★ **COURSE NAME :** Data science fundamentals programs

★ **PROJECT TITLE :** Bitcoin price prediction

★ **PROJECT SUBMITTED TO/IBM :** Anna University / Naan mudhalvan

★ **YEAR :** 3rd Year

★ **BRANCH/DEPARTMENT :** B.E. Mechanical Engineering

★ **SEMESTER :** 6th

★ **PROJECT SUBMITTED BY :** V. Kaviyarasu

Bitcoin Price Prediction Using Deep Learning Algorithm LSTM and Sentiment Analysis on Bitcoin.

Contents Overview

1. Introduction
2. Why this project?
3. What is NN and RNN?
4. Why LSTM over RNN?
5. Software model - Agile Process
6. User and System Requirements
7. Design
8. Implementation
9. Testing
10. Results
11. Future Work

References

Introduction

- Bitcoin is an innovative payment network and a new kind of money or a cryptocurrency, where cryptocurrency is a digital asset designed to work as a exchange medium that uses strong cryptography to secure financial transactions, control the creation of additional units, and verify the transfer of assets. BITCOIN, the first cryptocurrency was introduced in a paper published in 2008 by an author under pseudonym of SATOSHI NAKAMOTO.
- We aim to incorporate machine learning to analyse past fluctuations in currency prices, and attempt to decipher a trend in prices, this is because we can’t predict accurately and sentiment analysis of bitcoin from twitter tweets.

Why this project

- As we reach, in the final year, we will soon be responsible, earning individuals, who will want to save money and invest it properly to gain huge benefits.

- Since stocks and cryptocurrency trading are in trend, we choose to help common man to learn how this works, and enable them to invest judiciously by studying this **Trend Analysis.**

NN and RNN Neural Networks

Conventional Feed-Forward Neural Network

Feed-forward Neural Network is an unidirectional that moves the information only in one direction from input layers,through the hidden layers,to the output layers.

Feed-forward Neural Network ,have no memory of previously received input hence cannot predict the coming next input and it considers only current input. Simply they cannot remember the past inputs except their training

NN and RNN Neural Networks

Recurrent neural network

- Recurrent neural network(RNN) falls under the class of artificial neural network or advanced artificial neural network in which direct cycles in memory are involved. Recurrent neural networks has the ability to build networks with fixed sized input and output layers.In RNN the information cycles in the form of a loop. Hence,RNN not only considers the current input but also the previously received inputs.
- Thus RNN contains two inputs one is the current and the other is previous inputs.Also Feed-forward maps one input to one output but RNN can map one to many,many to one and many to many.

Why LSTM Over Other Neural Networks

Drawbacks of RNN neural network

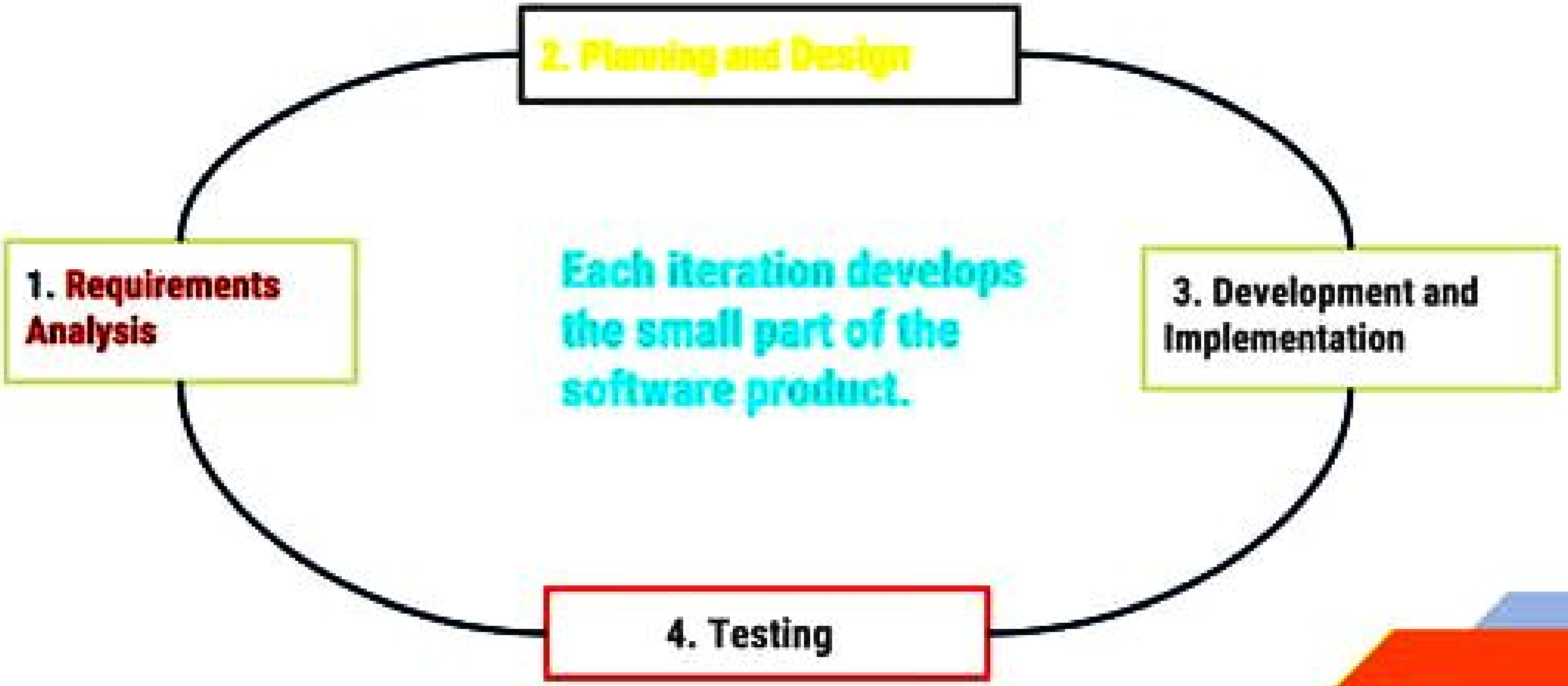
1. **Long term dependencies** is the drawback of RNN neural networks.The RNN neural networks consider both previous and present input to perform the task. But, in some cases it doesn't need to consider the previous input as the output purely depends on current input.Consider the example "the clouds are in the sky", in which we are trying to predict the last word based on previous word and following example needs small context or small gap between relevant information.But in some cases it needs large context.

