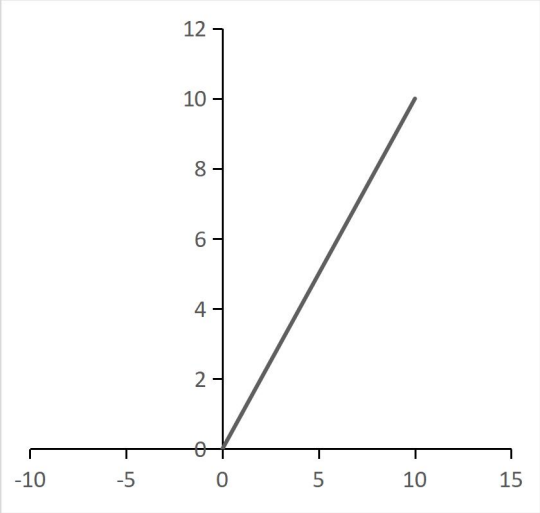


Fig-3.9: ReLU activation function

- **Batch size:** number of samples that must be processed by the model before updating the weights of the parameters
- **Epoch:** a complete pass through the given dataset by the training algorithm

- **Dropout:** a technique where randomly selected neurons are ignored during training i.e., they are "dropped out" randomly. Thus, their contribution to the activation of downstream neurons is temporally removed on the forward pass, and any weight updates are not applied to the neuron on the backward pass.
- **Loss function:** a function, defined on a data point, prediction and label, that measures a penalty such as square loss which is mathematically explained as follows

$$l(f(xi),yi) = (f(xi)-yi)^2$$

- **Cost function:** a sum of loss functions over the training set. An example is the Mean Squared Error (MSE), which is mathematically explained as follows

$$MSE() = \sum Ni = l(f(xi)-yi)^2/N$$

- **Root Mean Square Error (RMSE):** measure of the difference between values predicted by a model and the values actually observed. It is calculated by taking the summation of the squares of the differences between the predicted value and actual value and dividing it by the number of samples. It is mathematically expressed as follows

$$\sqrt{\frac{\sum(y\ predicted - y\ actual)2}{N}}$$

CHAPTER 4

REQUIREMENTS AND ANALYSIS

4.1 INTRODUCTION

Design is a multi-step that focuses on data structure software architecture, procedural details, procedure etc. and interface among modules. The design procedure also decodes the requirements into presentation of software that can be accessed for excellence before coding begins. Computer software design changed continuously as novel methods; improved analysis and border understanding evolved. Software proposal is at a relatively primary stage in its revolution. Therefore, software design methodology lacks the depth, flexibility and quantitative nature that are usually associated with more conventional engineering disciplines. However, methods for software designs do exist, criteria for design qualities are existing and design notation can be applied.

4.2 SYSTEM REQUIREMENTS

4.2.1 Hardware Requirements

- RAM: 4 GB
- MAIN MEMORY: 4GB RAM
- PROCESSING SPEED: 400 MHZ
- KEYBOARD :104 KEYS

4.2.2 Software Components

- FRONT END: PYTHON
- IDE: JUPYTER NOTEBOOK
- OPERATING SYSTEM: WINDOWS 10

PYTHON

Python offers concise and readable code. While complex algorithms and versatile workflows stand behind machine learning Python's simplicity allows developers to write reliable systems. Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.

JUPYTER NOTEBOOK

It allows users to create and share documents containing live code, equations visualizations. Jupyter Notebook supports multiple programming languages, including Python, R, and Julia. It provides an intuitive and flexible environment for data exploration, and collaboration. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning.

LIBRARIES

Python Library

Machine Learning, as the name suggests, is the science of programming a computer by which they are able to learn from different kinds of data. In the old days, people used to perform Machine Learning tasks by manually coding all the algorithms and mathematical and statistical formulas. This made the process time-consuming, tedious and inefficient. But in the modern days, it has become very easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reasons is its vast collection of libraries.

Python libraries that used in Machine Learning are:

- NumPy
- Keras
- Scikit- learn
- Pandas
- DataReader

- Yfinance
- DateTime
- Matplotlib

NumPy

NumPy is a very popular python library for large multi- dimensional array and matrix processing, with the help of a large collection of high- level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High- end libraries like TensorFlow use NumPy internally for manipulation of Tensors.

Keras

Keras is a very popular Machine Learning library for Python. It can run seamlessly on both CPU and GPU. Keras makes it really for ML beginners to build and design a Neural Network. One of the best thing about Keras is that it allows for easy and fast prototyping.