Why LSTM Over Other Neural Networks

Consider trying to predict the last word in the text "I grew up in France… I speak fluent *French*." Recent information suggests that the next word is probably the name of a language, but if we want to narrow down which language, we need the context of France, from further back. It's entirely possible for the gap between the relevant information and the point where it is needed to become very large.

Unfortunately, as that gap grows, RNNs become unable to learn to connect the information. In theory, RNNs are absolutely capable of handling such "long-term dependencies". But in practical they don't.This problem is solved in LSTM

Agile Process Model



Each iteration develops the small part of the software product.

User And System Requirements

User Requirements

- To get the predicted price of a bitcoin
- To see the trend in variation of bitcoin

System requirements

- To incorporate machine learning algorithms in order decipher the a trend in prices .
- To use various API which provides user-friendly UI.

ALGORITHM

“ LSTM (Long Short Term Memory) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information in and out of the cell.

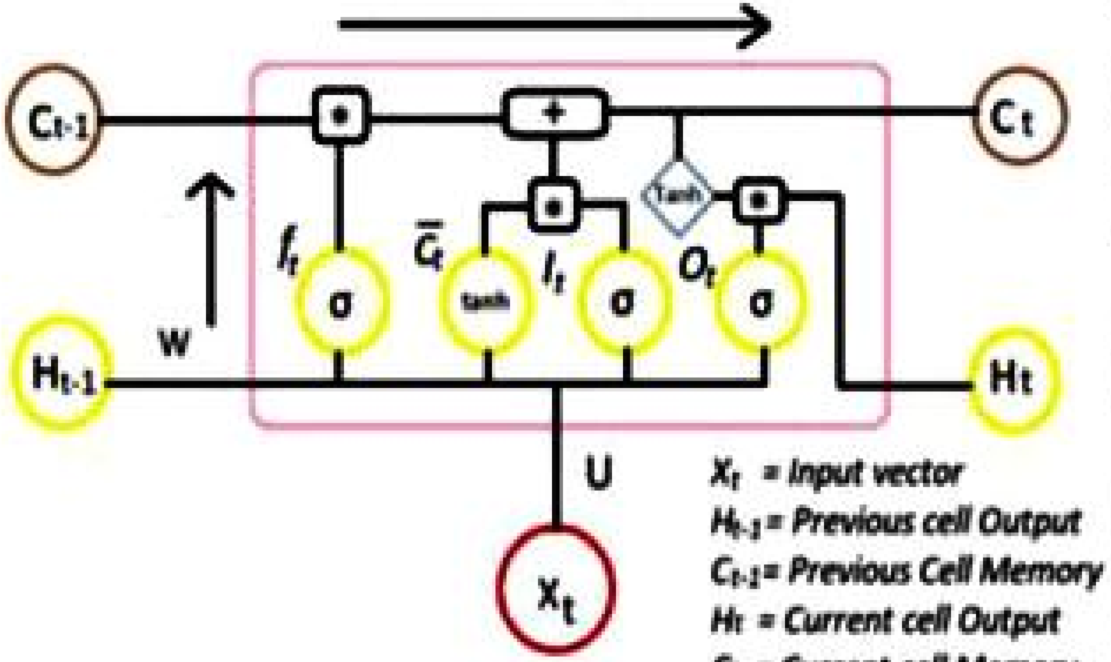
Step 1: First step is to discard cell state. This decision is taken by the Sigmoid layer.

Step 2: The input gate layer then decides which information to store in the cell state. Then the tanh layer creates a vector of new candidate values.

Step 3: Next, We multiply the old state by ft. This is the new value, scaled by how much we decided to update each state value.

Step 4: First, we run a sigmoid layer which decides what parts of the cell state .That is our going to output. Then, we put the cell state through tanh (to push the values to be between −1 and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

LSTM Network



$\boxed{\bullet}$ = Element-wise multiplication

$\boxed{+}$ = Element-wise addition

$$f_t = \sigma\,(X_t * U_f + H_{t-1} * W_f)$$
$$\tilde{C}_t = \tanh\,(X_t * U_c + H_{t-1} * W_c)$$
$$I_t = \sigma\,(X_t * U_i + H_{t-1} * W_i)$$
$$O_t = \sigma\,(X_t * U_o + H_{t-1} * W_o)$$

$$C_t = f_t \bullet C_{t-1} + I_t * \tilde{C}_t$$
$$H_t = O_t \bullet \tanh\,(C_t)$$

X_t = Input vector

H_{t-1} = Previous cell Output

C_{t-1} = Previous Cell Memory

H_t = Current cell Output

C_t = Current cell Memory

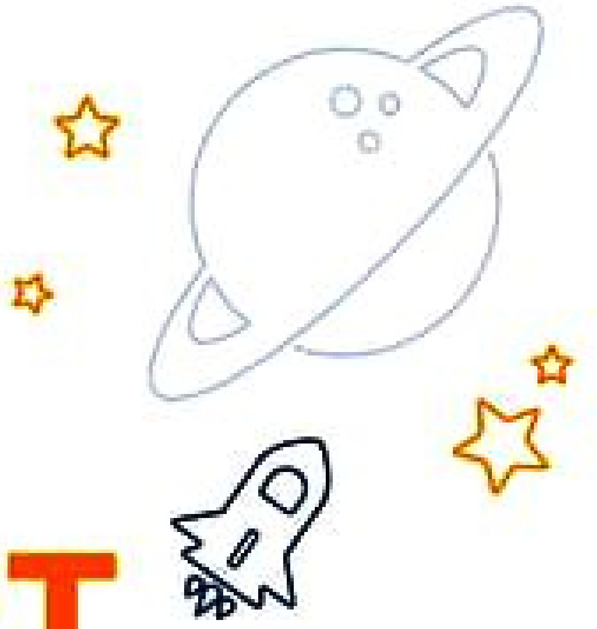
W, U = weight vectors for forget gate (f), candidate (c) , i/p gate (i) and o/p gate (O)

Note : These are different weights for different gates, for simpicity's sake, I mentioned W and U

Ex : **Mady** and **Monica** walk in to the room together , later **Richard** walks in to the room. **Mady** Said “hi” to ____??

The assumption I am making is memory might change from Monica to Richard.

SHORT EXAMPLE



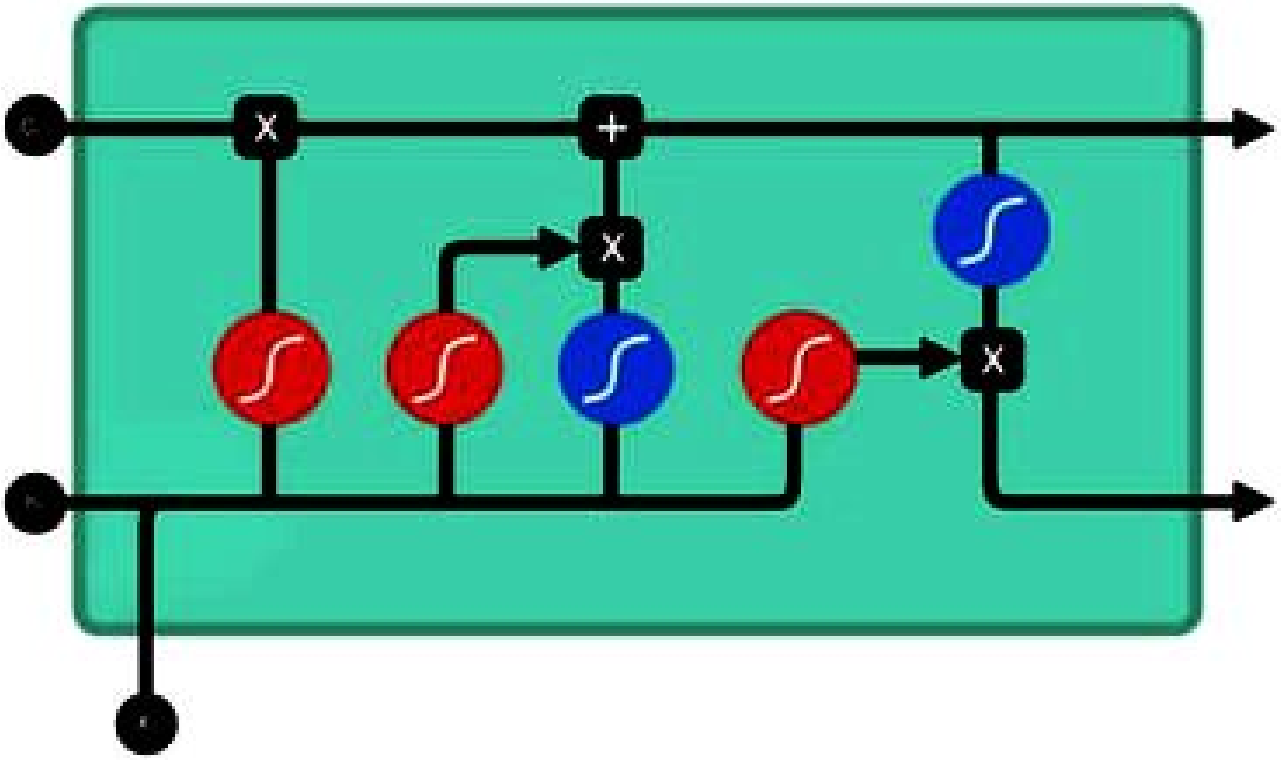
BIGGER CONCEPT

IMPLEMENTATION AND DESIGN

7

DESIGN

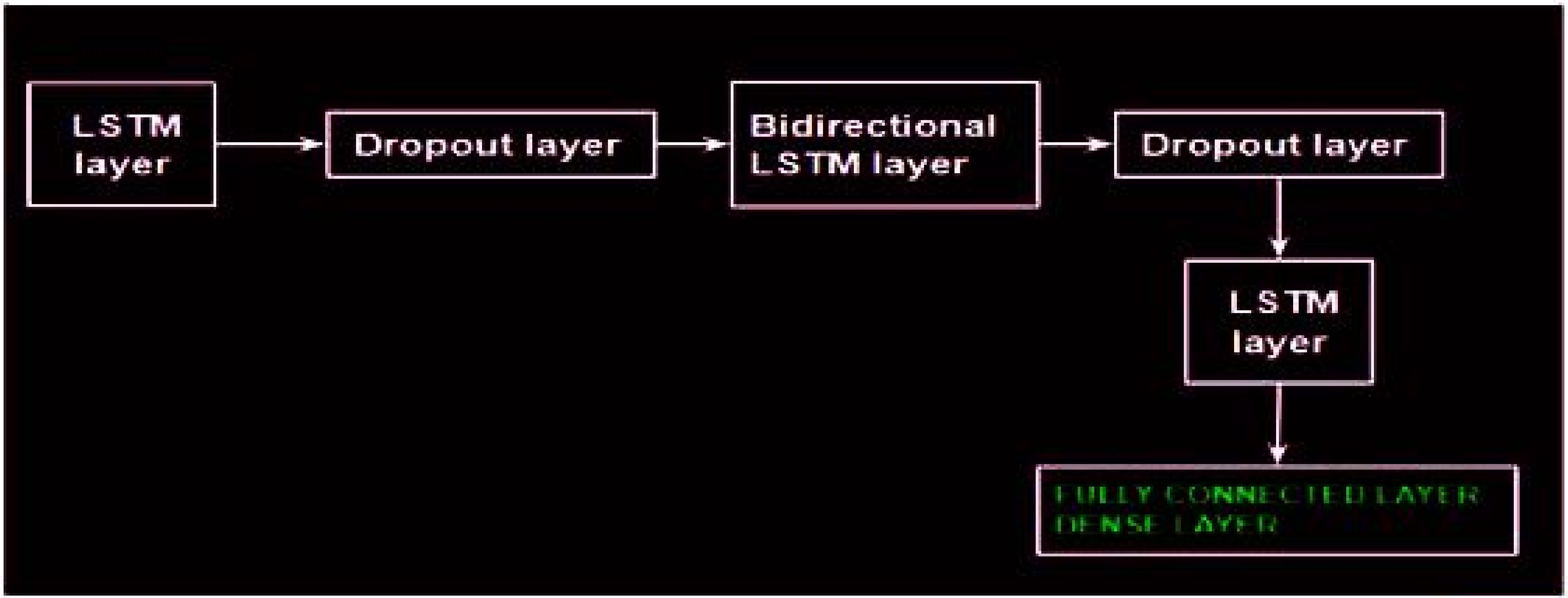
The Model



● previous cell state

● forget gate output

Model

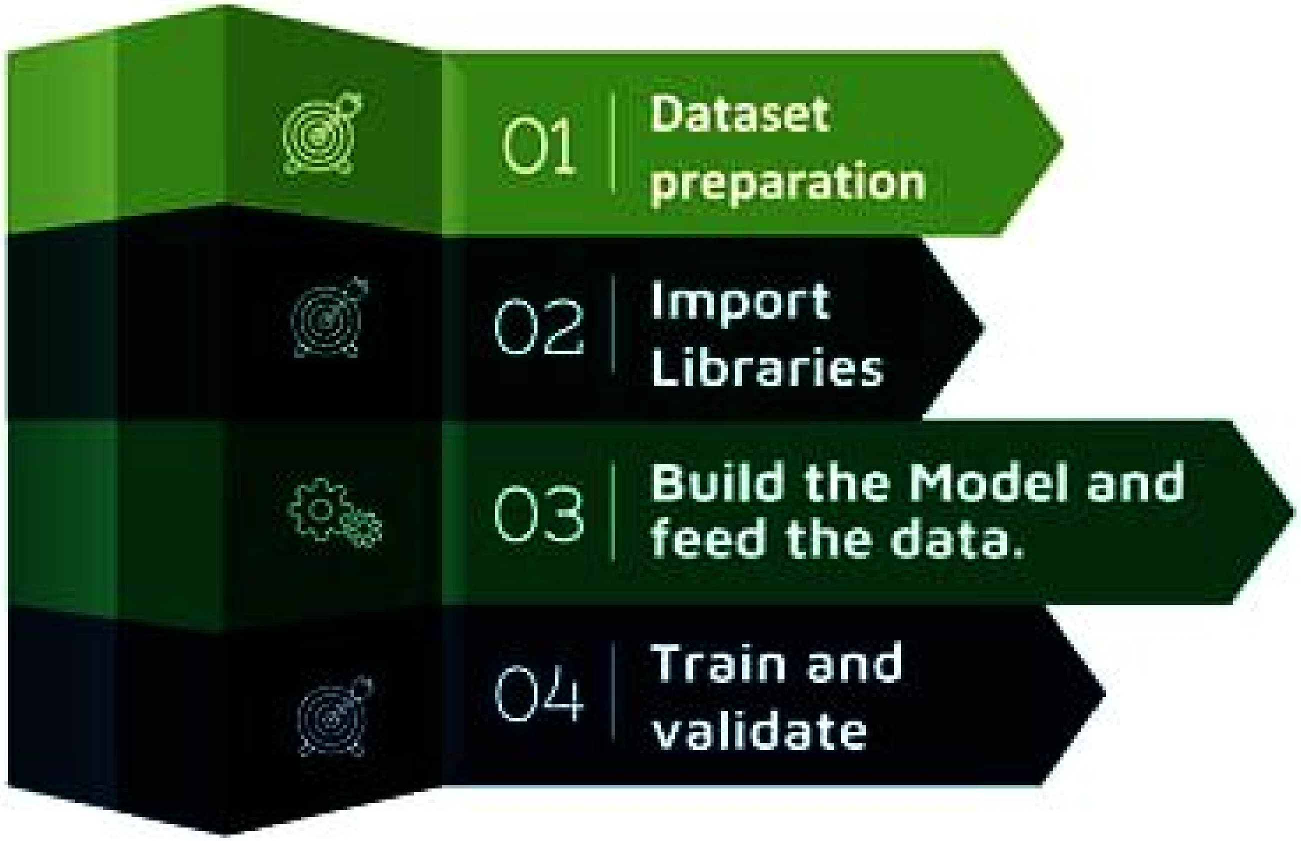

LSTM layer
Dropout layer
Bidirectional LSTM layer
Dropout layer
LSTM layer