Scikit-Learn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

Pandas

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high- level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

DataReader

Pandas DataReader is a Python library that enables you to retrieve data from various sources and store it in Pandas data structures such as Data Frame and Series. It is built on top of the popular data manipulation library, Pandas, and provides a simple and efficient way to retrieve data.Yahoo Finance, Google Finance, World Bank, Federal Reserve Economic Data (FRED) and many more.

Yfinance

Yfinance is a Python library that provides convenient access to the Yahoo Finance API. It allows users to download historical stock data, get real-time stock quotes, and obtain information about stock symbols, such as company name and stock exchange. Yfinance is built on top of the Yahoo Finance API, providing a simple and easy-to-use interface for accessing the data and functionality of the API in a Python environment. Yfinance is available on PyPI, so it can be easily installed using pip install yfinance.

DateTime

Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times, and time intervals. Date and DateTime are an object in Python, so when you manipulate them, you are actually manipulating objects and not strings or timestamps.

Matplotlib

Matpoltlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named Pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar chats, etc.

CHAPTER 5

IMPLEMENTATION

Import all required libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# For reading stock data from yahoo
from pandas_datareader.data import DataReader
import yfinance as yf
```

Fig-5.1: Importing required libraries

Load Dataset

```
#yf.pdr_override()
from datetime import datetime
end = datetime.now()
end_str = end.strftime('%Y-%m-%d')
start = datetime(end.year - 23, end.month, end.day)
start_str=start.strftime('%y-%m-%d')


# Fetch historical stock data for AAPL from Yahoo Finance
#df = data.DataReader('AAPL', 'yahoo', start, end)
df=yf.download('AAPL', start, end)

# Display the first few rows of the DataFrame
print(df.head())
```

Fig-5.2: Load dataset of any company

By utilizing the yfinance library we can load specified company stock dataset from yahoo finance adjust the start_date and end_date accordingly. The yf.download() function fetches the historical stock data for the specified symbol and date range. In our project we loaded data from the last 23 years to today.

For removing data as index, we use reset_index()

```
df = df.reset_index()
df.head()
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2000-08-14	0.849888	0.851563	0.827009	0.840402	0.713331	156660000
1	2000-08-15	0.843750	0.856027	0.830357	0.833705	0.707646	114200800
2	2000-08-16	0.837054	0.875000	0.835938	0.866071	0.735118	143673600
3	2000-08-17	0.863839	0.936384	0.862723	0.918527	0.779642	270900000
4	2000-08-18	0.917411	0.925223	0.890625	0.892857	0.757854	190176000

Fig-5.3: Displaing dataset

The obtained data contained five features:

1. Date: Date of stock price.

2. Opening price: When trading begins each day this is the opening price of stock.

3. High: The highest price at which the stock was traded during a period(day).

4. Low: The Lowest price at which the stock was traded during a period(day).

5. Volume: How much of a given financial asset has traded in a period of time.

6. Close Interest: The last price at which a particular stock traded for the trading session.

We are working on the close price column and 100 days moving averages so, from the 100th index of the dataset. we calculate the previous 100 data values average of closing values.

Splitting dataset into training and testing

```
# splitting Data into Training and Testing

data_training=pd.DataFrame(df['Close'][0:int(len(df)*0.70)])
data_testing= pd.DataFrame(df['Close'][int(len(df)*0.70): int(len(df))])
print(data_training.shape)
print(data_testing.shape)
```

```
(4050, 1)
(1736, 1)
```

Fig-5.4: Splitting the dataset into training and testing

We split close price data into 70% for training and 30 for testing.

Scaling on the data

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
scaler
```

```
▾ MinMaxScaler
MinMaxScaler()
```

```
data_training_array=scaler.fit_transform(data_training)
data_training_array
```

```
array([[0.01835841],
       [0.01815557],
       [0.01913589],
       ...,
       [0.86801739],
       [0.86309548],
       [0.85294882]])
```

Fig-5.5: Scaling on the data

The purpose of using the MinMaxScaler is to scale the features of your dataset to a specific range. here 0 to 1. We can't provide data directly. This can be particularly useful in scenarios where features have different scales or units, as scaling them to a common range can improve the performance of certain machine learning algorithms. can work more efficiently when the features are on a similar scale.

Machine Learning Model using LSTM from Keras

```
model=Sequential()
model.add(LSTM(units=50,activation='relu',return_sequences=True,input_shape=(x_train.shape[1],1)))
model.add(Dropout(0.2))

model.add(LSTM(units=60,activation='relu',return_sequences=True))
model.add(Dropout(0.3))

model.add(LSTM(units=80,activation='relu',return_sequences=True))
model.add(Dropout(0.4))

model.add(LSTM(units=120,activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(units=1))
```

Fig-5.6: Using LSTM model

Fit model and saving the model

```
model.compile(optimizer='adam',loss="mean_squared_error")
model.fit(x_train,y_train,epochs=100)
```

```
Epoch 1/100
124/124 [==============================] - 43s 286ms/step - loss: 0.0204
Epoch 2/100
124/124 [==============================] - 36s 287ms/step - loss: 0.0048
```

```
model.save('my_model.keras')
```

Fig-5.7: Build and save the Model

Split testing data into x_text and y_test and convert them into numpy array

```
x_test=[]
y_test=[]

for i in range(100,input_data.shape[0]):
    x_test.append(input_data[i-100:i])
    y_test.append(input_data[i,0])
```

```
x_test,y_test=np.array(x_test),np.array(y_test)
print(x_test.shape)
print(y_test.shape)
```

```
(1736, 100, 1)
(1736,)
```

Fig-5.8: Splitting testing data into x_text and y_test

make prediction

```
# make predictions

y_predicted=model.predict(x_test)
```

```
55/55 [==============================] - 7s 104ms/step
```

Fig-5.9: make prediction

Scaling the y_predicted and calculate r2_score

```
scaler.scale_
scaling_factor = scaler.scale_[0]
scaling_factor
```

```
0.005751588947095272
```

```
scaler_factor=1/scaling_factor
y_predicted=y_predicted*scaler_factor
y_test=y_test*scaler_factor
```

```
from sklearn.metrics import r2_score

r2 = r2_score(y_test, y_predicted)
print("R-squared:", r2)
```

```
R-squared: 0.9578967064240376
```

Fig-5.10: Scaling the y_predicted and calculate r2_score

```
import matplotlib.pyplot as plt

# Assuming y_predicted and y_test are correctly defined arrays/lists
# Make sure that they contain the correct data and are of the same length

plt.figure(figsize=(12, 6))

# Plotting the predicted and original prices
plt.plot(y_predicted, 'r', label="Predicted Price")
plt.plot(y_test, 'b', label='Original Price')