FULLY CONNECTED LAYER
DENSE LAYER

8

Implementation

Model Implementation

STEP WISE PROCEDURE



01 | Dataset preparation

02 | Import Libraries

03 | Build the Model and feed the data.

04 | Train and validate

Data Preparation

Source: https://coinmarketcap.com/

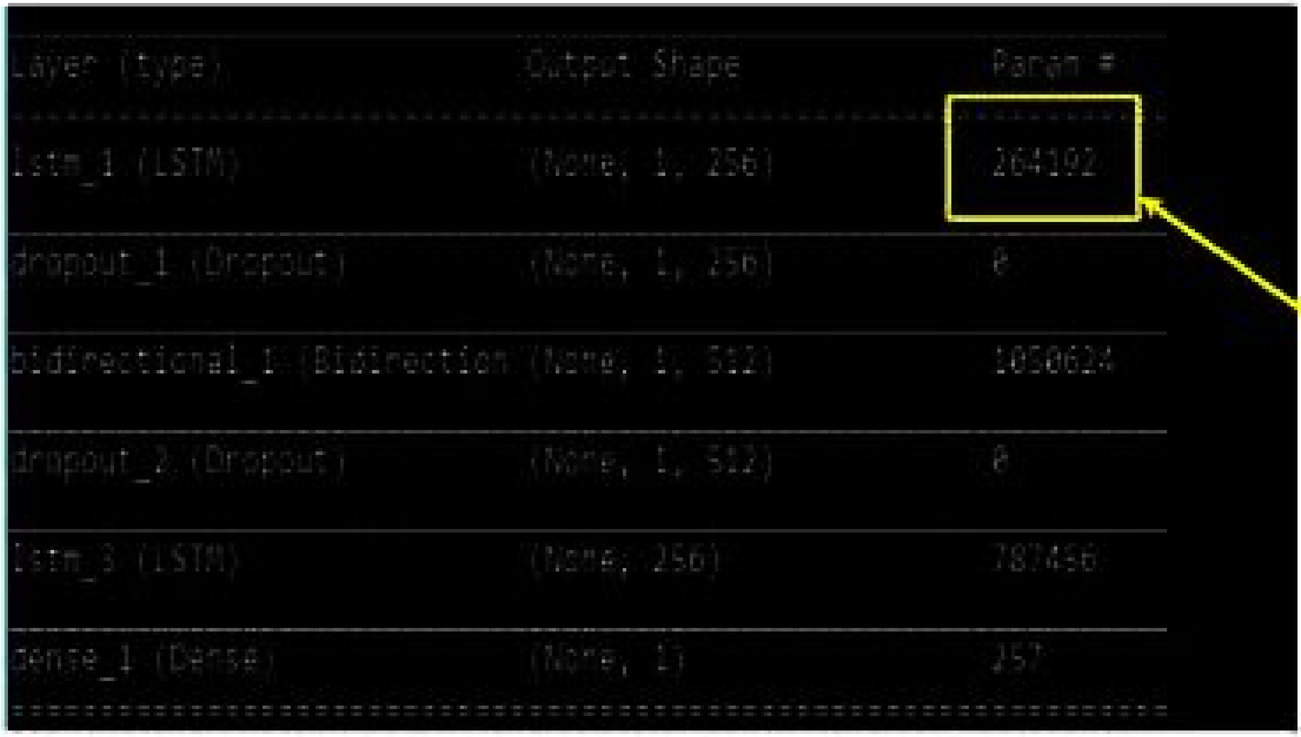
Data Preprocessing:

- Divide the dataset into and training and testing set.
- Normalize and reshape.

Libraries and Tools used

- Keras
- Sklearn
- NumPy
- SpyDer - IDE
- TextBlob

Model.summary()



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 1, 256)            264192
_________________________________________________________________
dropout_1 (Dropout)          (None, 1, 256)            0
_________________________________________________________________
bidirectional_1 (Bidirection (None, 1, 512)            1050624
_________________________________________________________________
dropout_2 (Dropout)          (None, 1, 512)            0
_________________________________________________________________
lstm_3 (LSTM)                (None, 256)               787456
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 257
=================================================================
```

4x[(input shape + 1)xoutput shape + output shape^2]

Therefore,

4*[(1+1)*256+256*256]

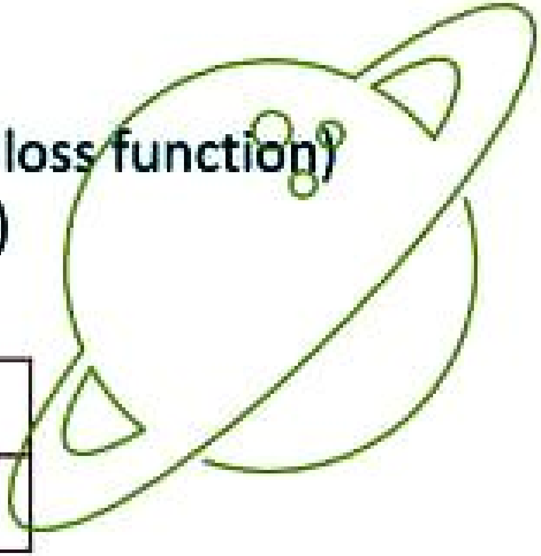
= 4*[2*256+65,536]

= 4*[66,048] = 2,64,192.

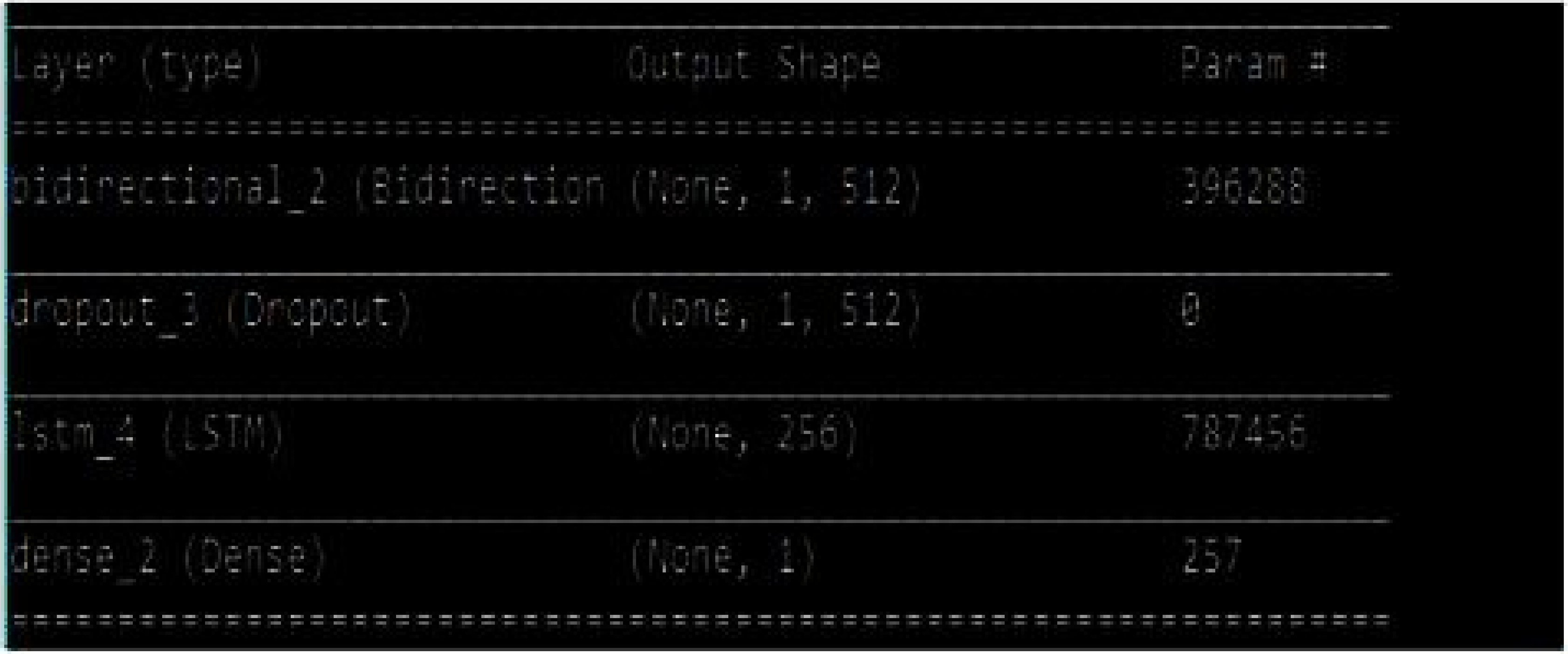
Model Fitting using Back Propagation

- Back propagate errors(error calculation with loss function)
- Loss optimization (optimizer-update weights)
- Calculate gradient.

Positive Gradient →	Decrease weight
Negative Gradient →	Increase weight



Transfer Learning



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional_2 (Bidirection (None, 1, 512)            396288
_________________________________________________________________
dropout_3 (Dropout)          (None, 1, 512)            0
_________________________________________________________________
lstm_4 (LSTM)                (None, 256)               787456
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 257
=================================================================
```

7

DESIGN

Sentiment Analysis

Sentiment Analysis

Sentiment Analysis is the process of determining(computationally) whether a piece of writing is positive, negative or neutral. It is also known as **opinion mining**.