plt.xlabel("Time")
plt.ylabel("Price")
plt.legend()
plt.title("Predicted vs Original Price Over Time")  # Adding a title for clarity
plt.show()
```

Fig-5.11: Plotting the predicted and testing values

APPL Company stock prediction

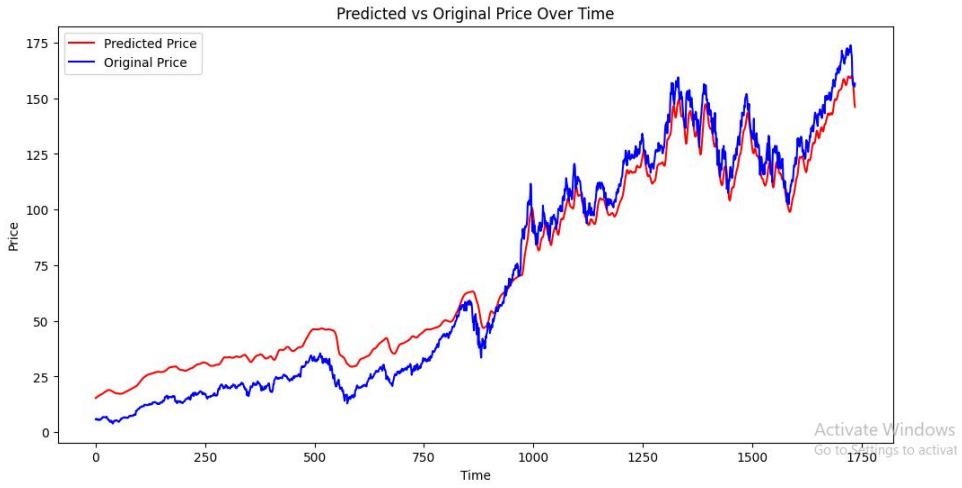


Fig-5.12: APPL company stock prediction

Streamlit python web app

It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, NumPy, pandas, Matplotlib etc.

CHAPTER 6

RESULT

We can predict future stock prices based on the last 23 years of the specified company.

User Interface

	Open	High	Low	Close	Adj Close	Volume
count	5,785	5,785	5,785	5,785	5,785	5,785
mean	34.3495	34.7316	33.9809	34.3712	33.0569	405,205,441.4852
std	48.2007	48.763	47.6777	48.2434	48.0474	389,729,917.3748
min	0.232	0.2355	0.2271	0.2343	0.1989	31,458,200
25%	2.4461	2.4857	2.4054	2.4393	2.0705	131,022,800
50%	15.0629	15.2136	14.9643	15.0618	12.9153	283,192,000
75%	39.945	40.2425	39.61	39.9625	37.734	546,123,200
max	196.24	198.23	195.28	196.45	196.1851	7,421,640,800

Fig-6.1: Describing the specified company stock price

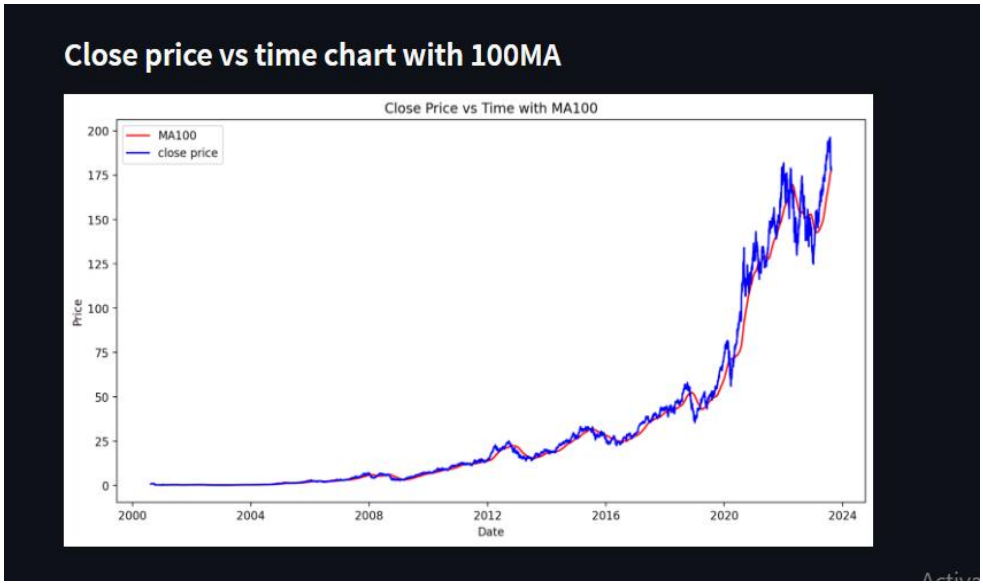


Fig-6.2: Closing price Vs time chart with 100 Moving Average

Fig-6.3: Prediction vs Original

CHAPTER 7

CONCLUSION

Stock investing has attracted the interest of many investors around the world. However, making a decision is a difficult task as many things are involved. By investing successfully, investors are eager to predict the future of the stock market. Even the slightest improvement in performance can be enormous. A good forecasting system will help investors make investments more accurate and more profitable by providing supporting information such as future stock price guidance. In addition to historical prices, other related information could affect prices such as politics, economic growth, financial matters and the atmosphere on social media. Numerous studies have proven that emotional analysis has a significant impact on future prices. Therefore, the combination of technical and basic analysis can produce very good predictions. In this project we used LSTM technique for Training the model and we have taken Apple Dataset and we trained our model. We split the data into 70% for training and 30% for Testing. We Predicted the Stock price closing for the Dataset. We have achieved 90% of stock price prediction for the dataset.

.

CHAPTER 8

FUTURE SCOPE

LSTM offer potential to capture complex temporal patterns and nonlinearities in stock price data. Challenges include data quality, market volatility, overfitting, and recognizing that accurate predictions don't guarantee profits. Hybrid models combining LSTM with other techniques, such as technical analysis or sentiment analysis, can enhance prediction robustness. Emphasis on model interpretability to provide traders and investors with actionable insights. Ensemble methods that combine multiple LSTM models or integrate other model types for improved accuracy. Continuous learning mechanisms to enable adaptation to evolving market conditions over time. Efforts towards uncertainty estimation techniques to provide prediction confidence intervals. Real-time predictions through improved speed of LSTM training and inference. Integrating LSTM predictions into trading strategies requires a balanced understanding of their capabilities and limitations within the dynamic landscape of financial markets.

REFERENCES

[1] Sen, J "Stock Price Prediction Using Machine Learning and Deep Learning Frameworks" Proceedings of the 6th International Conference on Business Analytics and Intelligence, B angalore, India, December 20-22 (2018)

[2] Jeevan B,Naresh E "Share Price Prediction using Machine Learning Technique", IEEE Third International Conference on Circuits, Control, Communication and Computing, 2018

[3] Adebiyi, A., Adewumi, O., and Ayo, C. K. Ayo: "Stock Price Prediction Using the ARIMA Model" In: Proc. of the Int. Conf. on Computer Modelling and Simulation, Cambridge, UK, pp. 105-111 (2018)