There are already many APIs developed for sentiment analysis so we decided to use those existing APIs. The APIs or python library which we planned to use are as follows,

- **Tweepy** - which is python client
- **TextBlob** - Textual Data Parser and provides built program of sentiment analysis for textual data.
- Natural language toolkit corpora

8

Implementation

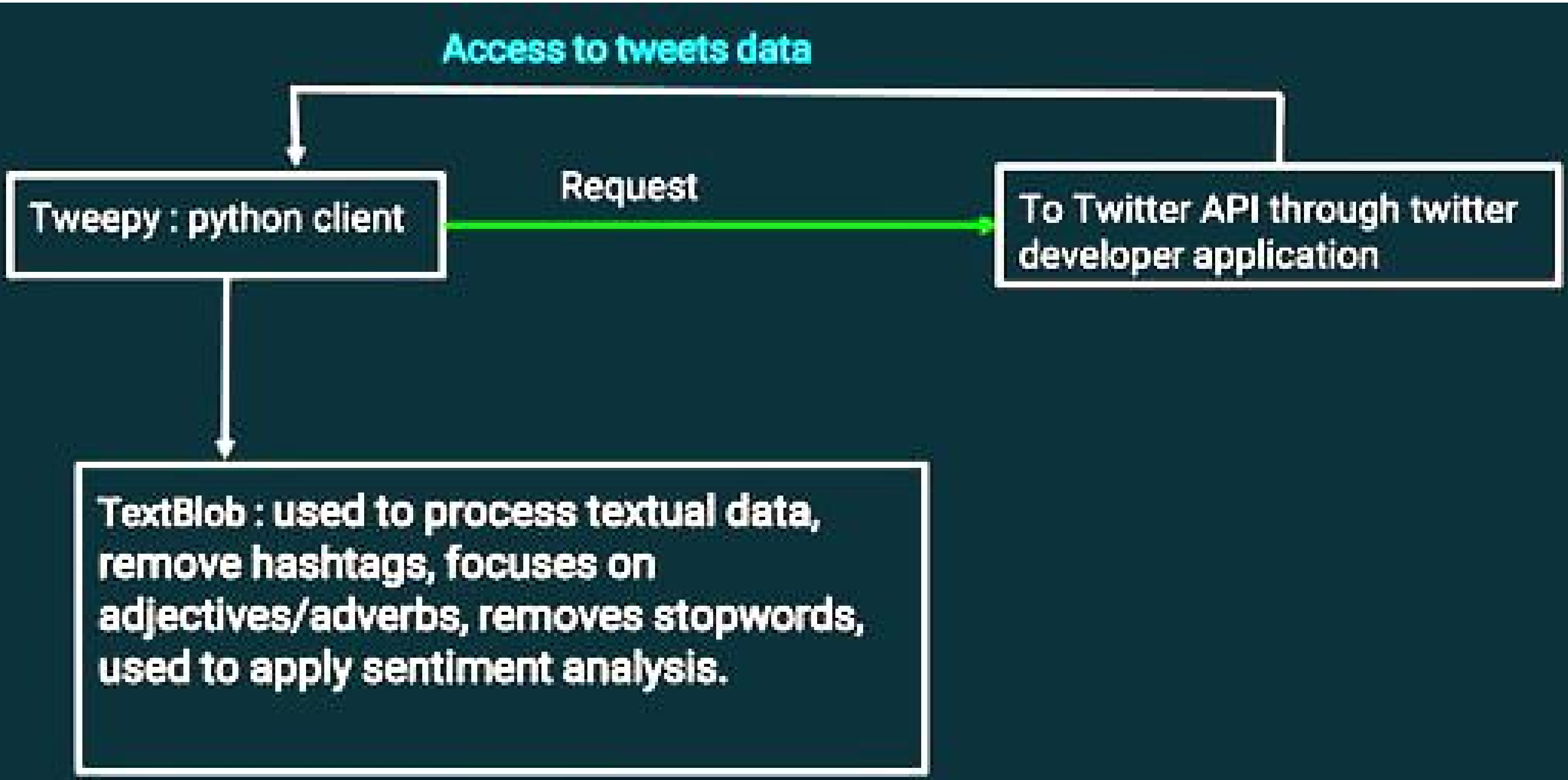
Sentiment Analysis

Step wise procedure

Step 1. Created a Developer application in twitter

Step 2. Parse the tweets data through TextBlob

Step 3. Returned parsed tweets and perform statistical analysis



Access to tweets data

Tweepy : python client

Request

To Twitter API through twitter developer application

TextBlob : used to process textual data, remove hashtags, focuses on adjectives/adverbs, removes stopwords, used to apply sentiment analysis.

Visualization of the implementation

9

Testing

Last step in the process of Development

Types Of Testing

Testing is the last step before the completion of any web application/ program/ website. Like it is mandatory to check all weapons before going for a war, similarly it is necessary to check that the application works well for any standard input and produces desired output, serving the purpose of the application. For the above means, the checklist includes the following testing methods we used in the project:

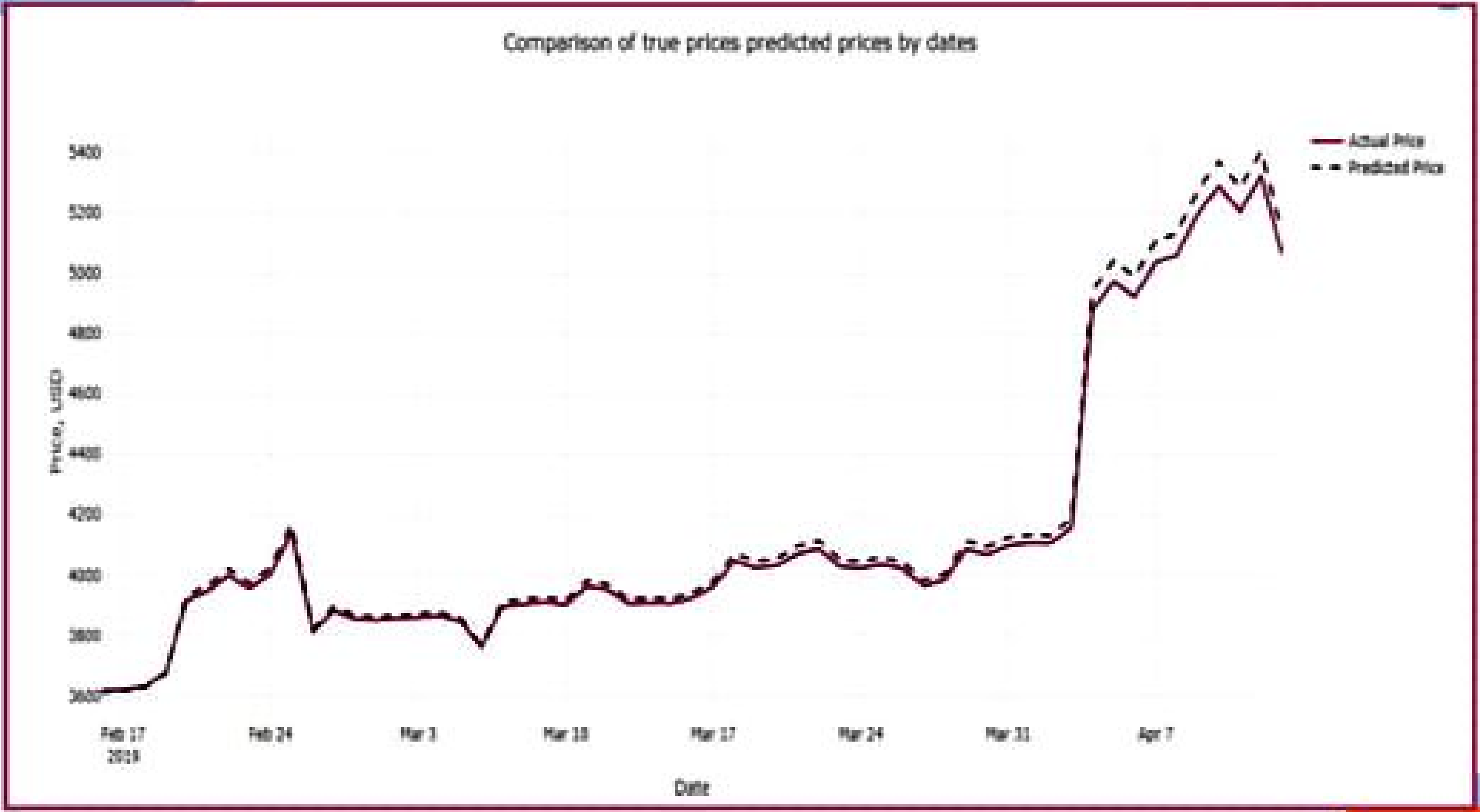
- **Unit Testing** : In this level of testing the individual units/ components of a software are tested. The main purpose is to testify that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

- **Integration Testing** : In this level of software testing , individual units are combined and tested as a group together. The purpose of this level of testing is to expose faults in the interaction between integrated units.

- **System Testing** : It is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

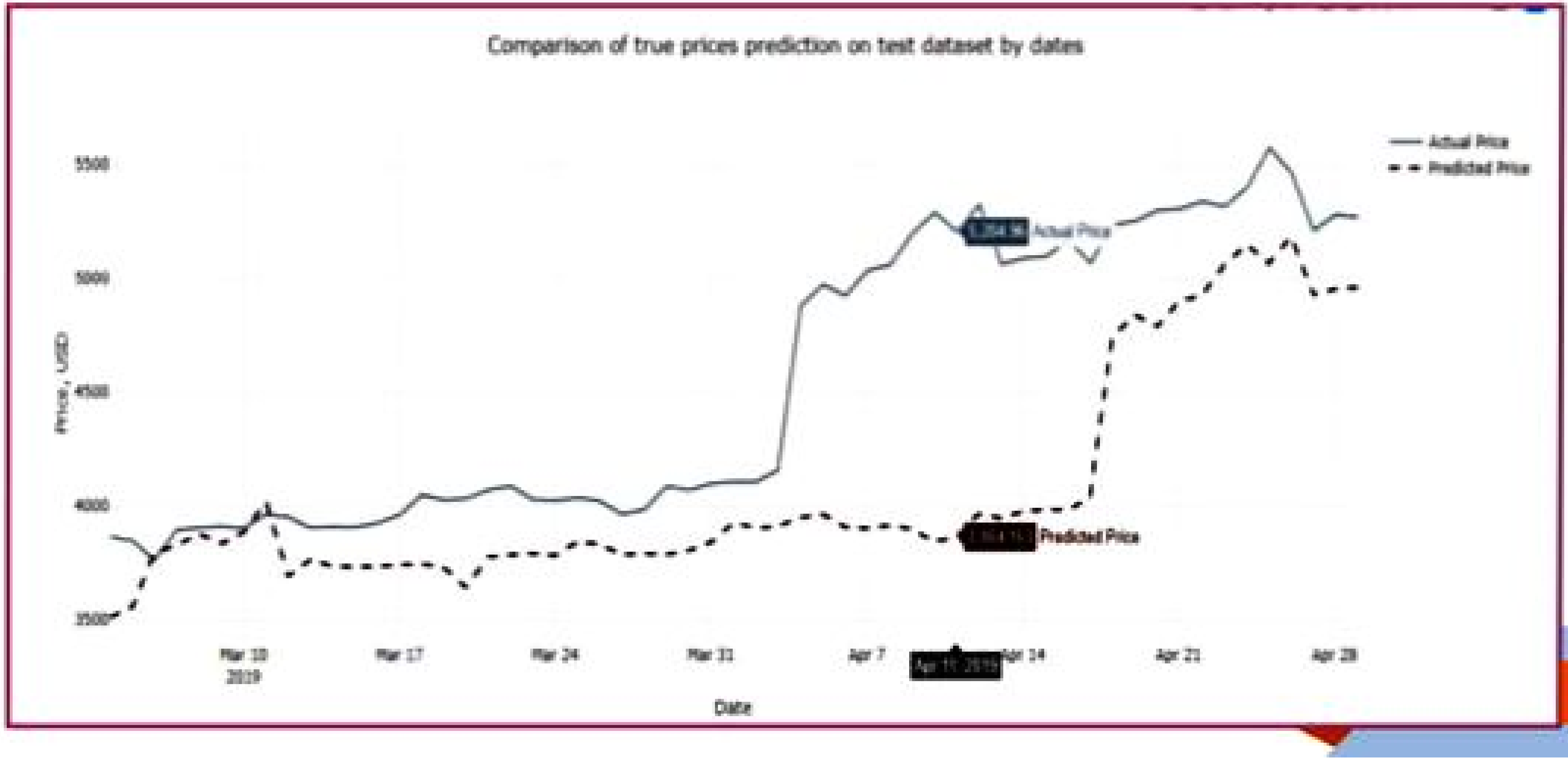
10

Results

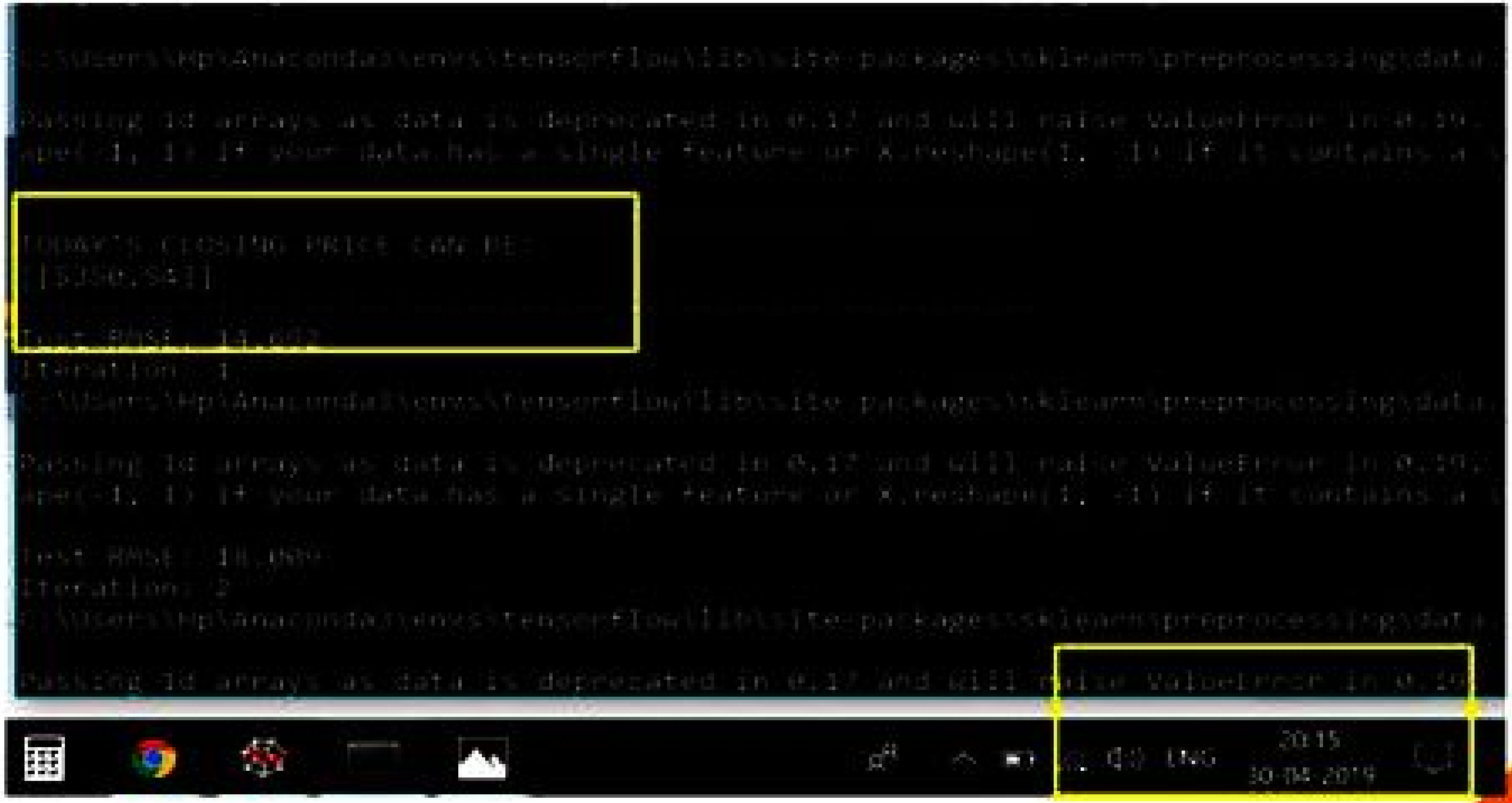


Comparison of true prices predicted prices by dates

Results of GRU Model



Prediction



```
C:\Users\Hp\Anaconda3\envs\tensorflow\lib\site-packages\sklearn\preprocessing\data.

Passing 1d arrays as data is deprecated in 0.17 and will raise ValueError in 0.19.
ape(-1, 1) if your data has a single feature or X.reshape(1, -1) if it contains a s


TODAY'S CLOSING PRICE CAN BE:
[[5356.54]]

Test RMSE: 14.602
Iteration: 1
C:\Users\Hp\Anaconda3\envs\tensorflow\lib\site-packages\sklearn\preprocessing\data.

Passing 1d arrays as data is deprecated in 0.17 and will raise ValueError in 0.19.
ape(-1, 1) if your data has a single feature or X.reshape(1, -1) if it contains a s

Test RMSE: 18.069
Iteration: 2
C:\Users\Hp\Anaconda3\envs\tensorflow\lib\site-packages\sklearn\preprocessing\data.

Passing 1d arrays as data is deprecated in 0.17 and will raise ValueError in 0.19
```

Sentiment Analysis

11

Future Scope

Future Aspects and Incremental Work possible

1. Use to predict the prices of other cryptocurrencies like litecoin, ether etc.,
2. We tried to incorporate the idea of transfer learning, this model can be optimized for better performance.
3. The web application which is the final software product of this project can provide numerous other applications on single site like - live sentiment analysis of bitcoin, tutorial to do trading using bitcoin crypto currency etc.,
4. The result predict from the model and the idea of sentiment analysis can be combined to predict the prices more accurately, *this predicted* price will use both - historical data and current events related to Bitcoin.

THANK YOU